ITMO UNIVERSITY

# Learning Book

## International summer school of Control Systems and Robotics

Saint Petersburg
2019

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

УНИВЕРСИТЕТ ИТМО

**О.И. Борисов, А.А. Ведяков, С.М. Власов,
Д.Н. Герасимов, К.А. Зименко, А.А. Капитонов,
С.А. Колюбин, А.Ю. Краснов, А.А. Маргун,
А.А. Пыркин, М.М. Синетова, А.А. Жиленков**

# INTERNATIONAL SUMMER SCHOOL OF CONTROL SYSTEMS AND ROBOTICS. LEARNING BOOK. PART 1 / МЕЖДУНАРОДНАЯ ЛЕТНЯЯ ШКОЛА ПО СИСТЕМАМ УПРАВЛЕНИЯ РОБОТОТЕХНИКЕ. УЧЕБНОЕ ПОСОБИЕ. ЧАСТЬ 1

УЧЕБНОЕ ПОСОБИЕ

РЕКОМЕНДОВАНО К ИСПОЛЬЗОВАНИЮ В УНИВЕРСИТЕТЕ ИТМО
по направлению подготовки 27.04.03, 27.04.04
в качестве учебного пособия для реализации основных профессиональных
образовательных программ высшего образования магистратуры

УНИВЕРСИТЕТ ИТМО

**Санкт-Петербург
2019**

Борисов О.И., Ведяков А.А., Власов С.М., Герасимов Д.Н., Зименко К.А., Капитонов А.А., Колюбин С.А., Краснов А.Ю., Маргун А.А., Пыркин А.А., Синетова М.М., Жиленков А.А., International summer school of Control Systems and Robotics. Learning book. Part 1 / Международная летняя школа по системам управления робототехнике. Учебное пособие. Часть 1– СПб: Университет ИТМО, 2019. – 405 с.

Рецензент:

Николаев Николай Анатольевич, кандидат технических наук, доцент (квалификационная категория "ординарный доцент") факультета систем управления и робототехники, Университета ИТМО.

The textbook contains theoretical material for studying Control Systems and Robotics. The order of topics follows the structure of the lectures given at ITMO University, Faculty of Control Systems and Robotics. The modern control approaches and digital control systems in robotics are considered. The textbook is intended to foreign students majoring in specializations 27.04.03 System Analysis and Control and 27.04.04 Control in Technical Systems.

Учебное пособие содержит теоретический материал для изучения систем управления и робототехники. Темы в пособии отражают структуру лекций, читаемых в Университете ИТМО на факультете Систем управления и робототехники. В учебном пособии рассматриваются современные подходы к управлению, а также цифровые системы управления в робототехнике. Учебное пособие предназначено для иностранных студентов, обучающихся по направлениям подготовки 27.04.03 Системный анализ и управление и 27.04.04 Управление в технических системах.

УНИВЕРСИТЕТ ИТМО

**Университет ИТМО** – ведущий вуз России в области информационных и фотонных технологий, один из немногих российских вузов, получивших в 2009 году статус национального исследовательского университета. С 2013 года Университет ИТМО – участник программы повышения конкурентоспособности российских университетов среди ведущих мировых научно-образовательных центров, известной как проект «5 в 100». Цель Университета ИТМО – становление исследовательского университета мирового уровня, предпринимательского по типу, ориентированного на интернационализацию всех направлений деятельности.

# Table of Contents

**The modern theory of control systems**

Automatic control theory

Stability types and Lyapunov equations

# Automatic Control Theory. Stability types and Lyapunov Equations

Madina Sinetova

## Stability

- *Stability* is the system ability to return to initial position after stopping action to system external disturbances.

Stable system

Unstable system

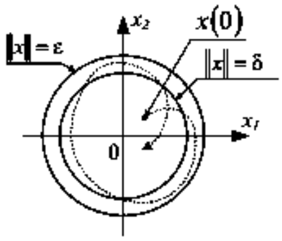
Segway

2

# Stability types

**1. Lyapunov stability.**

- Guarantees bounded of all trajectories, but not guarantees convergence to some steady value.




where $x_1, x_2$ are state coordinates, $\varepsilon, \delta$ – some small numbers; as norm of $x_1$ and $x_2$ can be used quadratic norm, for example; $x(0)$ – initial position of trajectory.

- The equilibrium $x = 0$ is *Lyapunov stable* if for any small number $\varepsilon > 0$, exists small number $\delta(\varepsilon) > 0$, that for all trajectories starting from the initial conditions $\|x(0)\| \leq \delta(\varepsilon)$ for any time $\forall t \geq 0$ following inequality is satisfied: $\|x(t)\| \leq \varepsilon$.

3

# Root stability criterion

Given continuous system:

$$\dot{x} = Fx, x \in R^n, F - n \times n.$$

Characteristic polynomial of the given system:

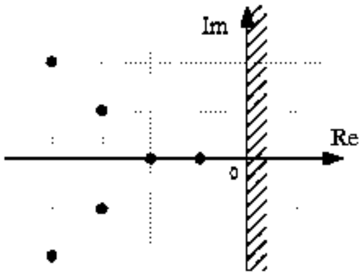
$$\det(F - sI) = s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + a_0 = 0,$$

where $s_i, i = \overline{1, n}$ – roots of the polynomial, $I$ – identity matrix.

If all roots have *negative real parts* $\mathrm{Re}(s_i) < 0, i = \overline{1, n}$, then the system is stable.

Im – imaginary axis (*stability border)*,

Re – real axis.




4

# Root stability criterion

In discrete case instead of function derivative is used value on the next discrete step:

$$\dot{f}(t) \sim f(m+1),$$

where $m$ – number of discrete interval, $t = mT$ – continuous time, $T$ – value of discrete interval.

Consider the discrete system:

$$x(m+1) = F_d x(m), x \in R^n, F_d - n \times n.$$

Characteristic polynomial of the given system:

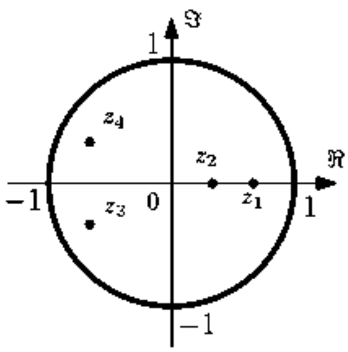
$$\det[F_d - zI] = z^n + \alpha_{n-1} z^{n-1} + \cdots + \alpha_1 z + \alpha_0 = 0,$$

where z – is a delay, $I$ – is an identity matrix, and $z_i, i = \overline{1,n}$ – roots of the polynomial.

5

# Root stability criterion

If all absolute values of roots less than one $|z_i| < 1, i = \overline{1,n}$, then the system is *stable*.




- The unit circle is a *stability border*.
- If one or more than one absolute values of roots more than $|z_i| > 1$, the system is *unstable*.

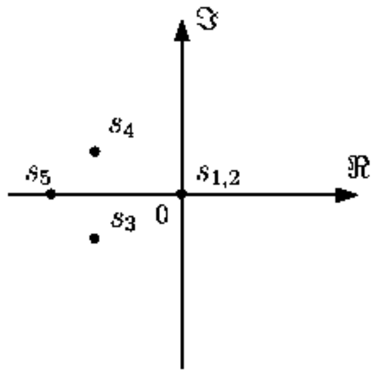
6

# Stability borders

**1.1. Stability border of neutral type.**

Dynamic **continuous** system is on the border of neutral type if one or two roots of characteristic polynomial are equal to zero and rest roots have negative real parts:

$$s_{1,2} = 0, \mathrm{Re}(s_i) < 0, i = \overline{3, n}.$$



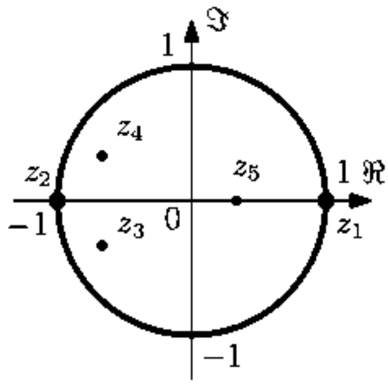

7

# Stability borders

**1.1. Stability border of neutral type.**

Dynamic **discrete** system is on the border of neutral type if one or two roots of characteristic polynomial are equal to one and rest roots are in the unit circle:

$$|z_{1,2}| = 1, z_i < 1, i = \overline{3, n}.$$




8

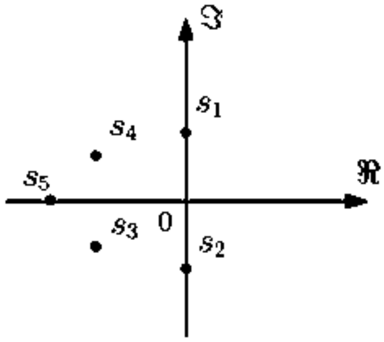
# Stability borders

**1.2. Stability border of oscillatory type.**

The dynamic **continuous** system is on the border of oscillatory type if the characteristic polynomial has pair of purely imaginary roots and rest roots have negative real parts:

$$s_{1,2} = \pm j\omega, \omega > 0, \mathrm{Re}(s_i) < 0, i = \overline{3, n},$$

where $j$ – imaginary unit.



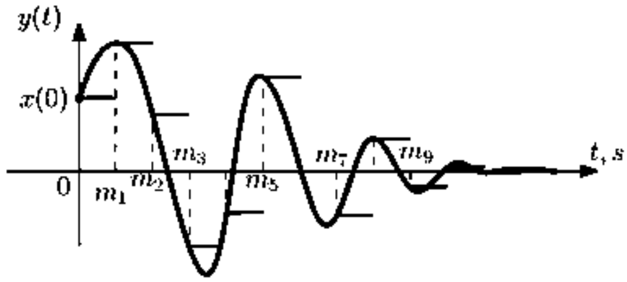

9

# Stability types

**2. Asymptotic stability**

The equilibrium $x = 0$ is *asymptotically stable* if the equilibrium is Lyapunov stable and for any motion trajectories $x(t)$ from the arbitrary initial conditions $x(0)$ the condition $\lim_{t\to\infty} \|x(t)\| = 0$ is satisfied.

In discrete case trajectories are $x(m)$ and condition is $\lim_{m\to\infty} \|x(m)\| = 0$.

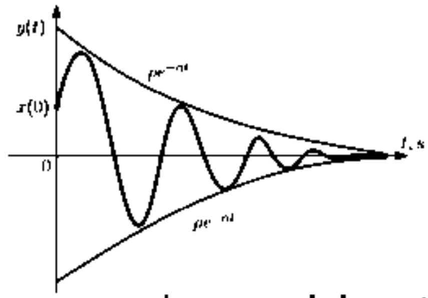



10

# Stability types

**3. Exponential stability**

The equilibrium $x = 0$ is *exponential stable* if for any motion trajectories $x(t)$ from the arbitrary initial conditions $x(0)$ exists positive number $\alpha > 0$ that for any time $\forall t \geq 0$ inequality: $\|x(t)\| \leq \rho e^{-\alpha t}\|x(0)\|$; $\rho \geq 1$ is satisfied.



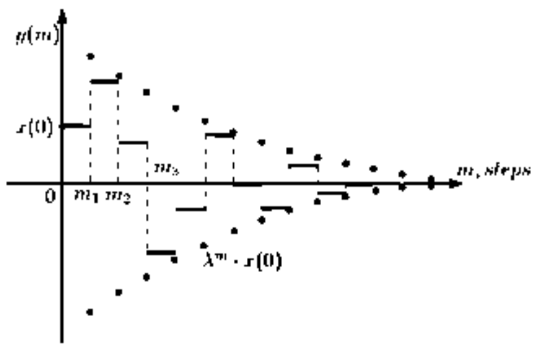

Constant $\alpha$ is the convergence degree and characterizes convergence velocity to equilibrium.

11

# Stability types

**3. Exponential stability**

In discrete case: $\|x(m)\| \leq \rho\lambda^m\|x(0)\|$; $\rho \geq 1, \lambda < 1$.




Number $\lambda$ characterizes *convergence velocity*. The smaller $\lambda$ the faster convergence.

12

# Stability types

**4. Qualitative exponential stability**

The equilibrium $x = 0$ is qualitative exponential stable if for any motion trajectories $x(t)$ from the arbitrary initial conditions exists numbers $\alpha > 0, r > 0, \rho \geq 1$ that for any time $\forall t \geq 0$ the following inequality:

$$\|x(t) - e^{-\alpha t}x(0)\| \leq \rho\left(e^{-(\alpha+r)t} - e^{-\alpha t}\right)\|x(0)\|$$

is satisfied.

In discrete case:

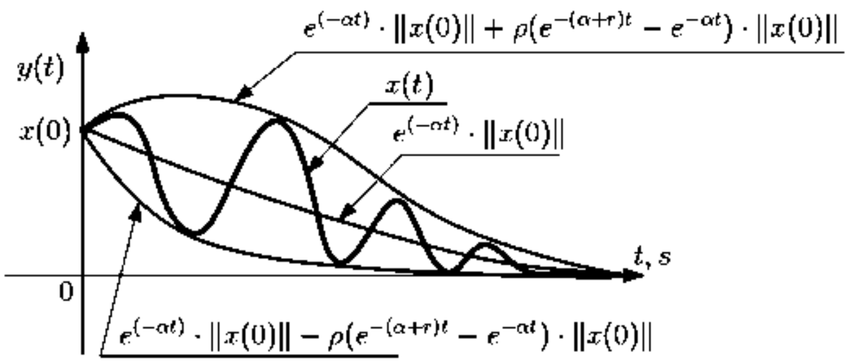
$$\|x(m) - \alpha^m x(0)\| \leq \rho\left((\alpha + r)^m - \alpha^m\right)\|x(0)\|,$$

where $0 \leq \alpha < 1 - r$.

13

# Stability types

**4. Qualitative exponential stability**




Parameter $\alpha$ characterizes velocity convergence to equilibrium.
Parameter $r$ characterizes trajectory average deviation.

14

# Lyapunov functions

Lyapunov functions $V(x)$ have properties:

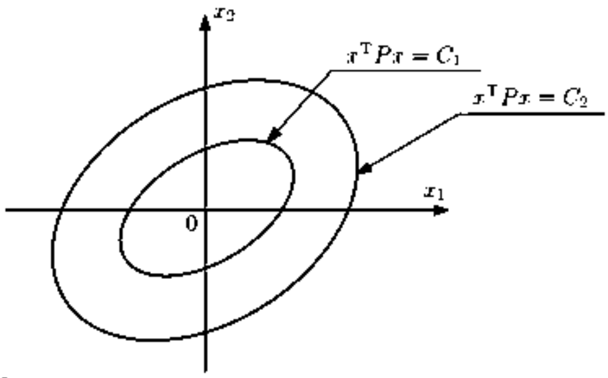
1. Lyapunov function $V(x)$ must be positive definite: for any $\forall x \in R^n$ Lyapunov function $V(x)$ is positive definite and $V(x) = 0$ in case $x$ is null-vector.
2. Lyapunov functions must increases (decreases) uniformly with uniform increasing (decreasing) of $x$-vector norm.
3. The surfaces of constant level $V(x) = C$, where $C$ – is a constant, must cover the origin of coordinates or equilibrium.

15

# Lyapunov functions

Quadratic forms: $V(x) = x^T P x$,

$P$ – $n \times n$ positive definite symmetric square matrix.




$C_2 > C_1$,

$P = I$ – identity matrix,

$x^T P x = \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = C$,

$x_1^2 + x_2^2 = C = \|x\|^2 = \left(\sqrt{x_1^2 + x_2^2}\right)^2$.

16

# Lyapunov Theorem

The equilibrium $x = 0$ is asymptotically stable if exists Lyapunov function $V(x)$ such that for any motion trajectories $x(t)$ starting from the arbitrary initial conditions for any time $\forall t \geq 0$ the derivative of the function is negative: $\frac{dV(x(t))}{dt} < 0$.

$$\frac{dV\big(x(t)\big)}{dt} = \frac{\partial V(x)}{\partial x}\frac{\partial x}{\partial t} = \frac{\partial V(x)}{\partial x}\dot{x}.$$

$x - n$-dimension state vector:

$$\frac{\partial V(x)}{\partial x} = \begin{bmatrix}\frac{\partial V(x)}{\partial x_1} & \dots & \frac{\partial V(x)}{\partial x_n}\end{bmatrix} = grad^T V(x).$$

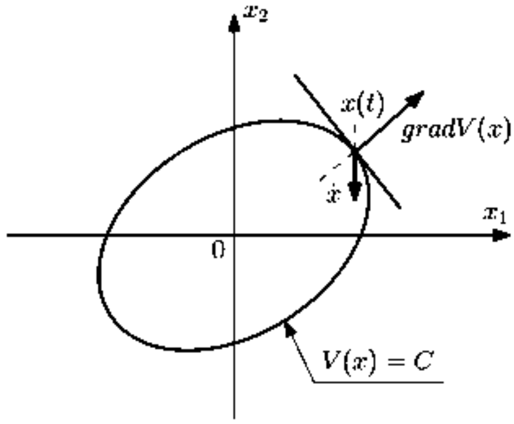
So:

$$\frac{dV\big(x(t)\big)}{dt} = grad^{\mathrm{T}} x \cdot \dot{x}.$$

17

# Lyapunov Theorem

Geometric interpretation:




$V(x) = C$ – surface of constant level.

18

# Example

Given:

$$\dot{x} = -x^3.$$

Lyapunov function:

$$V(x) = x^2.$$

Investigation:

$$\frac{dV\big(x(t)\big)}{dt} = 2x \cdot \dot{x} = 2x \cdot (-x^3) = -2x^4 < 0.$$

Lyapunov function derivative is negative anytime, so, the given system is asymptotically stable.

19

# Lyapunov inequalities

- For asymptotic stability:

$$\dot{V}\big(x(t)\big) < 0.$$

- For exponential stability:

$$\dot{V}\big(x(t)\big) \le -2\alpha V\big(x(t)\big), \alpha > 0.$$

- For qualitative exponential stability:

$$V\big(\dot{x}(t) + (r + \alpha)x(t)\big) \le r^2 V\big(x(t)\big).$$

20

# Rayleigh ratio

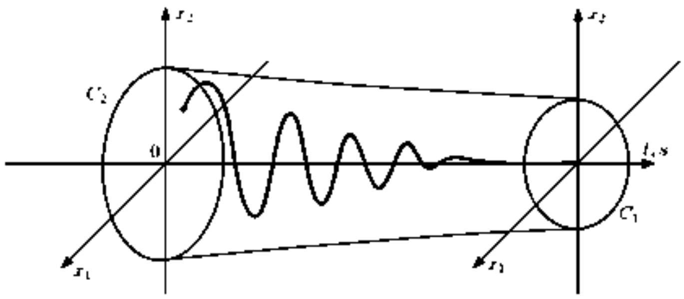
**Let's consider inequality:**

$$C_1^2\|x\|^2 \le x^T P x \le C_2^2\|x\|^2,$$

**where** $\dot{x} = Fx, x \in R^n, F - n{\times}n, V(x) = x^T P x, P - n{\times}n$ **positive definite symmetric square matrix.**

**Omitting intermediate calculations obtain ratio:**

$$\|x(t)\| \le \frac{C_2}{C_1} e^{-\alpha t}\|x(0)\|, \rho = \frac{C_2}{C_1} \ge 1.$$



21

# Lyapunov equations

**From Lyapunov inequalities follows Lyapunov equations:**

- **For asymptotic stability:**

$$F^T P + PF = -Q.$$

- **For exponential stability:**

$$F^T P + PF + 2\alpha F = -Q.$$

- **For qualitative exponential stability:**

$$(F + (r+\alpha)I)^T P(F + (r+\alpha)I) - r^2 P = -Q.$$

$F$ **–state matrix of closed system,** $P, Q$ **– positive definite symmetric square matrices of the same dimension.**

**For investigation stability it's required to choose matrix** $Q$**, solve Lyapunov equation with respect to matrix** $P$ **and check it for positive definition.**

22

# Example

Given:

$$\dot{x} = Fx, \begin{bmatrix} \dot{x_1} \\ \dot{x_2} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}.$$

Is system stable?

Choose:

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, P = \begin{bmatrix} p_1 & p_3 \\ p_3 & p_2 \end{bmatrix}.$$

And calculate:

$$F^T P = \begin{bmatrix} -p_3 & -p_2 \\ p_2 - 2p_3 & p_3 - 2p_2 \end{bmatrix}, PF = \begin{bmatrix} -p_3 & p_1 - 2p_3 \\ -p_2 & p_3 - 2p_2 \end{bmatrix},$$

23

# Example

$$\begin{cases} -p_3 - p_3 = -1 \\ -p_2 + p_1 - 2p_3 = 0 \\ p_2 - 2p_3 - p_2 = 0 \\ p_3 - 2p_2 + p_3 - 2p_2 = -1 \end{cases} \Rightarrow P = \begin{bmatrix} 1{,}5 & 0{,}5 \\ 0{,}5 & 0{,}5 \end{bmatrix}.$$

$$\det(P - \lambda I) = \lambda^2 - 2\lambda + 0{,}5 = 0 \Rightarrow \lambda_1 = 0{,}29, \lambda_2 = 1{,}7.$$

All eigenvalues $\lambda_i$ of matrix $P$ are more than zero, so, system is asymptotically stable.

24

Qualitative exponential stability

# Automatic Control Theory. Qualitative exponential stability for discrete and continuous linear systems

Madina Sinetova

## Qualitative exponential stability

Consider linear discrete system:

$$x(m+1) = F\big(x(m)\big).$$

Equilibrium $x = 0$ is *exponential stable*, if exists numbers $\rho > 0$, $\alpha > 0$, $d_x(\alpha) > 0$ that for all initial values of $\|x(0)\| \le d_x(\alpha)$ for any number of discreetness interval $m > 0$ the following inequality is satisfied:

$$\|x(m)\| \le \rho \cdot e^{-\alpha m} \cdot \|x(0)\|.$$

Introduce notation: $\lambda = e^{-\alpha}$ , where $0 < \lambda < 1$, so:

$$\|x(m)\| \le \rho \cdot \lambda^m \cdot \|x(0)\|.$$

All trajectories $x(m)$ of exponential stable system are in «estimated tube» bounded by surfaces:

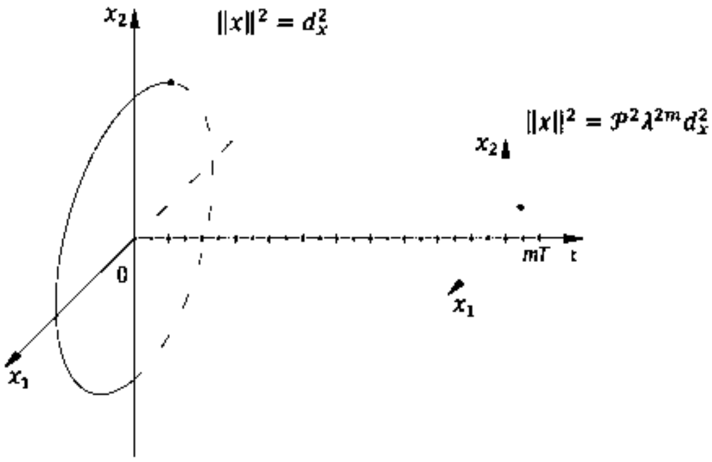
$$\|x(m)\|^2 = (\rho \cdot \lambda^m \cdot \|x(0)\|)^2.$$

2

# Qualitative exponential stability

Surfaces consist of circles with radiuses:

$$\rho \cdot \lambda^m \cdot \|x(0)\|.$$




3

# Local term

Equilibrium $x = 0$ is *qualitative exponential stable*, if system is exponential stable with parameters $\alpha$ ($\lambda = e^{-\alpha}$), $d_x(\alpha)$ and additionally exists positive number $0 < \lambda_0 < 1 + \lambda$ that for any number of discreetness interval $m > 0$ the following inequality is satisfied:

$$\|x(m) - x(0)\| \leq \lambda_0 \rho \sum_{i=0}^{m-1} \lambda^i \|x(0)\| = \lambda_0 \rho \frac{1 - \lambda^m}{1 - \lambda} \|x(0)\|.$$

This condition constrains state vector current values deviation from initial conditions $x(0)$.

System have very qualitative parameters under condition: $\lambda_0 < 1$. In this case inequality is strongest.

4

# Qualitative exponential stability

In case of non-zero initial conditions $x(0)$ trajectories are bounded by surfaces:

$$\|x(m) - x(0)\|^2 = \left[\lambda_0 \rho \frac{1-\lambda^m}{1-\lambda} d_x\right]^2.$$

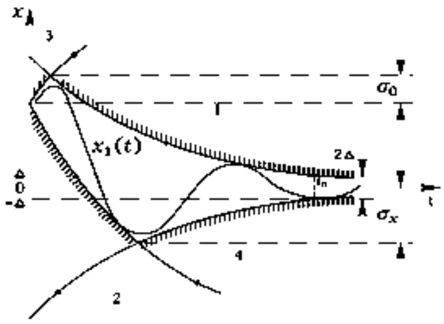
with circles' radiuses:

$$\lambda_0 \rho \frac{1-\lambda^m}{1-\lambda} d_x.$$

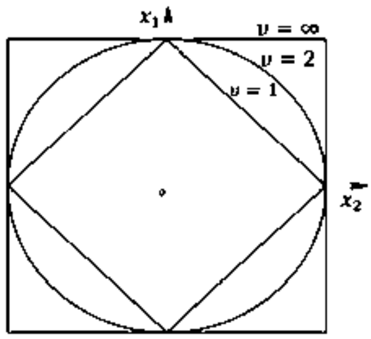
«Estimated tube» cross-section:

$\sigma_{0x}^* = \frac{(\rho-1)\lambda_0}{\lambda+\lambda_0-1}$ - first ejection;

$\sigma_x^* = \frac{\lambda-\rho\lambda_0-1}{1-\lambda+\lambda_0}$ - overcontrol.




# Vector norms




Arbitrary vector norm:

$$\|x\| = \left[\sum_{i=1}^{n} |x_i|^v\right]^{\frac{1}{v}},$$

$v$ is an integer and $v = 1, 2, \ldots$, and $x_i$ – $i$-th component of state vector $x$.

If $v = 2$ the norm is *Euclidean*, if $v = 1$ the norm is *absolute*.

In case of two state components constant level surfaces are:

$$\|x\|^v = 1 \; (x \in R^2).$$

*Remark.* For deterministic processes (not for stochastic) from the convergence by some norm follows convergence by any norm.

6

# Lyapunov function

Consider convex positively homogeneous Lyapunov function $V(x)$ from a class of $K^v$ such:

$$C_1^v\|x\|^v \leq V(x) \leq C_2^v\|x\|^v.$$

Using quadratic forms:

$$V(x) = x^T P x,$$

where $P$ – symmetric positive definite square matrix from a class of $K^2$, values $C_1^2$ and $C_2^2$ are minimum and maximum eigenvalues of matrix $P$ respectively.

7

# Sufficient conditions

For system $x(m+1) = F\big(x(m)\big)$ sufficiently existing number

$0 < \lambda < 1$ that for any number of discreetness interval $m > 0$ the inequality is satisfied:

$$V\big(x(m+1)\big) \leq \lambda^v V\big(x(m)\big),$$

and existing number $1 - \lambda < \lambda_0 < 1 + \lambda$ , that for the system the inequality is satisfied:

$$V\big(x(m+1) - x(m)\big) \leq \lambda_0^v V\big(x(m)\big).$$

From these conditions follows two consequences.

8

# Consequences

**Consequence 1.** For qualitative exponential stability are sufficiently existing numbers $0 < \lambda < 1$ and $1 - \lambda < \lambda_0 < 1 + \lambda$ that for any number of discreetness interval $m > 0$ the following inequality is satisfied:

$$V\big(x(m+1) - (r+\alpha)x(m)\big) \le r^{v} V\big(x(m)\big).$$

**Consequence 2.** For qualitative exponential stability are sufficiently existing numbers $0 < r < 1$ and $0 < \alpha < 1 - 2r$ that for any number of discreetness interval $m > 0$ the following inequality is satisfied:

$$V\left(x(m+1) - \frac{\lambda - \lambda_0 + 1}{2} x(m)\right) \le \left(\frac{\lambda + \lambda_0 - 1}{2}\right)^{v} V\big(x(m)\big).$$

$\lambda = 2r + \alpha, \lambda_0 = 1 - \alpha.$

9

# Geometric interpretation

**Exponential stability.**

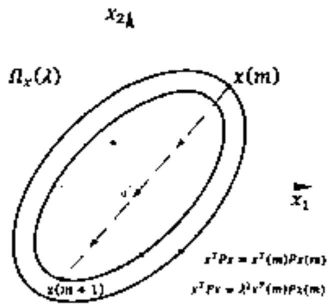
In case of quadratic form:

$$V\big(x(m+1)\big) \le \lambda^2 V\big(x(m)\big).$$

The each next value of state vector $x(m+1)$ must belongs to area:

$$\Omega_x(\lambda) = \{\, x : x^T P x \le \lambda^2 x^T(m) P x(m) \}$$

if the previous value of state vector was on a surface $x^T P x = x^T(m) P x(m)$.




10

# Geometric interpretation

**Consequence 1.**

1. System must be exponential stable.

2. For Lyapunov functions from a class of $K^2$ the inequality which should be satisfied takes form:

$$\left(x(m+1)-x(m)\right)^T P\left(x(m+1)-x(m)\right) \leq \lambda_0^2 x^T(m)Px(m).$$

And the each next value of state vector $x(m+1)$ with fixed $x(m)$ must belongs to area:

$$\Omega_x(\lambda_0) = \left\{ x\colon \left(x-x(m)\right)^T P\left(x-x(m)\right) \leq \lambda_0^2 x^T(m)Px(m)\right\}.$$

3. The each fixed arbitrary value $x(m)$ the next value of state vector $x(m+1)$ belongs to area:

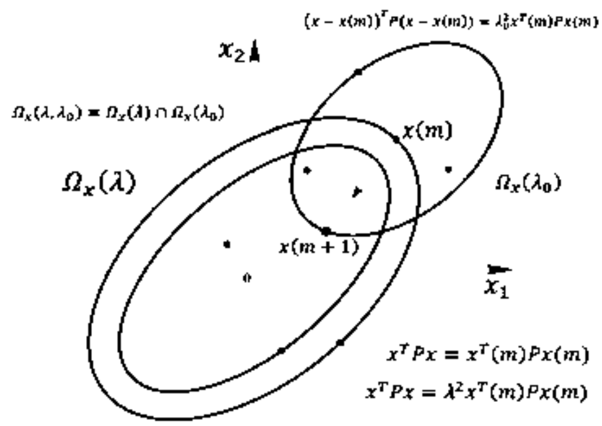
$$\Omega_x(\lambda,\lambda_0) = \Omega_x(\lambda) \cap \Omega_x(\lambda_0).$$

11

# Geometric interpretation

**Consequence 1.**

As a result, qualitative exponential stability distinguish from all values of state vector $\Omega_x(\lambda)$ some part $\Omega_x(\lambda,\lambda_0)$ and it localizes system motion trajectories behavior.




12

# Geometric interpretation

**Consequence 2.**

1. System must be exponential stable.

2. For Lyapunov functions from a class of $K^2$ the inequality which should be satisfied takes form:

$$\big(x(m+1)-(r+\alpha)x(m)\big)^T P\big(x(m+1)-(r+\alpha)x(m)\big) \le r^2 x^T(m)Px(m),$$

And the each next value of state vector $x(m+1)$ with fixed $x(m)$ must belongs to area:

$$\Omega_x(r,\alpha)=\left\{x:\big(x-(r+\alpha)x(m)\big)^T P\big(x-(r+\alpha)x(m)\big) < r^2 x^T(m)Px(m)\right\}.$$

3. The each fixed arbitrary value $x(m)$ the next value of state vector $x(m+1)$ belongs to area:

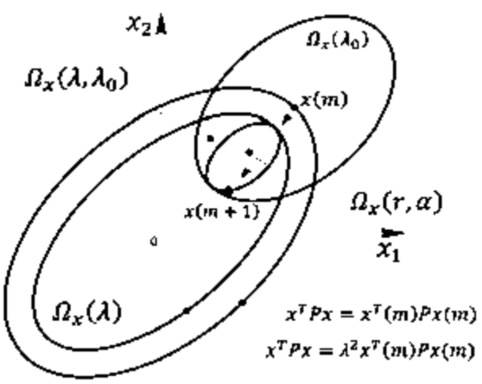
$$\Omega_x(r,\alpha)\subset\Omega_x(\lambda,\lambda_0).$$

13

# Geometric interpretation

**Consequence 2.**

In case $\lambda=2r+\alpha$, $\lambda_0=1-\alpha$ the area belongs to intersection areas $\Omega_x(\lambda_0)$ and $\Omega_x(\lambda)$: $\Omega_x(r,\alpha)\subset\Omega_x(\lambda,\lambda_0)$.




14

# Summary

Consider system:

$$x(m+1) = F \cdot \big(x(m)\big),$$

$F - n{\times}n$ matrix of closed system.

Lyapunov inequality:

$$(F - (r+\alpha)I)^T P (F - (r+\alpha)I) \le r^2 P,$$

where $\lambda = 2r + \alpha$, $\lambda_0 = 1 - \alpha$.

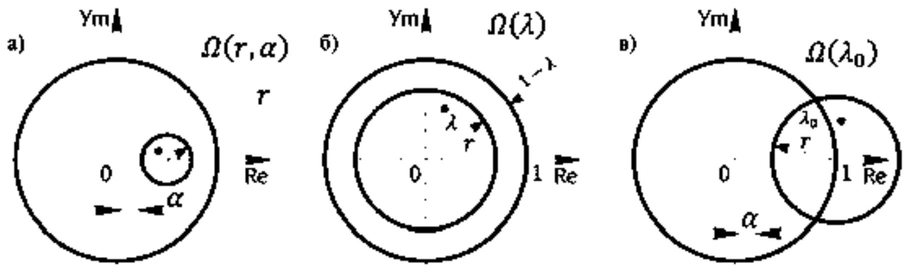
Lyapunov equation:

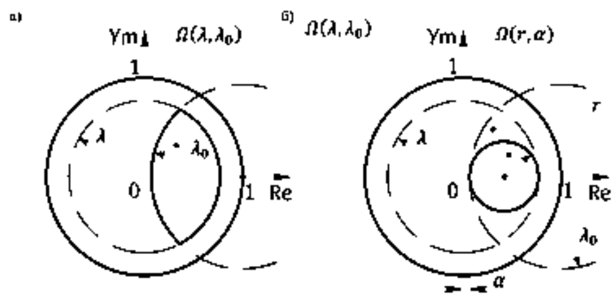
$$(F - (r+\alpha)I)^T P (F - (r+\alpha)I) - r^2 P = -Q,$$

where $P > 0$ – positive definite symmetric square matrix, $Q \ge 0$ – positive semi-definite symmetric square matrix.

15

# Roots distribution




16

# Continuous case

Consider continuous linear system:

$$\dot{x} = F(x),$$

$x$ – state vector.

Local conditions:

1. There is a number $\alpha > 0$ such for any time $t > 0$ following inequality is satisfied:

$$\dot{V}(x(t)) \leq -2\alpha V(x(t)),$$

2. There is a number $\lambda_0 \geq \alpha$ such for any time $t > 0$ following inequality is satisfied:

$$V(\dot{x}(t)) \leq \lambda_0^2 V(x(t)),$$

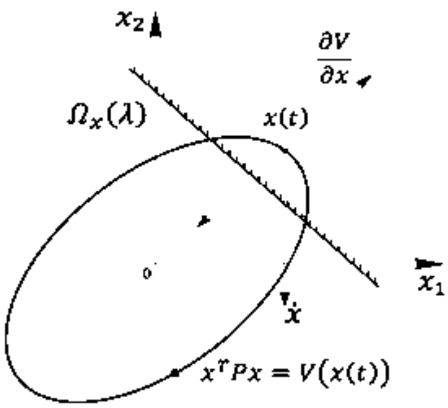
where $V(x(t))$ – Lyapunov function from a class $K^2$ of quadratic forms.

17

# Geometric interpretations

1st condition:

$$\frac{\partial V(x)}{\partial x}\dot{x} \leq -2\alpha V(x).$$




18

# Geometric interpretations

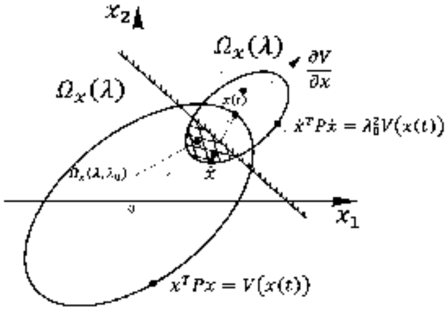
2nd condition:

All possible values of state vector must belong to area $\Omega_x(\lambda_0)$ bounded by surface:

$$\dot{x}^T P \dot{x} = \lambda_0^2 V\big(x(t)\big).$$

As a result, combine both conditions:

Values of state vector must belong to area $\Omega_x(\alpha, \lambda_0) = \Omega_x(\alpha) \cap \Omega_x(\lambda_0)$.




19

# Geometric interpretations

**As one condition.**

System is qualitative exponential stable if exists numbers $\alpha > 0$ and $r > 0$ such for any time $t > 0$ following inequality is satisfied:

$$V\big(\dot{x}(t) + (r+\alpha)x(t)\big) \leq r^2 V\big(x(t)\big),$$

where $\lambda = \alpha$, $\lambda_0 = 2r + \alpha$.

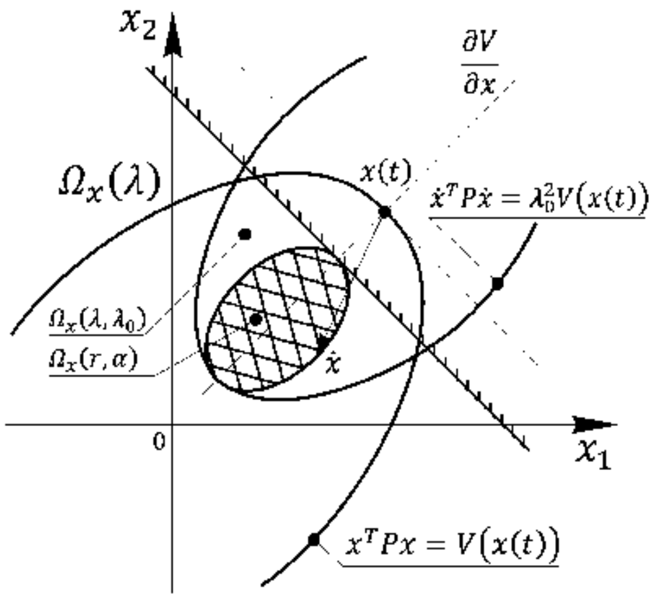
For any time $t > 0$ and for any state vector $x(t)$, velocity vector $\dot{x}(t)$ must belongs to area $\Omega_x(r, \alpha)$ bounded by surface:

$$\big(\dot{x} + (r+\alpha)x(t)\big)^T P\big(\dot{x} + (r+\alpha)x(t)\big) = r^2 V\big(x(t)\big),$$

where area $\Omega_x(r, \alpha) \subset \Omega_x(\alpha, \lambda_0)$.

20

# Geometric interpretations




21

# Continuous case

Linear system:

$$\dot{x} = F(x),$$

$x$ – state vector.

**Sufficient condition:**

Existing such numbers $(r, \alpha)$: $\lambda = \alpha$, $\lambda_0 = 2r + \alpha$

that Lyapunov equation:

$$(F + (r + \alpha)I)^T P (F + (r + \alpha)I) - r^2 P = -Q,$$

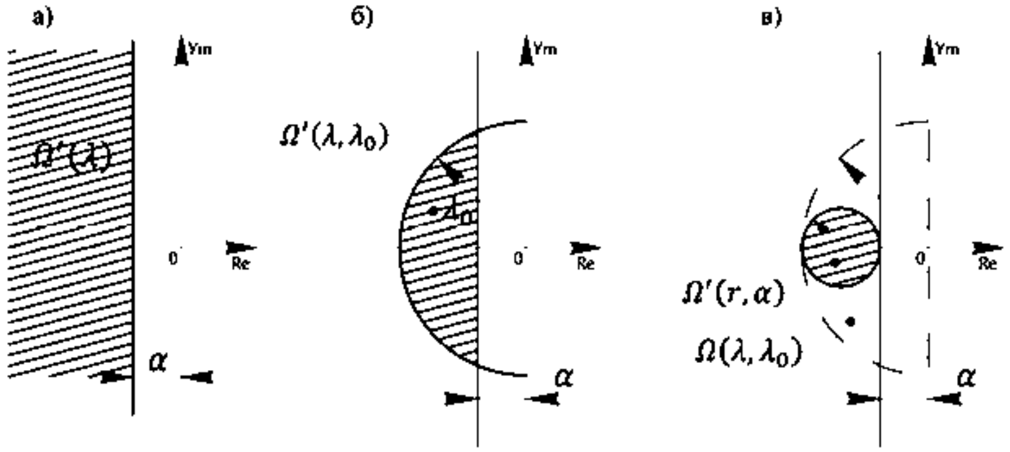
and Lyapunov inequality:

$$(F + (r + \alpha)I)^T P (F + (r + \alpha)I) \leq r^2 P$$

are satisfied.

22

# Roots distribution




23

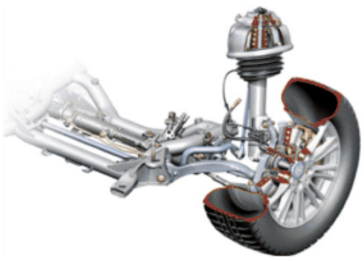
Identification theory

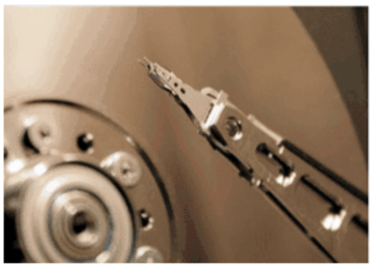
# Identification Theory

Alexey Vedyakov

## Application

- Measuring systems
- Disturbance compensation systems:
  - hard drives
  - ship
  - active suspension vehicle





## First global convergent estimators

- Hsu L., Ortega R., Damm G. A globally convergent frequency estimator // IEEE Transactions on Automatic Control. 1999.
- G. Obregon-Pulido, B. Castillo-Toledo, A. A. Loukianov, "Globally convergent estimator for n-frequencies," IEEE Trans. Autom. Control, vol. 47, no. 5, pp. 857-863, May 2002.
- A. Bobtsov, A. Lyamin, D. Romasheva, "Algorithm of parameter's identification of polyharmonic function," in Proc. 15th IFAC World Congress on Automatic Control, Barcelona, Spain, Jul. 2002.
- X. Xia, "Global frequency estimation using adaptive identifiers," IEEE Trans. Autom. Control, vol. 47, no. 7, pp. 1188-1193, Jul. 2002.
- R. Marino, P. Tomei, "Global estimation of unknown frequencies," IEEE Trans. Autom. Control, vol. 47, no. 8, pp. 1324-1328, Aug. 2002.

## Frequency estimation

Consider the measurable signal

$$y(t) = A\sin(\omega t + \phi), \tag{1}$$

where $A$ is the amplitude, $\omega$ is the frequency, $\phi$ is the phase.

The goal is to obtain the frequency estimate $\hat{\omega}(t)$ such that

$$\lim_{t\to\infty} |\omega - \hat{\omega}(t)| = 0.$$

## Sinusoidal signal generator

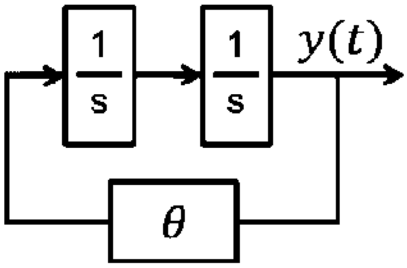
Consider derivatives of the signal (1)

$$\dot{y}(t) = \omega A \cos(\omega t + \phi),$$
$$\ddot{y}(t) = -\omega^2 A \sin(\omega t + \phi). \quad (2)$$

Using (1) and (2) we can obtain linear regression model

$$\ddot{y}(t) = \theta y(t), \quad (3)$$

where $\theta = -\omega^2$.




## Gradient method

Consider the cost criterion

$$J(\theta) = \frac{1}{2}(\ddot{y}(t) - \hat{\ddot{y}}(t))^2 = \frac{1}{2}(\theta y(t) - \hat{\theta}(t) y(t))^2 = \frac{1}{2} e^2(t),$$

which we minimize with respect to $\hat{\theta}(t)$ using the gradient method

$$\dot{\hat{\theta}}(t) = -\gamma \nabla J(\hat{\theta}),$$

where $\gamma > 0$. In our case,

$$\nabla J(\hat{\theta}) = \frac{dJ}{d\theta} = -e y(t) = -y(t)\left(\ddot{y}(t) - \hat{\theta}(t) y(t)\right).$$

Finally,

$$\dot{\hat{\theta}}(t) = \gamma y(t)\left(\ddot{y}(t) - \hat{\theta}(t) y(t)\right).$$

## Biased sinusoidal signal

Consider the measurable signal

$$y(t) = A_0 + A\sin(\omega t + \phi), \tag{4}$$

where $A_0$ is the constant bias. Consider derivatives of the signal (4)

$$\dot{y}(t) = \omega A\cos(\omega t + \phi), \tag{5}$$
$$\ddot{y}(t) = -\omega^2 A\sin(\omega t + \phi).$$
$$\dddot{y}(t) = -\omega^3 A\cos(\omega t + \phi). \tag{6}$$

Using (5) and (6) we can obtain linear regression model

$$\dddot{y}(t) = \theta\dot{y}(t).$$

The adaptive law

$$\dot{\hat{\theta}}(t) = \gamma\dot{y}(t)\left(\dddot{y}(t) - \hat{\theta}(t)\dot{y}(t)\right). \tag{7}$$

## Modified version

Let us consider additional variable

$$\chi(t) = \hat{\theta}(t) - \gamma\dot{y}(t)\ddot{y}(t), \quad \text{then} \tag{8}$$
$$\hat{\theta}(t) = \chi(t) + \gamma\dot{y}(t)\ddot{y}(t). \tag{9}$$

Differentiating equation (9) we obtain

$$\dot{\hat{\theta}}(t) = \dot{\chi}(t) + \gamma\ddot{y}^2(t) + \gamma\dot{y}(t)\dddot{y}(t). \tag{10}$$

On the other hand, from (7) we have

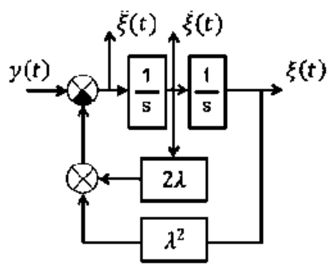
$$\dot{\hat{\theta}}(t) = \gamma\dot{y}(t)\left(\dddot{y}(t) - \hat{\theta}(t)\dot{y}(t)\right) = \gamma\dot{y}(t)\dddot{y}(t) - \gamma\hat{\theta}(t)\dot{y}^2(t). \tag{11}$$

Combining (10) and (11) gives

$$\hat{\theta}(t) = \chi(t) + \gamma\dot{y}(t)\ddot{y}(t),$$
$$\dot{\chi}(t) = -\gamma\hat{\theta}(t)\dot{y}^2(t) - \gamma\ddot{y}^2(t).$$

## Without measuring derivatives

Let us consider linear filter




The signals $\xi$, $\dot\xi(t)$, $\ddot\xi(t)$ are measurable. Moreover,

$$\xi(t) = B_0 + B_1 \sin(\omega t + \psi) + \epsilon(t), \tag{12}$$

where $\epsilon(t)$ is exponentially decaying term. In this case,

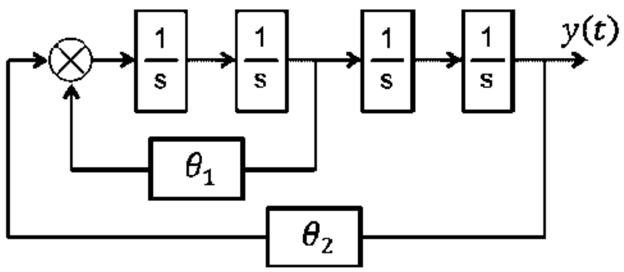
$$\hat\theta(t) = \chi(t) + \gamma\dot\xi(t)\ddot\xi(t),$$
$$\dot\chi(t) = -\gamma\hat\theta(t)\dot\xi^2(t) - \gamma\ddot\xi^2(t).$$

## Multi-Sinusoidal signal

Consider the measurable signal

$$y(t) = A_0 + \sum_{i=1}^{k} A_i \sin(\omega_i t + \phi_i). \tag{13}$$

Signal generator for $k = 2$




where $\theta_1 = -(\omega_1^2 + \omega_2^2)$, $\theta_2 = -\omega_1^2\omega_2^2$.

## Multi-Sinusoidal signal

The signal (13) can be generated by the following differential equation

$$p(p^2-\theta_1)(p^2-\theta_2)\dots(p^2-\theta_k)y(t)=0, \tag{14}$$

where $p=d/dt$ is the differentiation operator, $\theta_i=-\omega_i^2$, are constant parameters, $i=\overline{1,k}$. Equation (14) can be represented as

$$p^{2k+1}y(t)=\bar\theta_1 p^{2k-1}y(t)+\ldots+\bar\theta_k py(t), \tag{15}$$

where $\bar\theta_i$ can be calculated by the following system

$$\begin{cases} \bar\theta_1 = \theta_1+\theta_2+\dots+\theta_k, \\ \bar\theta_2 = -\theta_1\theta_2-\theta_1\theta_3-\dots-\theta_{k-1}\theta_k, \\ \vdots \\ \bar\theta_k = (-1)^{k+1}\theta_1\theta_2\cdots\theta_k. \end{cases}$$

## General linear filter

Introduce the linear filter

$$\xi(s)=F(s)y(s), \quad F(s)=\frac{\lambda_0^{2k}}{\gamma(s)}, \tag{16}$$

where $\lambda_0>0$, $\gamma(s)=s^{2k}+\gamma_{2k-1}s^{2k-1}+\dots+\gamma_1 s+\gamma_0$ is a Hurwitz polynomial.

Multiplying (15) by $\dfrac{\lambda_0^{2k}}{\gamma(s)}$ with (16) we obtain

$$s^{2k+1}\xi(s)=\bar\theta_1 s^{2k-1}\xi(s)+\ldots+\bar\theta_k s\xi(s).$$

## Regression model

After the inverse Laplace transformation for the filter (16) and the input signal $y(t)$ we get the relation

$$\xi^{(2k+1)}(t) = \Omega^T(t)\bar{\Theta} + \varepsilon(t),$$

where $\Omega(t)$ is a regressor of functions $\xi^{(j)}(t)$

$$\Omega^T(t) = \begin{bmatrix}\xi^{(2k-1)}(t) & \dots & \xi^{(3)}(t) & \xi^{(1)}(t)\end{bmatrix},$$

$\bar{\Theta}$ is a vector of unknown parameters depending on frequencies

$$\bar{\Theta}^T = \begin{bmatrix}\bar{\theta}_1 & \dots & \bar{\theta}_{k-1} & \bar{\theta}_k\end{bmatrix}.$$

## Adaptive Frequency Estimation

The update law

$$\hat{\omega}_i = \sqrt{\left|\hat{\theta}_i\right|}, \tag{17}$$

where estimates $\theta_i$ calculated using $\hat{\bar{\theta}}_i$ that are elements of a vector $\hat{\bar{\Theta}}$:

$$\hat{\bar{\Theta}} = \Upsilon(t) + K\Omega(t)\xi^{(2k)}(t), \tag{18}$$

$$\dot{\Upsilon}(t) = -K\Omega(t)\Omega^T(t)\hat{\bar{\Theta}}(t) - K\dot{\Omega}(t)\xi^{(2k)}(t). \tag{19}$$

where $K = \text{diag}\{k_i > 0, i = \overline{1,k}\}$, guarantees that the estimation error $\tilde{\omega}_i = \omega_i - \hat{\omega}_i$ exponentially converges to zero:

$$|\tilde{\omega}_i(t)| \leq \rho_1 e^{-\beta_1 t}, \quad \rho_1, \beta_1 > 0, \quad \forall t \geq 0. \tag{20}$$

## Harmonics observer

For the variable $\xi(t)$ we have

$$\xi(t) = \xi_0 + \xi_1(t) + \xi_2(t) + \cdots + \xi_k(t). \tag{21}$$

After differentiation (21) $2k$ times, we obtain two systems of $k$ linear equations:

$$\begin{cases} \xi^{(1)}(t) = \dot\xi_1(t) + \dot\xi_2(t) + \cdots + \dot\xi_k(t), \\ \xi^{(3)}(t) = \theta_1\dot\xi_1(t) + \theta_2\dot\xi_2(t) + \cdots + \theta_k\dot\xi_k(t), \\ \vdots \\ \xi^{(2k-1)}(t) = \theta_1^{k-1}\dot\xi_1(t) + \cdots + \theta_k^{k-1}\dot\xi_k(t), \end{cases}$$

and

$$\begin{cases} \xi^{(2)}(t) = \theta_1\xi_1(t) + \theta_2\xi_2(t) + \cdots + \theta_k\xi_k(t), \\ \xi^{(4)}(t) = \theta_1^2\xi_1(t) + \theta_2^2\xi_2(t) + \cdots + \theta_k^2\xi_k(t), \\ \vdots \\ \xi^{(2k)}(t) = \theta_1^k\xi_1(t) + \theta_2^k\xi_2(t) + \cdots + \theta_k^k\xi_k(t). \end{cases} \tag{22}$$

## Harmonics observer

From (21) and (22) we get the realizable estimation scheme for variables $\xi_0$ and $\xi_i(t)$

$$\begin{bmatrix} \hat\xi_1(t) \\ \hat\xi_2(t) \\ \vdots \\ \hat\xi_k(t) \end{bmatrix} = \begin{bmatrix} \hat\theta_1 & \cdots & \hat\theta_k \\ \hat\theta_1^2 & \cdots & \hat\theta_k^2 \\ \vdots & \ddots & \vdots \\ \hat\theta_1^k & \cdots & \hat\theta_k^k \end{bmatrix}^{-1} \begin{bmatrix} \xi^{(2)}(t) \\ \xi^{(4)}(t) \\ \vdots \\ \xi^{(2k)}(t) \end{bmatrix},$$

and

$$\hat\xi_0 = \xi(t) - \sum_{i=1}^{k} \hat\xi_i(t).$$

## Parameters estimation

The estimates of the amplitudes and phases

$$\hat{A}_i(t) = \frac{\hat{\gamma}_{\xi i}(t)}{\hat{L}_{\xi i}(t)}, \quad \hat{\phi}_i(t) = \left(-\hat{\varphi}_{\xi i}(t) + \hat{\phi}_{\xi i}(t)\right) \bmod 2\pi,$$

where

$$\hat{\gamma}_{\xi i}(t) = \sqrt{\hat{\xi}_i^2(t) + \left(\frac{\hat{\dot{\xi}}_i(t)}{\hat{\omega}_i(t)}\right)^2},$$

$$\hat{\phi}_{\xi i}(t) = \left(\operatorname{sign}\left(\hat{\xi}_i(t)\right)\arccos\left(\frac{\hat{\dot{\xi}}_i(t)}{\hat{\gamma}_{\xi i}(t)\hat{\omega}_i(t)}\right) - \hat{\omega}_i(t)t\right) \bmod 2\pi,$$

$\hat{L}_{\xi i}(t)$ and $\hat{\varphi}_{\xi i}(t)$ can be obtained from filter frequency response

$$\hat{L}_{\xi i}(t) = |F(j\omega)|_{\omega=\hat{\omega}_i}, \quad \hat{\varphi}_{\xi i}(t) = \arg F(j\omega)|_{\omega=\hat{\omega}_i}.$$

## Dynamic Regressor Extension and Mixing

Consider the regression model

$$\psi(t) = \theta^T \varphi(t), \tag{23}$$

where $\psi(t) \in \mathbb{R}$ is the regressand, $\theta \in \mathbb{R}^n$ is the constant vector of unknown parameters, $\varphi(t) \in \mathbb{R}^n$ is the regressor.

Consider two linear operators

- The stable LTI filter. For example, we can choose exponentially stable LTI filters
$$H_l(p) = \frac{\lambda_l}{p + \lambda_l}, \tag{24}$$
where $p = \frac{d}{dt}$, $\lambda_l \in \mathbb{R}_+$, $l = \overline{1, n}$.
- The delay operator
$$[H_l(\cdot)](t) = (\cdot)(t - d_l), \tag{25}$$
where $d_l > 0$ is a delay.

Let us choose delay operator and define the filtered signals

$$\phi_{f,l}(t) = \phi(t - d_l), \tag{26}$$

$$\psi_{f,l}(t) = \psi(t - d_l). \tag{27}$$

Combine (26)–(27) and signals $\phi(t)$, $\psi(t)$ as follows

$$\Phi_e(t) = \begin{bmatrix} \phi^\top(t) \\ \phi^\top_{f,1}(t) \\ \vdots \\ \phi^\top_{f,n-1}(t) \end{bmatrix}, \quad \Psi_e(t) = \begin{bmatrix} \psi(t) \\ \psi_{f,1}(t) \\ \vdots \\ \psi_{f,n-1}(t) \end{bmatrix}, \tag{28}$$

where $\Phi(t) \in \mathbb{R}^{n\times n}$, $\Psi(t) \in \mathbb{R}^{n\times 1}$.

Defining

$$\zeta(t) = \det\{\Phi(t)\}, \tag{29}$$

$$\xi(t) = \mathrm{adj}\{\Phi(t)\}\Psi(t), \tag{30}$$

where $\det\{\Phi(t)\}$ is the determinant and $\mathrm{adj}\{\Phi(t)\}$ is the adjugate of matrix $\Phi(t)$, we obtain a set of $n$ equations of the form

$$\xi_l(t) = \zeta(t)\theta_l, \quad l = \overline{1,n}. \tag{31}$$

In the obtained first order regression models (31) we can identify parameters $\theta_l$ separately.

The standard gradient method can be used for identification of the obtained models with scalar regressor and parameter

$$\dot{\hat{\theta}}_l(t) = \gamma_d \zeta(t) \left( \xi_l(t) - \zeta(t)\hat{\theta}_l(t) \right), \tag{32}$$

where $\gamma_d \in \mathbb{R}_+$.

From (31) and (32) we can write

$$\dot{\tilde{\theta}}_l(t) = -\gamma_d \zeta^2(t)\tilde{\theta}_l(t). \tag{33}$$

Solving this differential equation we obtain

$$\tilde{\theta}_l(t) = \tilde{\theta}_l(0) \exp\left( -\gamma_d \int_0^t \zeta^2(\tau) d\tau \right). \tag{34}$$

If $\zeta(t)$ is bounded and not square-integrable function, *i.e.*

$$\zeta(t) \notin \mathcal{L}^2 \leftrightarrow \int_0^\infty \zeta^2(\tau) d\tau = \infty, \tag{35}$$

then (32) provides convergence of the estimation error to zero, *i.e.*

$$\lim_{t\to\infty} \left\| \theta_l - \hat{\theta}_l(t) \right\| = 0. \tag{36}$$

For exponential convergence, the following inequality should hold

$$\int_0^t \zeta^2(\tau) d\tau \geq Dt, \tag{37}$$

where $D \in \mathbb{R}_+$.

Time-Delayed Control Systems

# Stabilization problems of the time-delay systems

**Anton Pyrkin**

## Outline

Introduction

Tsypkin's criterion of stability

Smith predictor

State-feedback predictor

Output-feedback predictor

## Time-delay systems

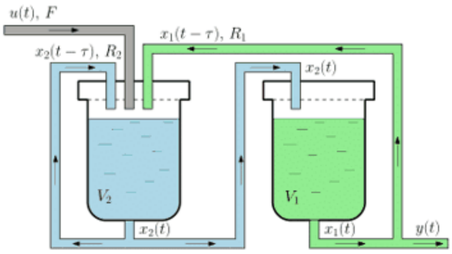
Time-delay systems can be separated to three classes

- plants with input delay
- plants with state delay
- plants with output delay
- plants with several delays

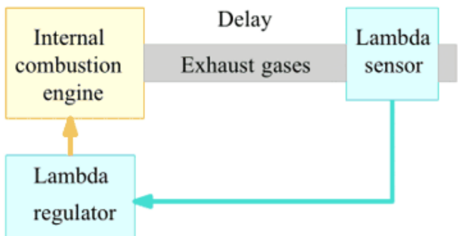
The most complicated and popular in literature are systems with input delay and with input and state delays.

## Technical systems with time delays

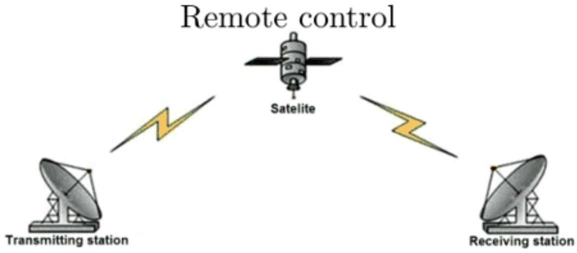
Chemical reactor




Combustion engine



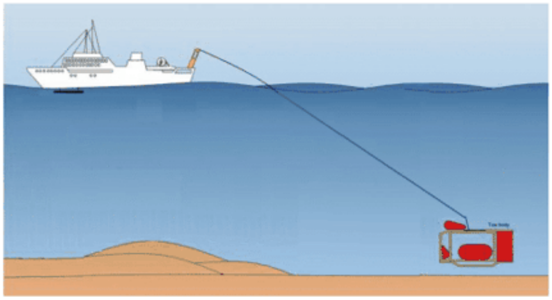

Remote control

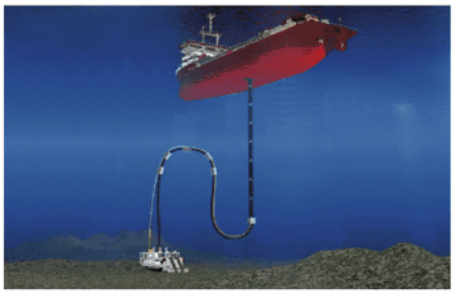


## Systems with delays and external disturbances

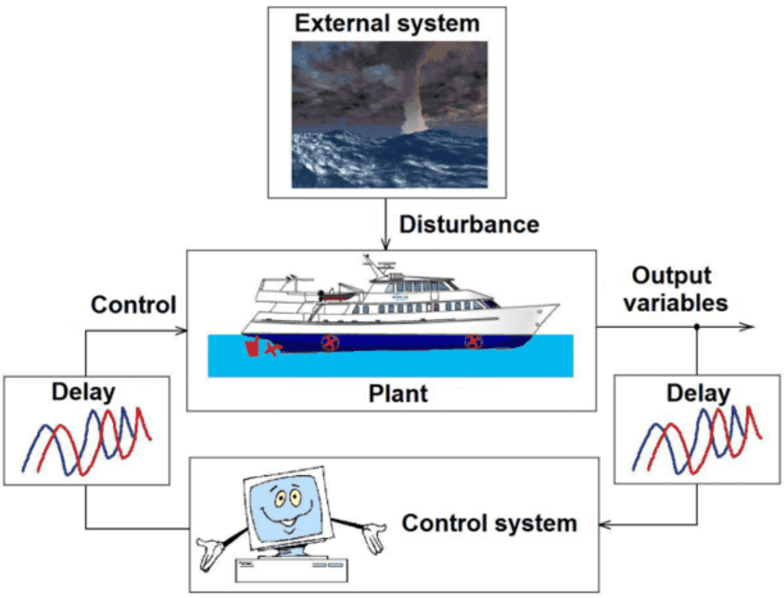
Towing of an underwater vehicle

Extraction of nodules

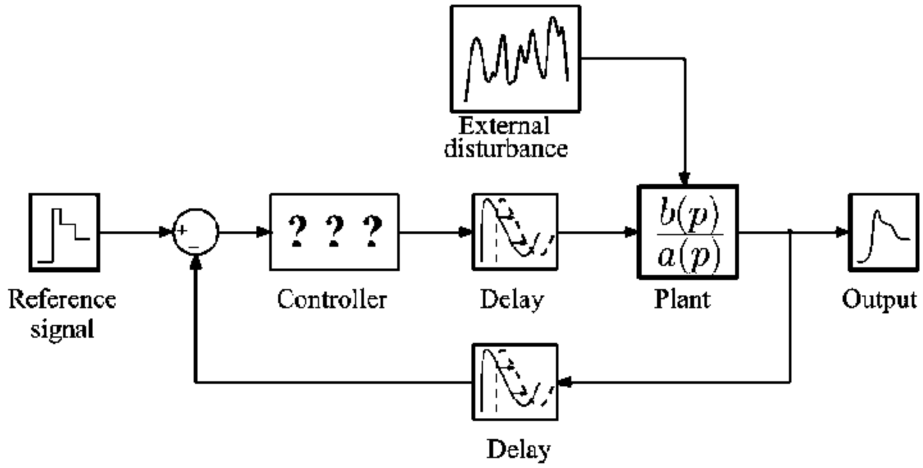




## Present-day view at the problem

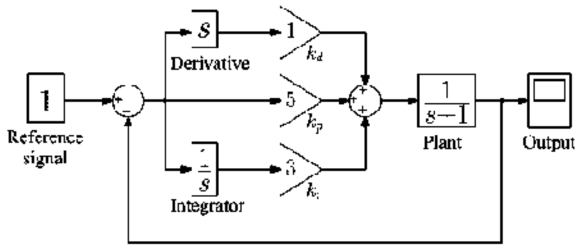


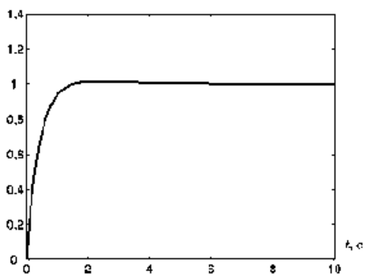
## General problem formulation




Closed-loop system with delays

## Closed-loop system with PID-controller

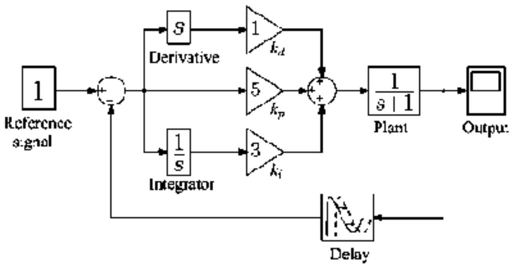


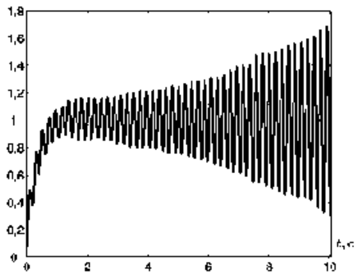
Structural scheme of the system

Transients for the output

## Closed-loop system with delay




Structural scheme of the system




Transients for the output

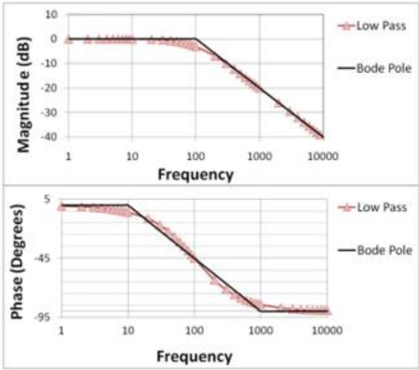
## Basic control approaches

- Tsypkin's criterion of stability
- Smith predictor
- Predictor for unstable systems

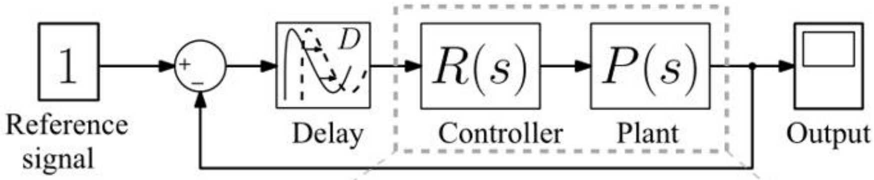
## The first work devoted to the time-delayed systems

Tsypkin Y.Z. Stability of systems with retarding ferd-back // Avtomat. i Telemekh., 1946, V. 7, N. 2-3, P. 107–129.

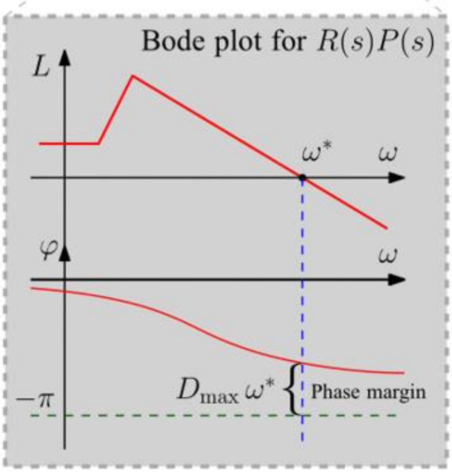
This approach using Bode magnitude and phase plots and Nyquist stability criterion allows to define the maximum delay for which the closed-loop system keeps stability.




## Maximum allowable delay in the closed loop




Transfer function of the delay
$W_D(s) = e^{-Ds}$



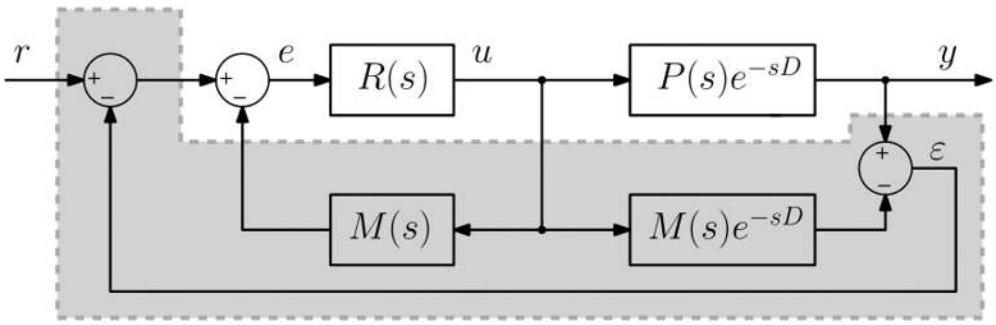
## Smith predictor

Smith predictor is a special structure of the controller proposed by Otto Smith in 1957.

- Smith O.J.M., Closer control of loops with dead time // Chem. Eng. Prog., 1959, N. 53, P. 217–219.
- Smith O.J.M., A controller to overcome dead time // ISA, 1959, V. 6, P. 28–33.

> The main goal of the Smith predictor is to predict which signal will appear before it will happen.

## Smith predictor




Control system supplied with Smith predictor

$M(s)$ is a model of the plant
$e^{-sD}$ is a transfer function of the delay
$R(s)$ is a structure of the nominal controller
$P(s)e^{-sD}$ is a transfer function of the plant with the input delay

## Smith predictor

Assume that the model of the plan is ideal, i.e. $M(s) = P(s)$. Then the error between real output and output estimate will be zero ($\varepsilon = 0$). Thus, we have

$$y = Pe^{-sD}\left(\frac{R}{1+RM}\right)r = \left(\frac{PR}{1+RP}e^{-sD}\right)r. \tag{1}$$

The term $\left(\frac{PR}{1+RP}\right)$ is a transfer function of the closed-loop system without delay.

It means that the delay does not exist in the feedback loop and does not affect the stability and performance of the closed-loop system. In other words controller does the job independently on the time delay. The delay exist only in a numerator of the transfer function that means the output after regulation is delayed.

## Smith predictor

Consider the Smith predictor without assumption $\varepsilon = 0$. In this case the model of the closed-loop system will be

$$y = Pe^{-sD}R\left(r - \varepsilon - Mu\right),\ \varepsilon = y - Me^{-sD}u,$$
$$y = Pe^{-sD}u, \tag{2}$$

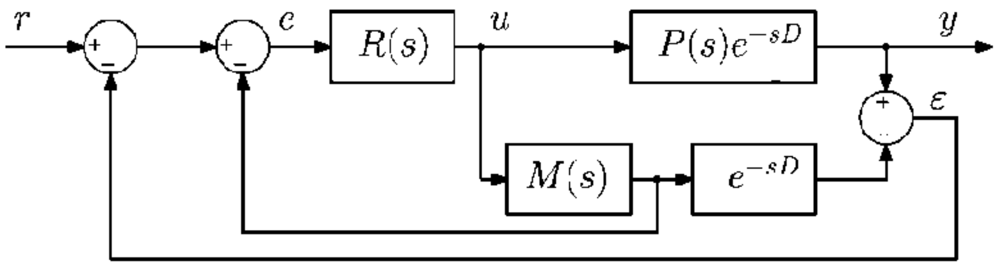
hence

$$y = \left[\frac{PR}{1+RM+R(P-M)e^{-sD}}\right]e^{-sD}r. \tag{3}$$

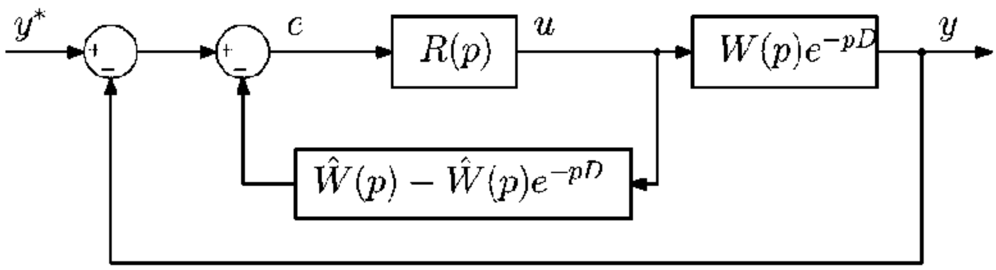
One can see that the error $M - P$ converges to zero if the model is precise, and the exponential term in denominator associated with the delay disappears (in square brackets (3)).

## Modified Smith predictor

Using topological transformations one can get several equivalent structures of Smith predictor.

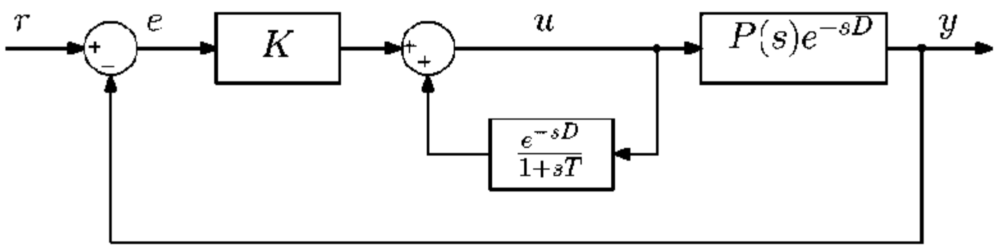



## Modified Smith predictor



## Modified Smith predictor

Predictable proportional-integral controller (PPI-controller) is a modified Smith predictor which is widely utilized in automatic control. Its structure presented on the figure below




## Remarks

Tsypkin's approach and Smith predictor are effective only for linear **stable** systems with known parameters.

The closed-loop system is very sensitive to accuracy of model. Parametric disturbances can be reason of an instability.

## Problem formulation

Consider a linear plant

$$\dot{x}(t) = Ax(t) + Bu(t - D), \tag{4}$$

where $x \in \mathbb{R}^n$ is a state vector, the pair $(A, B)$ is completely controllable, and control $u(t)$ is delayed on $D$ seconds.

The trivial controller for the system (4) may be constructed in the form

$$u(t - D) = Kx(t), \tag{5}$$

where the vector $K$ guaranties that the matrix $A + BK$ is Hurwitz. Hence we have the nominal controller (ideal, although not realizable)

$$u(t) = Kx(t + D). \tag{6}$$

## Control law

However using the solution of (4) for $x(t)$

$$x(t) = e^{At}x(0) + \int_0^t e^{A(t-\tau)}Bu(\tau - D)d\tau \tag{7}$$

we get

$$x(t + D) = e^{AD}x(t) + \int_{t-D}^t e^{A(t-\tau)}Bu(\tau)d\tau, \quad \forall t \geq 0, \tag{8}$$

hence we have the state-feedback controller

$$u(t) = K\left[e^{AD}x(t) + \int_{t-D}^t e^{A(t-\tau)}Bu(\tau)d\tau\right], \quad \forall t \geq 0, \tag{9}$$

which is realizable.

But this controller has an infinite-dimensional term with distributed delay $\int_{t-D}^t e^{A(t-\tau)}Bu(\tau)d\tau$.

## Closed-loop system

Delay has been eliminated in the model of the closed-loop system

$$\dot{x}(t) = (A + BK)x(t), \quad \forall t \geq D. \tag{10}$$

Equation (10) holds only after $D$ seconds. Before $D$ seconds the state of the plant corresponds to the following expression

$$x(t) = e^{At}x(0) + \int_0^t e^{A(t-\tau)}Bu(\tau - D)d\tau, \quad \forall t \in [0, D]. \tag{11}$$

## Original source

Control law (9) was firstly proposed in terms of finite-dimensional systems (Ordinary Differential Equations)

Kwon W.H., Pearson A.E., Feedback stabilization of linear systems with delayed control // IEEE Transactions on Automatic Control, 1980, V. 25, P. 266–269.

Manitius A.Z., Olbrot A.W., Finite spectrum assignment for systems with delays // IEEE Transactions on Automatic Control, 1979, V. 24, P. 541–553.

and reduced approach

Arstein Z., Linear systems with delayed controls: A reduction // IEEE Transactions on Automatic Control, 1982, V. 27, P. 869–879.

Such intuitively clear solution looks simple, however the proof of stability of the closed-loop system is not obvious.

## Backstepping approach

Further we will consider the "backstpping" approach for time-delay systems, which was proposed by Miroslav Krstic

Kristic M., Delay compensation for nonlinear, adaptive, and PDE systems. Birkhauser, 2009, 466 p.

The delay may be presented as partial differential equation (PDE) of the first order

$$U_t(z,t) = U_z(z,t), \quad (12)$$

$$U(D,t) = u(t), \quad (13)$$

where subscripts $z$ and $t$ mean partial derivatives with respect to corresponding arguments.

## PDE model of the delay

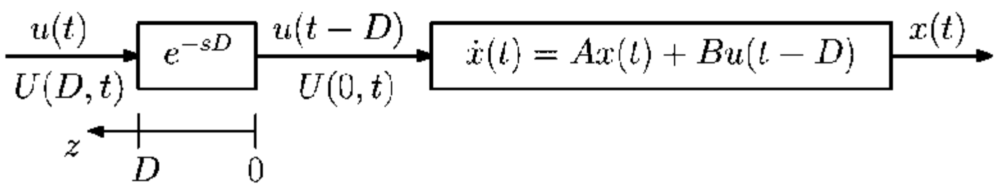
Solution of (12), (13) is

$$U(z,t) = u(t+z-D), \quad (14)$$

where the output of the delay

$$U(0,t) = u(t-D) \quad (15)$$

describes the delayed control signal




Linear plant with the input delay

## Backstepping transformation

Consider the following transformation [1]

$$W(z,t) = U(z,t) - \int_0^z q(z,\zeta)U(\zeta,t)d\zeta - \gamma(z)^T x(t), \qquad (16)$$

which maps the system (4), (12)–(15) to internally stable system

$$\dot{x}(t) = (A + BK)x(t) + BW(0,t), \qquad (17)$$

$$W_t(z,t) = W_z(z,t), \qquad (18)$$

$$W(D,t) = 0. \qquad (19)$$

## Control law

Computation of derivatives $W_t(z,t)$ and $W_z(z,t)$, it is not difficult to find $q(z,\zeta)$ and $\gamma(z)$:

$$q(z,\zeta) = Ke^{A(z-\zeta)}B, \quad \gamma(z)^T = Ke^{Az}. \qquad (20)$$

Substitution $q(z,\zeta)$ and $\gamma(z)$ into (16) together with $z = D$ yields the control law

$$U(D,t) = \int_0^D Ke^{A(D-\zeta)}BU(\zeta,t)d\zeta + Ke^{AD}x(t), \qquad (21)$$

which equals to (9).

## Stability proof

Consider the Lyapunov candidate

$$V(t) = x^T(t)Px(t) + \frac{\gamma}{2}\int_0^D (1+z)W(z,t)^2 dz, \tag{22}$$

where $P = P^T > 0$ is a solution of the Lyapunov equation

$$P(A+BK) + (A-BK)^T P = -Q \tag{23}$$

for any arbitrary $Q + Q^T > 0$ and

$$\gamma = 4\lambda_{max}(PBB^TP)/\lambda_{min}(Q).$$

Then

$$\dot{V}(t) \leq -CV(t),$$

where

$$C = \min\left\{\frac{\lambda_{min}(Q)}{2\lambda_{max}(P)}, \frac{1}{1+D}\right\}.$$

Therefore, the system (4), (9) is an exponentially stable.

## Problem formulation

Consider a linear plant

$$\dot{x}(t) = Ax(t) + Bu(t-D), \quad y(t) = Cx(t), \tag{24}$$

where $x \in \mathbb{R}^n$ is a state vector, $y(t) \in \mathbb{R}$ is a measurable output, and control $u(t)$ which is delayed on $D$ seconds.

It is assumed that pair $(A, B)$ is completely controllable, and pair $(A, C)$ is completely observable.

## State observer

Consider the state observer

$$\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t-D) + L(y(t) - \hat{y}(t)), \quad \hat{y}(t) = C\hat{x}(t), \tag{25}$$

where $L$ makes the matrix $(A - LC)$ Hurwitz.

For the error $\tilde{x}(t) = x(t) - \hat{x}(t)$ and $\tilde{y}(t) = y(t) - \hat{y}(t)$ we have

$$\dot{\tilde{x}}(t) = (A - LC)\tilde{x}(t), \quad \tilde{y}(t) = C\tilde{x}(t), \tag{26}$$

hence it is easy to show that $\tilde{x}(t)$ exponentially converges to zero, i.e. each term of this vector is bounded by decaying exponent.

## Backstepping transformation

Consider backstepping transformation like (16)

$$\begin{aligned} \hat{W}(z,t) &= U(z,t) - Ke^{Az}\hat{x}(t) - K\int_0^z e^{A(z-\zeta)}BU(\zeta,t)d\zeta \\ &\quad + K\int_z^D e^{A(z+D-\zeta)}L\tilde{Y}(\zeta,t)d\zeta, \end{aligned} \tag{27}$$

$$\tilde{Y}(z,t) = \tilde{y}(t+z-D), \tag{28}$$

$$\tilde{Y}_t(z,t) = \tilde{Y}_z(z,t), \tag{29}$$

$$\tilde{Y}(D,t) = \tilde{y}(t). \tag{30}$$

## Control law

Choosing $z = D$ and equating $\hat{W}(D,t)$ to zero in (27) we get a realizable control law

$$u(t) = Ke^{AD}\hat{x}(t) + K\int_{t-D}^{t} e^{A(t-\tau)}Bu(\tau)d\tau, \tag{31}$$

which uses estimates of the state $\hat{x}(t)$.

Substitute in (24) the transformation (27) with $z = 0$:

$$\begin{aligned}\dot{x}(t) &= Ax(t) + BK\hat{x}(t) + K\int_0^D e^{A(D-\zeta)}L\tilde{Y}(\zeta,t)d\zeta + B\hat{W}(0,t)\\ &= (A+BK)x(t) + B\hat{W}(0,t)\\ &\quad - BK\tilde{x}(t) + BK\int_0^D e^{A(D-\zeta)}L\tilde{Y}(\zeta,t)d\zeta\\ &= (A+BK)x(t) + B\hat{W}(0,t) + B\varepsilon(t). \end{aligned} \tag{32}$$

## The closed-loop system

The model of the closed-loop system

$$\dot{x}(t) = (A+BK)x(t) + B\hat{W}(0,t) + B\varepsilon(t), \tag{33}$$

$$y(t) = Cx(t), \tag{34}$$

$$\hat{W}_t(z,t) = \hat{W}_z(z,t), \tag{35}$$

$$\hat{W}(D,t) = 0, \tag{36}$$

where

$$\varepsilon(t) = -K\tilde{x}(t) + K\int_0^D e^{A(D-\zeta)}L\tilde{Y}(\zeta,t)d\zeta$$

is an exponentially decaying function due to exponential convergence to zero of $\tilde{x}(t)$ and, correspondingly, $\tilde{y}(t)$.

Stability of the closed-loop system (33)-(36) may be shown with the Lyapunov function (22) in the similar way.

## Conclusion

Predictor for unstable systems is one of the basic and fundamental solutions that allows to stabilize plants by state or output feedback.

Presented solution is suitable only for linear systems (and additional calculations are necessary for a class of nonlinear systems). The plant parameters are required with good accuracy.

Using this approach it is possible to solve more complicated problems with external disturbances and parametric uncertainties of the plant model.

Adaptive and robust control

ITMO UNIVERSITY

# Adaptive and robust control

Dmitry N. Gerasimov
gerasimovdn@mail.ru

Saint-Petersburg, 2019

## Outline

1. **Introduction to Adaptive and Robust Control**
2. **Lyapunov Functions Method. Short Tutorial**
3. **Simple Example of Adaptive Controller Design**
4. **Simple Example of Robust Controller Design**
5. **Generalized Algorithm of Adaptive and Robust Controller Design**
6. **Standard Error Models**
   - **6.1. Static Error Model. Problem of Identification**
   - **6.2. Dynamic Error Model with Measurable State. State Feedback Adaptive Control**
   - **6.3. Dynamic Error Model with Measurable Output. Output Feedback Adaptive Control**

# 1. Introduction

## Problems and motivation

mathematical models have limited accuracy over the whole range of plants operating

*Aircraft*


*DC motors*

*U*

α

DC motor dynamics

$$\dot{I} = -\frac{R}{L}I - \frac{k_E}{L}\omega + \frac{1}{L}U,$$

$$\dot{\omega} = \frac{k_M}{J}I - \frac{1}{J}M_L,$$

$$\dot{\alpha} = \omega$$

*Spark ignition engines*


Fuel evaporation process dynamics

$$\dot{m}_{ff} = -\frac{1}{T_f}m_{ff} + \frac{K_f}{T_f}m_{fi}$$

$$m_{fc} = m_{ff} + (1 - K_f)m_{fi}$$

*Blood system*

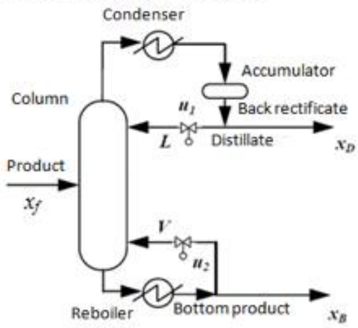

Blood pressure dynamics with delay

$$y(t) = \frac{Ke^{-T_1 s}\left(1 + ae^{-T_2 s}\right)}{1 + \tau s}\left[u(t)\right]$$

*y – deviation of mean arterial pressure from normal*

*u – infusion rate of drug* (nitroprusside)

*Distillation column*




Distillation system dynamics

$$\begin{bmatrix} x_d \\ x_b \end{bmatrix} = G(s)\left(I + W(s)\Delta\right)\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} + G_d(s)\left[x_f\right]$$

$$m_{fc} = m_{ff} + (1 - K_f)m_{fi}$$

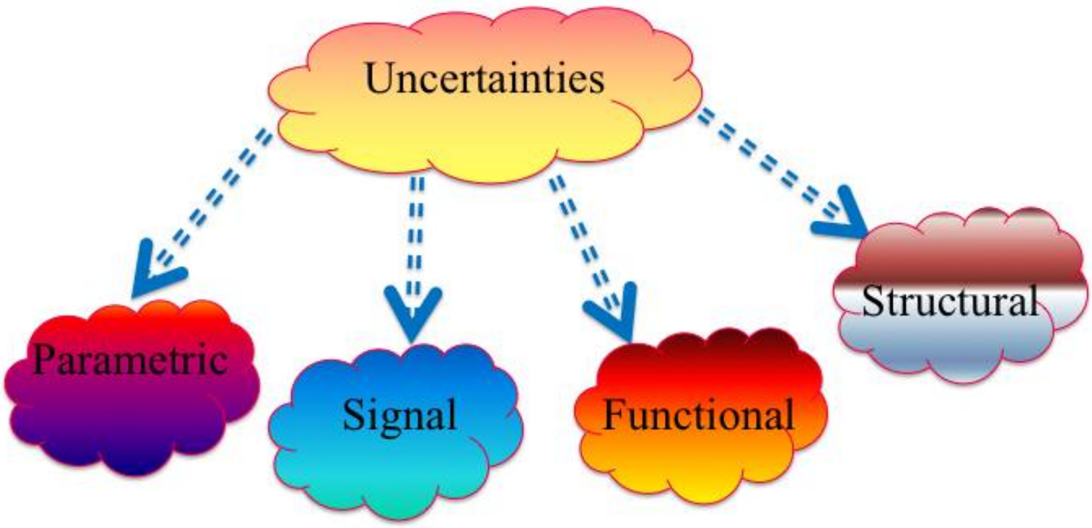
In this context, the approaches of control theory that can come up with the problems of plants uncertainties are of special interest.

Can the control system choose the correct control to improve the performance of the plant operating in presence of uncertainties?


How to design an adaptive control?

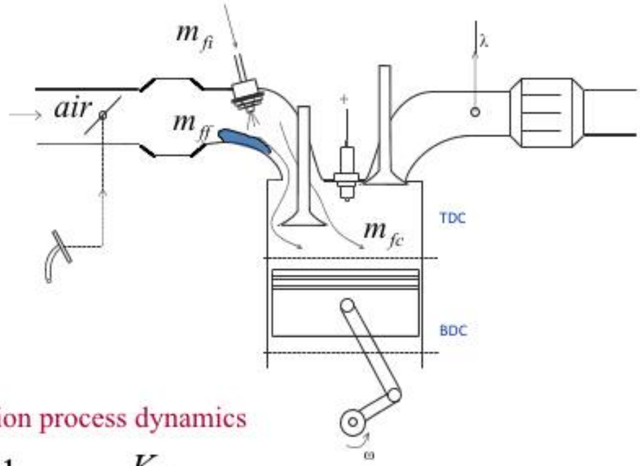
**Definitions, clarifications:**

1. Model with uncertainties that is the potential basis for controller design belongs to some **class of models** and is called **nominal.**
2. Characteristics of the nominal model are called **nominal.**
3. Uncertainties – unknown or not known precisely characteristics, structure or parameters of the plant .
4. Uncertainties of the plant ≡ uncertainties of the model.



**Parametric uncertainties** imply that the parameters of the plant model are constant and unknown.

*Spark ignition engines*




Fuel evaporation process dynamics

$$\dot{m}_{ff} = -\frac{1}{T_f} m_{ff} + \frac{K_f}{T_f} m_{fi}$$

$$m_{fc} = m_{ff} + (1 - K_f) m_{fi}$$

**Signal uncertainties** imply that the plant model contains unknown functions of time.

*DC motors*

DC motor dynamics

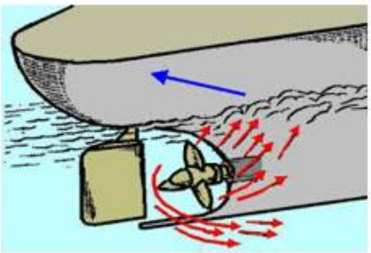
$$\dot{I} = -\frac{R}{L} I - \frac{k_E}{L}\omega + \frac{1}{L} U,$$

$$\dot{\omega} = \frac{k_M}{J} I - \frac{1}{J} M_L,$$

$$\dot{\alpha} = \omega$$

$$R = R(temperature) = R(time)$$

**Functional uncertainties** imply that plant model contains unknown functions of state.



Tail-shaft dynamics equation

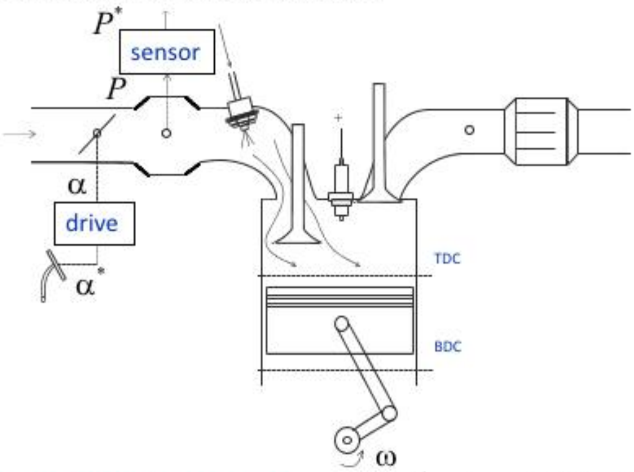
$$J\dot{\omega} = M - M_V$$

$M$ is the engine effective torque

$M_V$ is the viscous friction

$$M_V = M_V(\omega) \approx c_0 + c_1\omega + c_2\omega^2$$

**Structural uncertainties** imply that the plant model contains unknown structures.




Manifold air pressure dynamics equation: $\dot{\boldsymbol{P}} + k_1\eta_c(\omega)\boldsymbol{P} = k_2\eta_t(\boldsymbol{P})\varphi_1(\boldsymbol{P})\varphi_2(\boldsymbol{\alpha})$

Pressure sensor dynamics: $\dot{\boldsymbol{P}}^* = -a\boldsymbol{P}^* + b\boldsymbol{P}$

Throttle drive dynamics: $\dot{\alpha} = -c\alpha + d\alpha^*$

**Definitions:**

Adaptive and robust control are the controls providing desired performance of the plant operating in presence of uncertainties

1. Adaptive control implies the compensation of uncertainties.
2. Robust control does not imply the compensation of uncertainties, but using high gain control.

# 2. Lyapunov Functions Method. Short tutorial

Universal approach of stability analysis for autonomous plants

$$\dot{x} = f(x), \quad x(0), \tag{2.1}$$

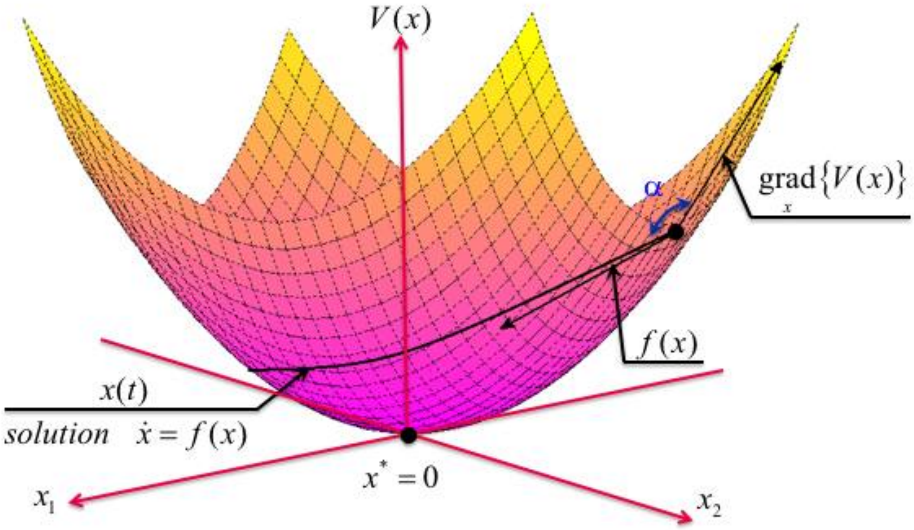
at equilibrium $x^*$, where $x \in R^n$ is the state vector, $f \in R^n$ is the continuous nonlinear mapping.

**Lyapunov functions** $V(x)$ **:**

1. $V(x)$ is monotonic ;
2. $V(x) > 0,\ if\ \|x\| \neq 0$ ,
   $V(0) = 0$ ;
3. $V(x) \in C^1$ (*continuous and differentiable*) .

**Time derivative of Lyapunov function in amount of (1):**

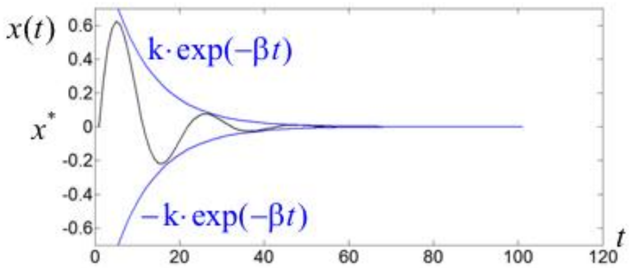
$$\dot{V}(x)=\frac{\partial V(x)}{\partial x}\dot{x}=\frac{\partial V(x)}{\partial x}f(x)=\operatorname{grad}_x\{V(x)\}f(x)=\left\|\operatorname{grad}_x\{V(x)\}\right\|\left\|f(x)\right\|\cos\alpha$$




**Stability criterias:**

1. If $\dot{V}(x)\le 0$, then the equilibrium $x^*=0$ is Lyapunov stable;
2. If $\dot{V}(x)<0$, then the equilibrium $x^*=0$ is asymptotically stable;
3. If $\dot{V}(x)\le -\beta V(x),\ \beta>0$, then the equilibrium $x^*=0$ is exponentially stable;

$$\dot{V}(x)\le -\beta V(x) \quad\Rightarrow\quad V(x)\le \exp(-\beta t)V(0)$$



**Examples of Lyapunov functions:**

1. Linear system

$$\dot{x} = Ax, \quad x(0) \tag{2.2}$$

where $A$ is the time-invariant matrix.

Lyapunov function candidate

$$V(x) = x^T P x, \tag{2.3}$$

where $P = P^T \succ 0$ is the time-invariant matrix.

$$\dot{V}(x) = \dot{x}^T P x + x^T P \dot{x} = x^T A^T P x + x^T P A x =$$
$$= x^T \left(A^T P + PA\right) x = -x^T Q x < 0$$

Conclusion: If there exists $P = P^T \succ 0$ such that

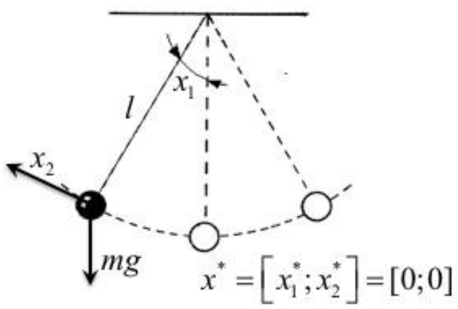
$$A^T P + PA = -Q, \tag{2.4}$$

where $Q = Q^T \succ 0$, system (2) is asymptotically stable.

2. Pendulum

$$\dot{x}_1 = x_2,$$
$$\dot{x}_2 = -gl\sin(x_1) - \frac{k}{m}x_2 \tag{2.5}$$

where $g$ is the gravity acceleration, $l$ is the length of rod, $k$ is the friction coefficient.




Lyapunov function candidate: **sum of potential and kinetic energy**

$$V(x) = mg\left(1 - \cos(x_1)\right)l + \frac{mx_2^2}{2}. \tag{2.6}$$

Time derivative:

$$\dot{V}(x) = mg\sin(x_1)\dot{x}_1 l + mx_2\dot{x}_2 = mg\sin(x_1)x_2 l - mx_2 gl\sin(x_1) - mx_2^2.$$

or

$$\dot{V}(x) = -mx_2^2 < 0 \tag{2.7}$$

Conclusion: pendulum is asymptotically stable at the equilibrium $x^* = [0;0]$.

# 3. Simple Example of Adaptive Controller Design

**Motivation**

**Problem statement:**

**Plant:**

$$\dot{x} = \theta x + u, \tag{3.1}$$

where $x$ is the scalar state, $u$ is the control, $\theta$ is the known parameter.

**Objective** is to design a control providing the following limiting equality:

$$\lim_{t\to\infty} x = 0. \tag{3.2}$$

**Solution:**

$$u = -\theta x - \lambda x, \tag{3.3}$$

where $\lambda$ is the positive constant parameter.

$$u = -\theta x - \lambda x, \quad \dot{x} = \theta x + u \quad \Rightarrow \quad \dot{x} = -\lambda x \quad \Rightarrow \quad x(t) = \exp(-\lambda t)x(0). \tag{3.4}$$

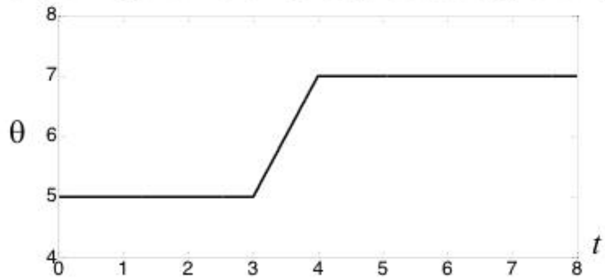
Let us the design parameter is $\lambda = 1$ and plant parameter $\theta = 5$,

i.e.

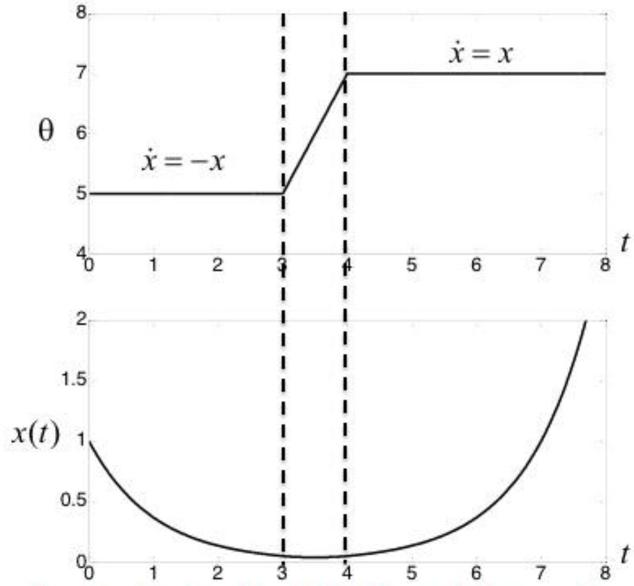
control: $u = -6x$

system $\dot{x} = -x$ is stable.

Now let us imagine, the plant parameter $\theta$ unpredictably changes from 5 to 13:



*Control* : $u = -6x$




**Classical control is not reliable and does not work properly in presence of uncertainties**

**Problem statement of adaptive control:**

**Plant:** (3.5)

$$\dot{x} = \theta x + u,$$

where $\theta$ is the unknown parameter.

**Objective** is to design a control providing the following limiting equality:

$$\lim_{t \to \infty} \left( x_M - x \right) = 0, \quad (3.6)$$

where $x_M$ is the output of reference model

$$\dot{x}_M = -\lambda x_M + \lambda g, \quad (3.7)$$

$g$ is the reference signal, $\lambda$ is the positive parameter responsible for transient time.

**Solution:**

**1.** Let the parameter $\theta$ be known.

Form the error signal $\varepsilon = x_M - x$ and take its derivative in amount of plant and reference model equations:

$$\dot{\varepsilon} = \dot{x}_M - \dot{x} = (-\lambda x_M + \lambda g) - (\theta x + u)$$

Let $\dot{\varepsilon} \triangleq -\lambda\varepsilon = -\lambda x_M + \lambda x \;\Rightarrow\; \varepsilon(t) = \exp(-\lambda t)\varepsilon(0)$. Therefore

$$(-\lambda x_M + \lambda g) - (\theta x + u) = -\lambda x_M + \lambda x$$

$$\Downarrow$$

$$\boxed{u = -\theta x - \lambda x + \lambda g} \qquad (3.8)$$

**Solution:**

**2.** Let the parameter $\theta$ be unknown. Therefore the control

$$u = -\theta x - \lambda x + \lambda g$$

is not implementable. Substitute estimate $\hat{\theta}$ for $\theta$ and obtain implementable adjustable control:

$$\boxed{u = -\hat{\theta} x - \lambda x + \lambda g} \qquad (3.10)$$

Replace (3.10) in the plant equation $\dot{x} = \theta x + u$ :

$$\dot{x} = \theta x - \hat{\theta} x - \lambda x + \lambda g, \qquad (3.11)$$

Take the derivative of the error

$$\dot{\varepsilon} = \dot{x}_M - \dot{x} = (-\lambda x_M + \lambda g) - \left(\theta x - \hat{\theta} x - \lambda x + \lambda g\right)$$

*Signal Error Model* $$\boxed{\dot{\varepsilon} = -\lambda\varepsilon - \tilde{\theta} x,} \qquad (3.12)$$

where $\tilde{\theta} = \theta - \hat{\theta}$ is the parametric error.

**Solution:**

**3.** Let us choose the algorithm generating estimate $\hat{\theta}$ :

$$\dot{\hat{\theta}} = \Omega(t) \tag{3.13}$$

where $\Omega(t)$ is implementable (measurable) function.

Taking into account that $\tilde{\theta} = \theta - \hat{\theta}$ and

$$\dot{\tilde{\theta}} = -\dot{\hat{\theta}}$$

we get

*Parametric Error Model*

$$\dot{\tilde{\theta}} = -\Omega(t) \tag{3.14}$$

**How to choose the function $\Omega(t)$???**

**Solution:**

**4.** Models

*Signal Error Model* $\dot{\varepsilon} = -\lambda\varepsilon - \tilde{\theta}x,$ (3.12)

*Parametric Error Model* $\dot{\tilde{\theta}} = -\Omega(t)$ (3.14)

Choose the Lyapunov function candidate

$$V(\varepsilon, \tilde{\theta}) = \frac{1}{2}\varepsilon^2 + \frac{1}{2\gamma}\tilde{\theta}^2, \qquad \gamma > 0 \tag{3.15}$$

and take its time derivative using (18) and (20):

$$\dot{V}(\varepsilon, \tilde{\theta}) = \varepsilon\dot{\varepsilon} + \frac{1}{\gamma}\tilde{\theta}\dot{\tilde{\theta}} = -\lambda\varepsilon^2 - \tilde{\theta}x\varepsilon - \frac{1}{\gamma}\tilde{\theta}\Omega(t)$$

If $\Omega(t) = -\gamma x\varepsilon$ then $\dot{V}(\varepsilon, \tilde{\theta}) = -\lambda\varepsilon^2 < 0$

$$\dot{\hat{\theta}} = -\gamma x\varepsilon \tag{3.16}$$

## Summary

**Adjustable controller:**

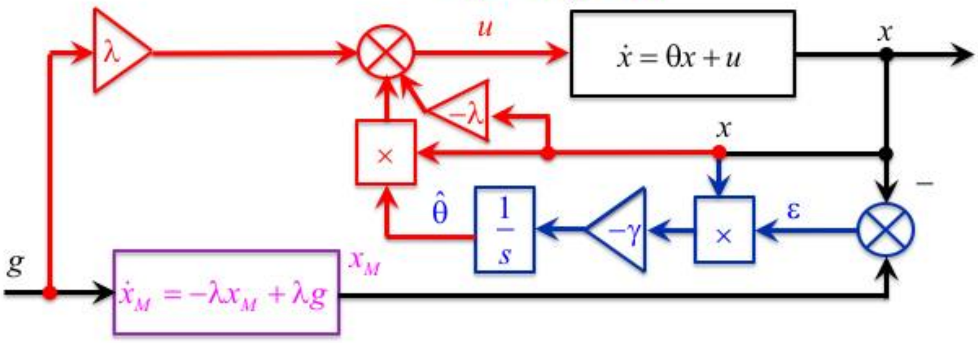
$$u = -\hat{\theta}x - \lambda x + \lambda g \tag{3.10}$$

**Adaptation algorithm:**

$$\dot{\hat{\theta}} = -\gamma x \varepsilon \tag{3.16}$$

with $\varepsilon = x_M - x$ and reference model

$$\dot{x}_M = -\lambda x_M + \lambda g. \tag{3.7}$$




## Summary

**Properties of the closed-loop system:**

1. All signals in the system are bounded;
2. Control error $\varepsilon = x_M - x$ asymptotically tends to zero;
3. Parametric error $\tilde{\theta} = \theta - \hat{\theta}$ in general case tends to a constant;

$$V(\varepsilon, \tilde{\theta}) = \frac{1}{2}\varepsilon^2 + \frac{1}{2\gamma}\tilde{\theta}^2,$$

$$\dot{V}(\varepsilon, \tilde{\theta}) = -\lambda\varepsilon^2 < 0$$

4. There is an optimal adaptation gain $\gamma$ corresponding the fastest parametrical convergence;

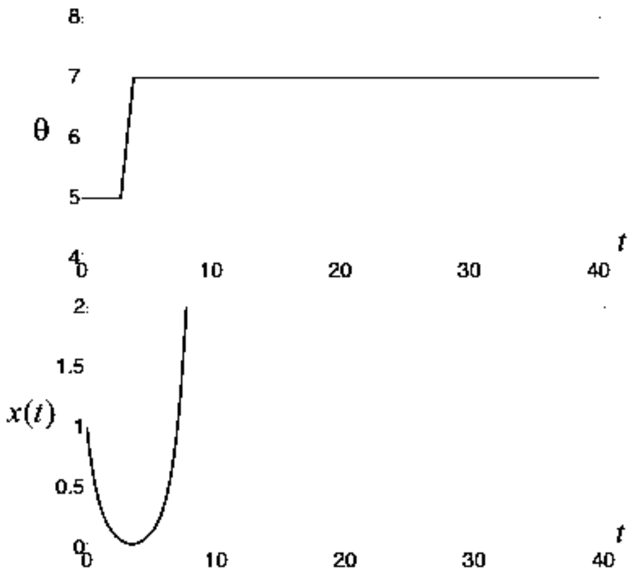
5. There can be parametric drift phenomena in presence of noise, i.e., if

$$\dot{x} = \theta x + u + \delta,$$

where $\delta$ is bounded disturbance, $\hat{\theta} \to \infty$ .

**Example: Classical stabilizing control for unstable plant** $\dot{x} = 5x + u$

$$u = -6x$$

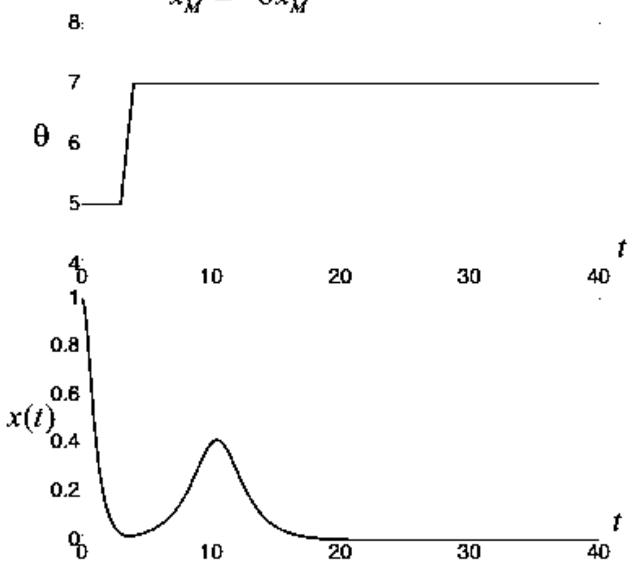



**Example: Adaptive stabilizing control for unstable plant** $\dot{x} = 5x + u$

$$u = -6x - \hat{\theta}x,$$

$$\dot{\hat{\theta}} = -2x\varepsilon, \qquad \varepsilon = x_M - x,$$

$$\dot{x}_M = -6x_M$$

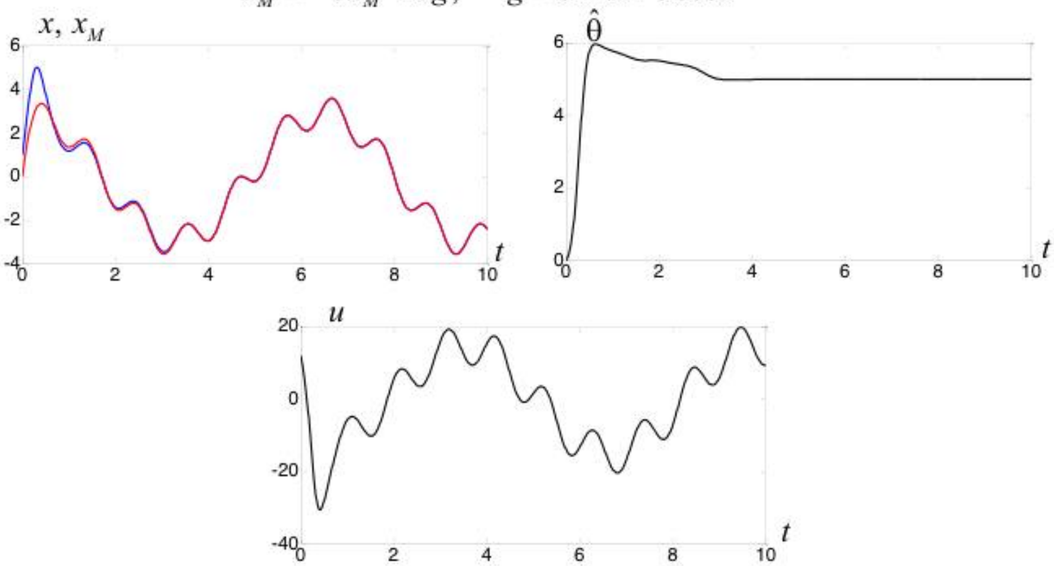


**Example: Adaptive tracking control for unstable plant** $\dot{x} = 5x + u$

$$u = -6x - \hat{\theta}x + 6g,$$

$$\dot{\hat{\theta}} = -2x\varepsilon, \quad \varepsilon = x_M - x,$$

$$\dot{x}_M = -6x_M + 6g, \quad g = \sin 6t + 3\cos t$$




# 4. Simple Example of Robust Controller Design

**Problem statement of adaptive control:**

**Plant:**

$$\dot{x} = \theta x + u + \delta, \qquad |\delta| \le \bar{\delta} \tag{4.1}$$

where $\theta$ is the unknown parameter, $\delta(t)$ is unpredictable bounded noise.

**Objective** is to design a control providing the following inequality:

$$|x_M(t) - x(t)| \le \Delta \;\text{ for } any\; t \ge T, \tag{4.2}$$

where $x_M$ is the output of reference model

$$\dot{x}_M = -\lambda x_M + \lambda g, \tag{4.3}$$

$g$ is the reference signal, $\lambda$ is the positive parameter responsible for transient time.

**Solution #1**

**Adjustable controller:**

$$u = -\hat{\theta}x - \lambda x + \lambda g \tag{4.4}$$

**~~Adaptation algorithm:~~** ⟶ **Nonlinear static feedback:**

$$\hat{\theta} = -\gamma x \varepsilon \tag{4.5}$$

with $\varepsilon = x_M - x$ and reference model

$$\dot{x}_M = -\lambda x_M + \lambda g.$$

Substitution of (4.5) into (4.4) gives "high-gain" type controller:

$$u = \gamma x^2 \varepsilon - \lambda x + \lambda g.$$

Then substitute this control into disturbed plant $\dot{x} = \theta x + u + \delta.$

$$\dot{x} = \theta x + \gamma x^2 \varepsilon - \lambda x + \lambda g + \delta.$$

**Solution #1**

Again, take the derivative of the error $\varepsilon = x_M - x$

$$\dot{\varepsilon} = \dot{x}_M - \dot{x} = \left(-\lambda x_M + \lambda g\right) - \left(\theta x + \gamma x^2 \varepsilon - \lambda x + \lambda g + \delta\right)$$

$$\boxed{\dot{\varepsilon} = -\lambda\varepsilon - \theta x - \gamma x^2 \varepsilon - \delta} \tag{4.6}$$

Choose the Lyapunov function candidate

$$V(\varepsilon, \tilde{\theta}) = \frac{1}{2}\varepsilon^2$$

and take its time derivative using (4.6):

$$\dot{V}(\varepsilon) = \varepsilon\dot{\varepsilon} = -\lambda\varepsilon^2 - \theta x\varepsilon - \gamma x^2\varepsilon^2 - \delta\varepsilon = -\frac{\lambda}{2}\varepsilon^2 - \frac{\lambda}{2}\varepsilon^2 - \theta x \varepsilon - \gamma x^2 \varepsilon^2 - \delta\varepsilon =$$

$$= -\frac{\lambda}{2}\varepsilon^2 - \frac{\lambda}{2}\varepsilon^2 - \delta\varepsilon \pm \frac{1}{2\lambda}\delta^2 - \gamma x^2 \varepsilon^2 - \theta x \varepsilon \pm \frac{\theta^2}{4\gamma}$$

## Solution #1

$$\dot V(\varepsilon) = -\frac{\lambda}{2}\varepsilon^2 - \left(\sqrt{\frac{\lambda}{2}}\varepsilon + \sqrt{\frac{1}{2\lambda}}\delta\right)^2 + \frac{1}{2\lambda}\delta^2 - \left(\sqrt{\gamma}x\varepsilon + \frac{\theta}{2\sqrt{\gamma}}\right)^2 + \frac{\theta^2}{4\gamma}$$

$$\dot V(\varepsilon) \le -\frac{\lambda}{2}\varepsilon^2 + \frac{1}{2\lambda}\delta^2 + \frac{\theta^2}{4\gamma}$$

$$\dot V(\varepsilon) \le -\frac{\lambda}{2}\varepsilon^2 + \frac{1}{2\lambda}\bar\delta^2 + \frac{\theta^2}{4\gamma}$$

$$|\delta(t)| \le \bar\delta$$

$$\bar\Delta = \frac{1}{2\lambda}\bar\delta^2 + \frac{\theta^2}{4\gamma}$$

$$\bar\Delta = \frac{1}{2\lambda}\bar\delta^2 + \frac{\theta^2}{4\gamma}$$

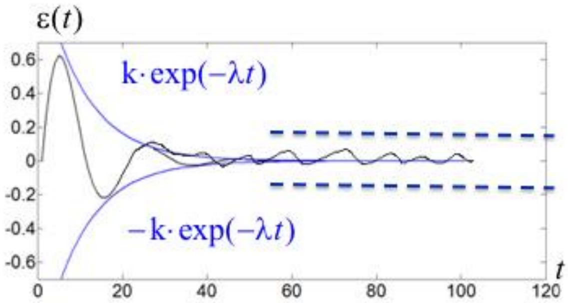
$$\boxed{\dot V(\varepsilon) \le -\lambda V(\varepsilon) + \bar\Delta}$$

$$V(\varepsilon) = \frac{1}{2}\varepsilon^2 \quad (4.7)$$

## Solution #1

$$\dot V(\varepsilon) \le -\lambda V(\varepsilon) + \bar\Delta \qquad \Rightarrow \qquad V(t) \le \exp(-\lambda t)V(0)\left(1 - \frac{\bar\Delta}{\lambda}\right) + \frac{\bar\Delta}{\lambda}V(0)$$

Exponential convergence of $\varepsilon$ to bounded set is proved.

**Solution #1** **Summary**

**Adjustable controller:**

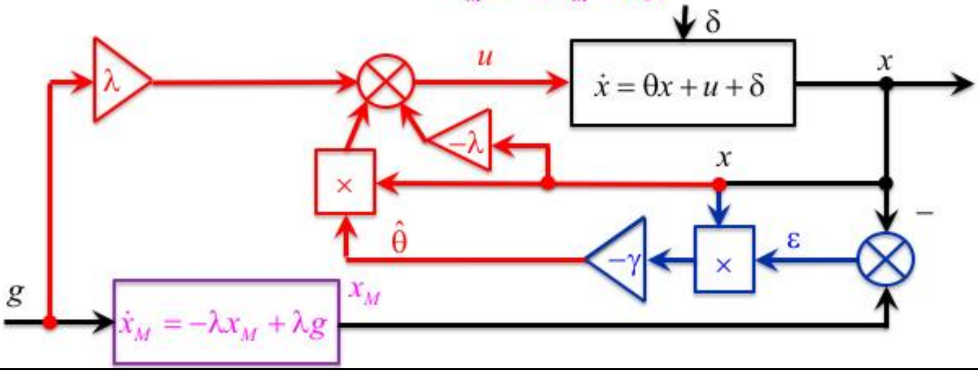
$$u = -\hat{\theta}x - \lambda x + \lambda g \tag{4.4}$$

**Nonlinear static feedback :**

$$\dot{\hat{\theta}} = -\gamma x \varepsilon \tag{4.5}$$

with $\varepsilon = x_M - x$ and reference model

$$\dot{x}_M = -\lambda x_M + \lambda g. \tag{4.3}$$




**Solution #1** **Summary**

**Properties of the closed-loop robust system:**

1. All signals in the system are bounded;
2. Control error $\varepsilon = x_M - x$ exponentially tends to the neighborhood of zero;
3. The radius of neighborhood can be arbitrary reduced by

$\lambda$  *or* $\gamma$ 

$$\dot{V}(\varepsilon) \le -\lambda V(\varepsilon) + \bar{\Delta} \quad \text{where} \quad \bar{\Delta} = \frac{1}{2\lambda}\bar{\delta}^2 + \frac{\theta^2}{4\gamma}$$

4. There is no compensation of uncertainty!

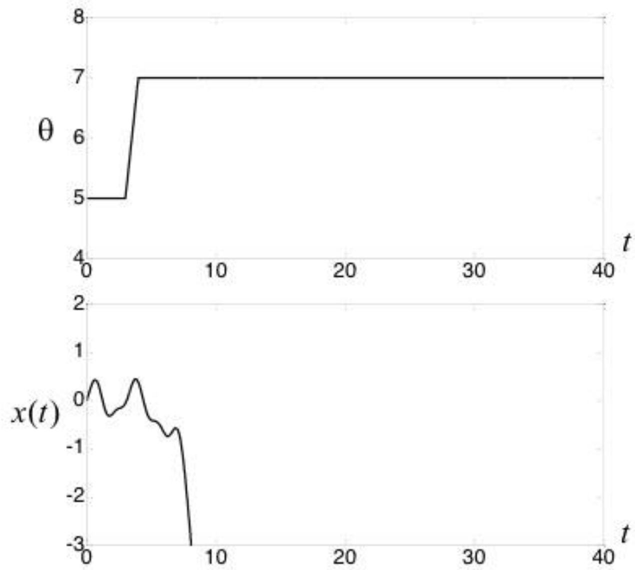
Even, if the plant is not disturbed ( $\delta = \bar{\delta} = 0$ ), the error $\varepsilon = x_M - x$ does not go to zero!

**Example: Classical stabilizing control for unstable plant** $\dot{x} = \theta x + u + \delta$

$u = -6x$

$\theta = 5$

$\delta = 0,5\sin(4t) + 0,75\cos(2t)$



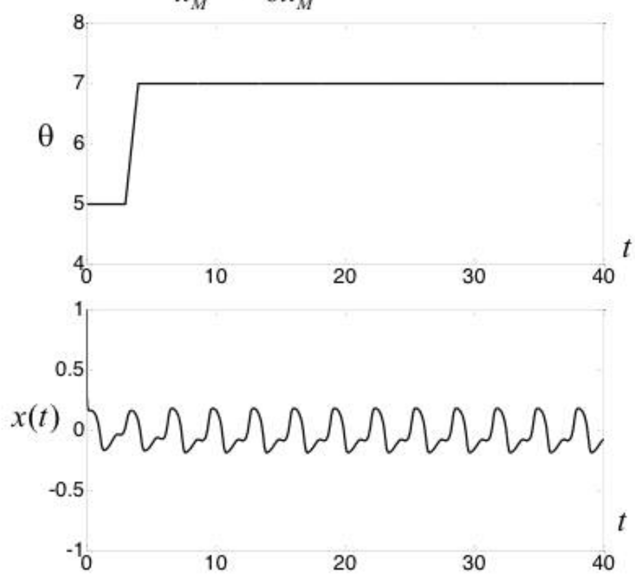

**Example: Robust stabilizing control for unstable plant** $\dot{x} = \theta x + u + \delta$

$$u = -6x - \hat{\theta}x,$$
$$\dot{\hat{\theta}} = -\gamma x\varepsilon, \qquad \varepsilon = x_M - x,$$
$$\dot{x}_M = -6x_M$$

$\theta = 5$

$\delta = 0,5\sin(4t) + 0,75\cos(2t)$

$\gamma = 200$



**Example: Robust tracking control for unstable plant** $\dot{x} = \theta x + u + \delta$

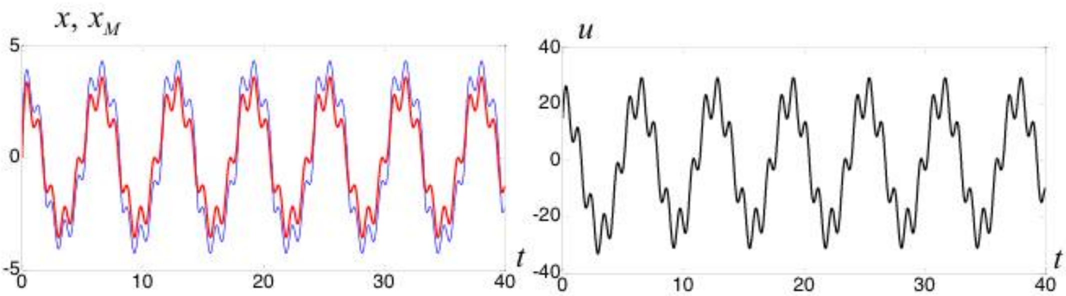
$$u = -6x - \hat{\theta}x + 6g,$$

$\theta = 5$

$$\dot{\hat{\theta}} = -2x\varepsilon, \qquad \varepsilon = x_M - x,$$

$\delta = 0,5\sin(4t) + 0,75\cos(2t)$

$$\dot{x}_M = -6x_M + 6g, \qquad g = \sin 6t + 3\cos t$$




**Adaptive control provides the complete compensation of uncertainties,**

**but can be not reliable under disturbance condition** $\hat{\theta} \to \infty$

**Robust control guarantee the strongest exponential stability,**

**but does not compensate the uncertainties, therefore** $\varepsilon \not\to 0$

**Adaptive control provides the complete compensation of uncertainties,**

**but can be not reliable under disturbance condition** $\hat{\theta} \to \infty$

*trade off* ?

**Robust control guarantee the strongest exponential stability,**

**but does not compensate the uncertainties, therefore** $\varepsilon \not\to 0$

**Solution #2**

**Adjustable controller:**

$$u = -\hat{\theta}x - \lambda x + \lambda g \tag{4.8}$$

**~~Adaptation algorithm:~~** $\longrightarrow$ **Robust modification of AA:**

$$\dot{\hat{\theta}} = -\gamma x\varepsilon - \sigma\hat{\theta} \tag{4.9}$$

where $\sigma$ is a positive feedback gain,

$\varepsilon = x_M - x$, $x_M$ is the output of reference model

$$\dot{x}_M = -\lambda x_M + \lambda g.$$

Then substitute control (31) into disturbed plant $\dot{x} = \theta x + u + \delta$.

$$\dot{x} = \theta x - \hat{\theta}x - \lambda x + \lambda g + \delta.$$

$$\dot{x} = \tilde{\theta}x - \lambda x + \lambda g + \delta. \qquad (\tilde{\theta} = \theta - \hat{\theta})$$

**Solution #2**

Again, form take the derivative of the error $\varepsilon = x_M - x$

$$\dot{\varepsilon} = \dot{x}_M - \dot{x} = (-\lambda x_M + \lambda g) - (\tilde{\theta}x - \lambda x + \lambda g + \delta)$$

*Signal Error Model*

$$\boxed{\dot{\varepsilon} = -\lambda\varepsilon - \tilde{\theta}x - \delta} \tag{4.10}$$

$$\dot{\hat{\theta}} = -\gamma x\varepsilon - \sigma\hat{\theta} \longrightarrow \dot{\tilde{\theta}} = -\dot{\hat{\theta}}$$

*Parametric Error Model*

$$\boxed{\dot{\tilde{\theta}} = \gamma x\varepsilon + \sigma\hat{\theta}} \tag{4.11}$$

Choose the Lyapunov function candidate

$$V(\varepsilon, \tilde{\theta}) = \frac{1}{2}\varepsilon^2 + \frac{1}{2\gamma}\tilde{\theta}^2, \qquad \gamma > 0 \tag{4.12}$$

## Solution #2

Take the time derivative of Lyapunov function using (4.10) and (4.11):

*Signal Error Model* $\dot{\varepsilon} = -\lambda\varepsilon - \tilde{\theta}x - \delta$

*Parametric Error Model* $\dot{\tilde{\theta}} = \gamma x\varepsilon + \sigma\hat{\theta}$

$$\dot{V}(\varepsilon,\tilde{\theta}) = \varepsilon\dot{\varepsilon} + \frac{1}{\gamma}\tilde{\theta}\dot{\tilde{\theta}} = -\lambda\varepsilon^2 - \tilde{\theta}x\varepsilon - \frac{1}{\gamma}\tilde{\theta}\Omega(t)$$

$$\dot{V}(\varepsilon,\tilde{\theta}) = \varepsilon\dot{\varepsilon} + \frac{1}{\gamma}\tilde{\theta}\dot{\tilde{\theta}} = \left(-\lambda\varepsilon^2 - \tilde{\theta}x\varepsilon - \delta\varepsilon\right) + \frac{1}{\gamma}\tilde{\theta}\left(\gamma x\varepsilon + \sigma\hat{\theta}\right)$$

$$\dot{V}(\varepsilon,\tilde{\theta}) = -\lambda\varepsilon^2 - \delta\varepsilon + \frac{\sigma}{\gamma}\tilde{\theta}\hat{\theta}$$

$$\tilde{\theta} = \theta - \hat{\theta}$$

$$\dot{V}(\varepsilon,\tilde{\theta}) = -\lambda\varepsilon^2 - \delta\varepsilon - \frac{\sigma}{\gamma}\tilde{\theta}^2 + \frac{\sigma}{\gamma}\tilde{\theta}\theta$$

## Solution #2

$$\dot{V}(\varepsilon,\tilde{\theta}) = -\frac{\lambda}{2}\varepsilon^2 - \frac{\lambda}{2}\varepsilon^2 - \delta\varepsilon - \frac{\sigma}{2\gamma}\tilde{\theta}^2 - \frac{\sigma}{2\gamma}\tilde{\theta}^2 + \frac{\sigma}{\gamma}\tilde{\theta}\theta$$

$$\dot{V}(\varepsilon,\tilde{\theta}) = -\frac{\lambda}{2}\varepsilon^2 - \frac{\sigma}{2\gamma}\tilde{\theta}^2 - \frac{\lambda}{2}\varepsilon^2 - \delta\varepsilon \pm \frac{1}{2\lambda}\delta^2 - \frac{\sigma}{2\gamma}\tilde{\theta}^2 + \frac{\sigma}{\gamma}\tilde{\theta}\theta \pm \frac{\sigma}{2\gamma}\theta^2$$

$$\dot{V}(\varepsilon,\tilde{\theta}) = -\frac{\lambda}{2}\varepsilon^2 - \frac{\sigma}{2\gamma}\tilde{\theta}^2 - \left(\sqrt{\frac{\lambda}{2}}\varepsilon + \sqrt{\frac{1}{2\lambda}}\delta\right)^2 + \frac{1}{2\lambda}\delta^2 - \frac{\sigma}{2\gamma}\left(\tilde{\theta} - \theta\right)^2 + \frac{\sigma}{2\gamma}\theta^2$$

$$\dot{V}(\varepsilon,\tilde{\theta}) \le -\frac{\lambda}{2}\varepsilon^2 - \frac{\sigma}{2\gamma}\tilde{\theta}^2 + \frac{1}{2\lambda}\delta^2 + \frac{\sigma}{2\gamma}\theta^2$$

$$|\delta(t)| \le \bar{\delta}$$

$$\dot{V}(\varepsilon,\tilde{\theta}) \le -\frac{\lambda}{2}\varepsilon^2 - \frac{\sigma}{2\gamma}\tilde{\theta}^2 + \frac{1}{2\lambda}\bar{\delta}^2 + \frac{\sigma}{2\gamma}\theta^2$$

**Solution #2**

$$\dot{V}(\varepsilon,\tilde{\theta}) \leq -\frac{\lambda}{2}\varepsilon^2 - \frac{\sigma}{2\gamma}\tilde{\theta}^2 + \frac{1}{2\lambda}\bar{\delta}^2 + \frac{\sigma}{2\gamma}\theta^2$$

$$\dot{V}(\varepsilon,\tilde{\theta}) \leq -\frac{\lambda}{2}\varepsilon^2 - \frac{\sigma}{2\gamma}\tilde{\theta}^2 + \bar{\Delta}$$

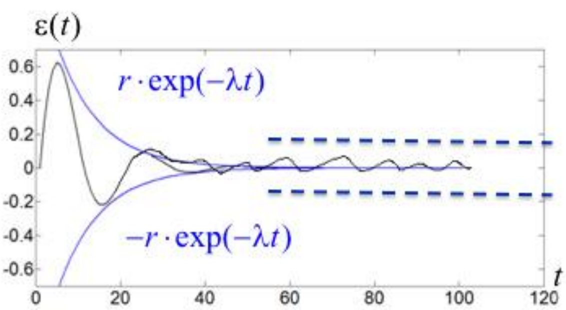
$$\bar{\Delta} = \frac{1}{2\lambda}\bar{\delta}^2 + \frac{\sigma}{2\gamma}\theta^2$$

$$\dot{V}(\varepsilon,\tilde{\theta}) \leq -kV(\varepsilon,\tilde{\theta}) + \bar{\Delta} \qquad k = \min\left\{\lambda, \frac{\sigma}{\gamma}\right\} \qquad (4.13)$$

**Solution #2**

$$\dot{V}(\varepsilon) \leq -kV(\varepsilon) + \bar{\Delta} \qquad \Rightarrow \qquad V(t) \leq \exp(-kt)V(0)\left(1 - \frac{\bar{\Delta}}{k}\right) + \frac{\bar{\Delta}}{k}V(0)$$

Exponential convergence of $\varepsilon$ to bounded set is proved.




**Solution #2**

## Summary

**Adjustable controller:**

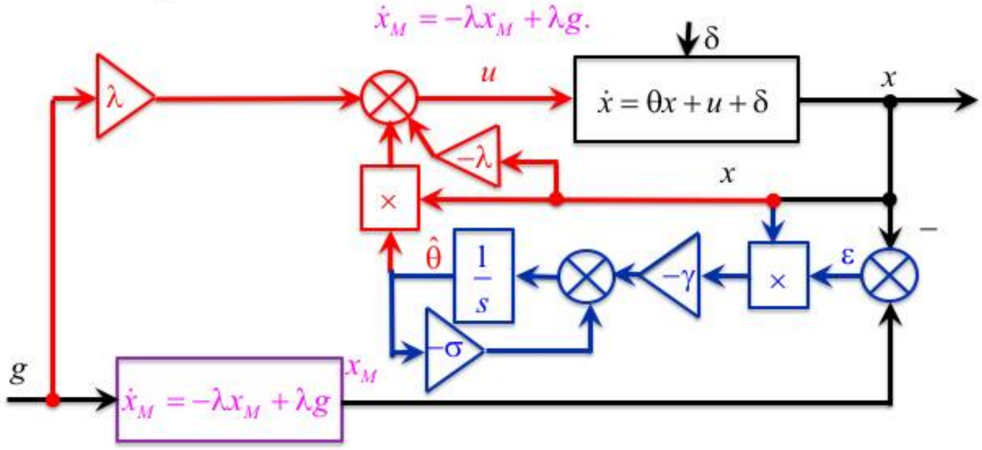
$$u = -\hat{\theta}x - \lambda x + \lambda g \tag{4.8}$$

**Robust modification of adaptation algorithm:**

$$\dot{\hat{\theta}} = -\gamma x \varepsilon - \sigma \hat{\theta} \tag{4.9}$$

with $\varepsilon = x_M - x$ and reference model

$$\dot{x}_M = -\lambda x_M + \lambda g. \tag{4.3}$$




**Solution #2**

## Summary

**Properties of the closed-loop robust system:**

1. All signals in the system are bounded;
2. Control error $\varepsilon = x_M - x$ exponentially tends to the neighborhood of zero;
3. The radius of neighborhood can be arbitrary reduced by

$\lambda$  *or* $\gamma$  *or* $\sigma \downarrow$

$$\dot{V}(\varepsilon) \le -kV(\varepsilon) + \bar{\Delta} \quad \text{where} \quad \bar{\Delta} = \frac{1}{2\lambda}\bar{\delta}^2 + \frac{\sigma}{2\gamma}\theta^2$$

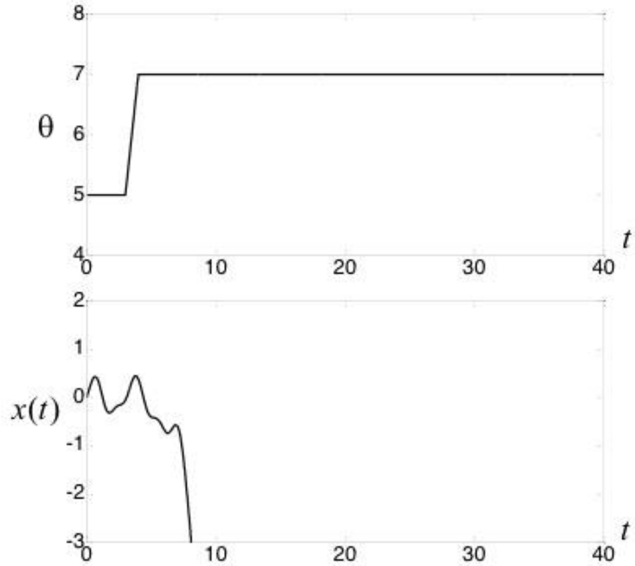
4. Algorithm provides the compensation of uncertainty. If the plant is not disturbed ($\delta = \bar{\delta} = 0$), the error $\varepsilon = x_M - x$ can go to zero, if $\sigma = 0$.

**Example: Classical stabilizing control for unstable plant** $\dot{x} = \theta x + u + \delta$

$$u = -6x$$

$\theta = 5$

$\delta = 0,5\sin(4t) + 0,75\cos(2t)$




**Example: Adaptive robust stabilizing control for the plant** $\dot{x} = \theta x + u + \delta$

$$u = -6x - \hat{\theta}x,$$

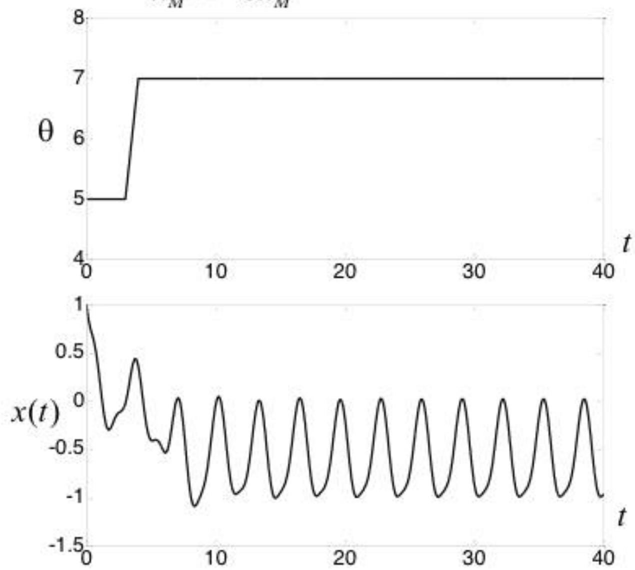
$$\dot{\hat{\theta}} = -\gamma x \varepsilon - \hat{\theta}, \qquad \varepsilon = x_M - x,$$

$$\dot{x}_M = -6x_M$$

$\theta = 5$

$\delta = 0,5\sin(4t) + 0,75\cos(2t)$

$\gamma = 2$



**Example: Adaptive robust stabilizing control for the plant** $\dot{x} = \theta x + u + \delta$

$$u = -6x - \hat{\theta}x,$$

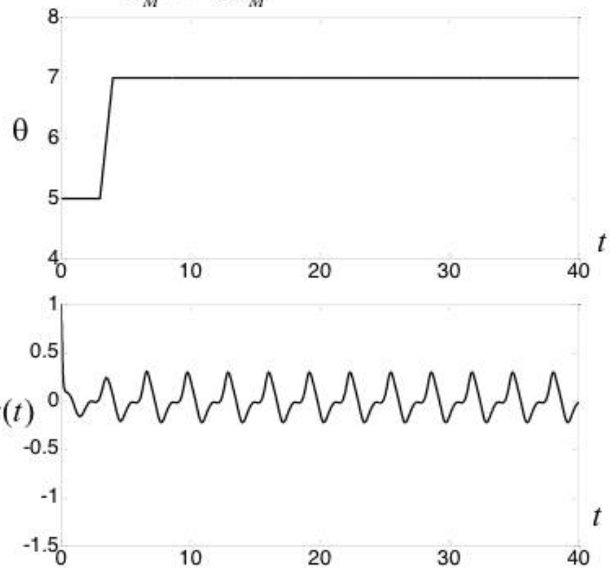
$$\dot{\hat{\theta}} = -\gamma x\varepsilon - \hat{\theta}, \qquad \varepsilon = x_M - x,$$

$$\dot{x}_M = -6x_M$$

$\theta = 5$

$\delta = 0,5\sin(4t) + 0,75\cos(2t)$

$\gamma = 200$




**Example: Adaptive robust tracking control for the plant** $\dot{x} = \theta x + u + \delta$

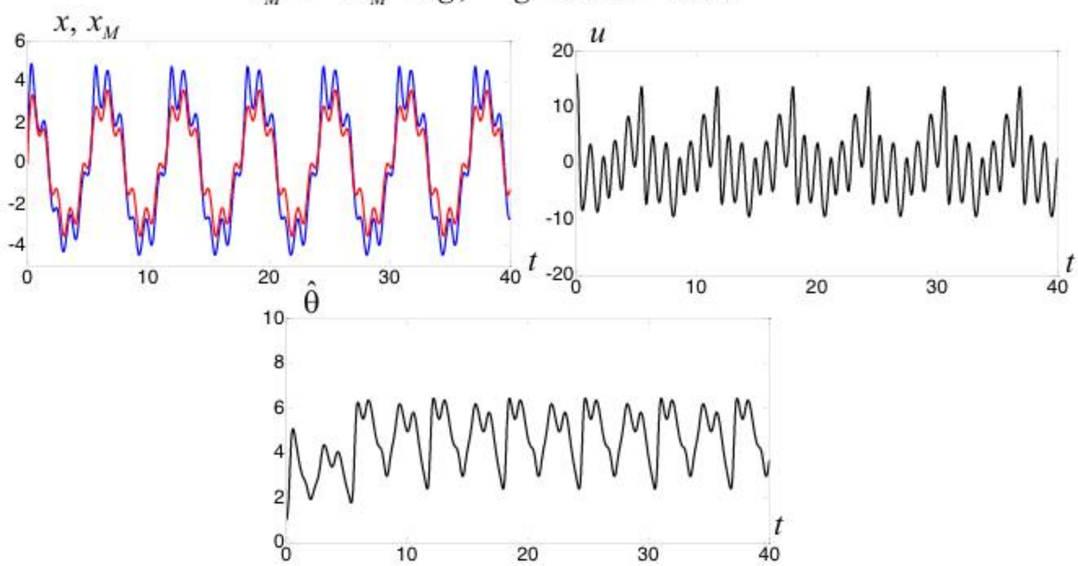
$$u = -6x - \hat{\theta}x + 6g,$$

$$\dot{\hat{\theta}} = -2x\varepsilon - \hat{\theta}, \qquad \varepsilon = x_M - x,$$

$$\dot{x}_M = -6x_M + 6g, \qquad g = \sin 6t + 3\cos t$$

$\theta = 5$

$\delta = 0,5\sin(4t) + 0,75\cos(2t)$



# 5. Generalized Algorithm of Adaptive and Robust Controller Design

How to design an adaptive control?

# 5. Generalized Algorithm of Adaptive and Robust Controller Design

## 1. Problem statement of adaptive control:

**Plant:**

$$\dot{x} = f(\theta, x, u, \delta), \qquad x(0), \tag{5.1}$$

where $\theta$ is the vector of unknown parameters (or functions),

$f \in R^n$ is continuous nonlinear mapping, $\delta \in R^m : \|\delta\| \le \bar{\delta}$ is the disturbance.

**Objective** is to design a control $u$ providing the following inequality:

$$|x_M(t) - x(t)| \le \Delta \text{ for } any\ t \ge T, \tag{5.2}$$

where $x_M$ is the output of reference model

$$\dot{x}_M = A_M x_M + b_M g, \tag{5.3}$$

$g$ is the reference signal, $\lambda$ is the positive parameter.

# 5. Generalized Algorithm of Adaptive and Robust Controller Design

## 2. Nonadaptive controller design:

**Let the plant parameters (functions) $\theta$ be known.**

Luggage of classical control theory


$\Rightarrow$ Nonadaptive control

$$u = U(\theta, x, e, g), \tag{5.4}$$

where $e = x_M - x$ is the control error, $U$ is the nonlinear static or dynamical scalar function.

# 5. Generalized Algorithm of Adaptive and Robust Controller Design

## 3. Adjustable controller design

**Parameters (functions) $\theta$ are unknown.**

Substitute estimates $\hat{\theta}$ for $\theta$ in control (5.4) and obtain adjustable controller:

$$\boxed{u = U(\hat{\theta}, x, e, g)} \tag{5.5}$$

Substitute (38) into the plant $\dot{x} = f(\theta, x, u, \delta)$ :

$$\dot{x} = f(\theta, x, U(\hat{\theta}, x, e, g), \delta)$$

Form the error $e = x_M - x$ and take its derivative:

# 5. Generalized Algorithm of Adaptive and Robust Controller Design

Form the error $e = x_M - x$ and take its derivative:

$$\dot{e} = \dot{x}_M - \dot{x} = (-\lambda x_M + \lambda g) - f(\theta, x, U(\hat{\theta}, x, \varepsilon, g), \delta)$$

$$\Downarrow$$

*Signal Error Model* $\boxed{\dot{e} = E(e, \tilde{\theta}, t)}$ (5.6)

where $E$ is the nonlinear static vector function,

$\tilde{\theta} = \theta - \hat{\theta}$ is the parametric error.

# 5. Generalized Algorithm of Adaptive and Robust Controller Design

## 4. Adaptation algorithm design

Form the parametric error model

*Parametric Error Model* $\boxed{\dot{\tilde{\theta}} = \Omega(e, t),}$ (5.7)

where $\Omega$ is the implementable (measurable) function to be determined.

Adaptation algorithm:

$\boxed{\dot{\hat{\theta}} = -\Omega(e, t),}$ $\quad \dot{\tilde{\theta}} = -\dot{\hat{\theta}}$ (5.8)

## 5. Generalized Algorithm of Adaptive and Robust Controller Design

**5. Determination of $\Omega$**

*Signal Error Model* $$\dot{e} = E\left(e, \tilde{\theta}, t\right) \quad (5.6)$$

*Parametric Error Model* $$\dot{\tilde{\theta}} = \Omega\left(e, t\right), \quad (5.7)$$

Choose a Lyapunov function candidate

$$V = V(e, \tilde{\theta}, t).$$

Take its derivative in amount of (5.6) and (5.7) $\dot{V}(e, \tilde{\theta})$.

## 5. Generalized Algorithm of Adaptive and Robust Controller Design

$$\dot{e} = E\left(e, \tilde{\theta}, t\right) \qquad \dot{\tilde{\theta}} = \Omega\left(e, t\right),$$

$$\searrow \quad \swarrow$$

$$\dot{V}(e, \tilde{\theta}).$$

**Condition**

$$\dot{V}(e, \tilde{\theta}) < 0.$$

**gives**

$$\Downarrow$$

**Adaptation algorithm:** $\Omega\left(e, t\right)$

$$\dot{\hat{\theta}} = -\Omega\left(e, t\right) \quad (5.8)$$

## 5. Generalized Algorithm of Adaptive and Robust Controller Design

**Summary**

Adjustable control

$$u = U(\hat{\theta}, x, \varepsilon, g) \tag{5.5}$$

Adaptation algorithm

$$\dot{\hat{\theta}} = -\Omega(e, t) \tag{5.8}$$

## 5. Generalized Algorithm of Adaptive and Robust Controller Design

**There is no any universal approach of Lyapunov function choice!**

## 5. Generalized Algorithm of Adaptive and Robust Controller Design

**There is no any universal approach of Lyapunov function choice!**

**However, there are standard errors models with preliminary selected Lyapunov functions and designed Adaptation Algorithms**

## 6. Standard Error Models and Adaptation Algorithms Design

### 6.1. Static Error Model

$$\varepsilon(t) = \tilde{\theta}^T(t)\omega(t), \tag{6.1}$$

where $\varepsilon(t)$ is the output, $\tilde{\theta}(t) = \theta - \hat{\theta}(t) \in \mathbb{R}^m$ is the vector of parametric errors, $\omega(t) \in \mathbb{R}^m$ is the vector of measurable functions (regressor).

***Remark 6.1.*** **The model is widely used in the problems of identification (see example below).**

$$V = \frac{1}{2\gamma}\tilde{\theta}^T\tilde{\theta} \tag{6.2}$$

with a positive gain $\gamma$.

Time derivative:

$$\dot{\tilde{\theta}} = -\dot{\hat{\theta}}$$

$$\dot{V} = \frac{1}{\gamma}\tilde{\theta}^T\dot{\tilde{\theta}} = -\frac{1}{\gamma}\tilde{\theta}^T\dot{\hat{\theta}} = \quad ?$$

## 6.1. Static Error Model

$$\dot{V} = \frac{1}{\gamma}\tilde{\theta}^T\dot{\tilde{\theta}} = -\frac{1}{\gamma}\tilde{\theta}^T\dot{\hat{\theta}} = -\frac{1}{\gamma}\tilde{\theta}^T\gamma\omega\varepsilon = -\varepsilon^2 < 0$$

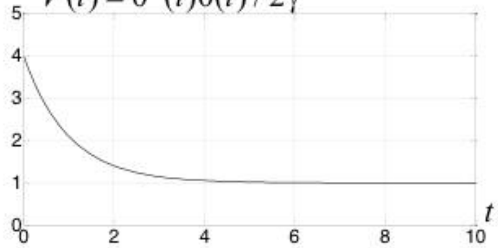
**Summary and Discussion**

| | |
|---|---|
| *Error Model* | $\varepsilon = \tilde{\theta}^T\omega$ , |
| *Adaptation Algorithm* | $\dot{\hat{\theta}} = \gamma\omega\varepsilon$ |
| *Lyapunov function* | $V = \frac{1}{2\gamma}\tilde{\theta}^T\tilde{\theta} > 0$ |
| *Its time derivative* | $\dot{V} = -\varepsilon^2 < 0$ |

**What it means?**

## 6.1. Static Error Model

**1.** $V(t) = \tilde{\theta}^T(t)\tilde{\theta}(t)/2\gamma$




$\Rightarrow$ $\varepsilon \to 0$ *asymptotically fast*

$\|\tilde{\theta}\|^2$ *is monotonically decreasing function*

**2. Example 6.1**

For a given error model $\varepsilon = \tilde{\theta}_1\omega_1 + \tilde{\theta}_2\omega_2$ there are following scenarios:

a) $\omega_1 = 1,\ \omega_2 = 2$ *and* $\tilde{\theta}_1 \to 2,\ \tilde{\theta}_2 \to -1$

b) $\omega_1 = 1,\ \omega_2 = 2$ *and* $\tilde{\theta}_1 \to 4,\ \tilde{\theta}_2 \to -2$

c) $\omega_1 = \sin t,\ \omega_2 = 2\sin t$ *and* $\tilde{\theta}_1 \to 2,\ \tilde{\theta}_2 \to -1$

d) $\omega_1 = \sin t,\ \omega_2 = 2\sin 2t$ *and* $\tilde{\theta}_1 \to ?,\ \tilde{\theta}_2 \to ?$

**How many options for convergence?**

## 6.1. Static Error Model

**Example 6.2**

For a given error model $\varepsilon = \tilde{\theta}_1\omega_1 + \tilde{\theta}_2\omega_2 + \tilde{\theta}_3\omega_3$ there are following scenarios:

a) $\omega_1 = \sin t,\ \omega_2 = 2\sin t,\ \omega_3 = 3\sin t$ *and* $\tilde{\theta}_{1,2,3} \to ?$

b) $\omega_1 = \sin t,\ \omega_2 = 2\sin t,\ \omega_3 = 3\sin 2t$ *and* $\tilde{\theta}_{1,2,3} \to ?$

c) $\omega_1 = \sin t,\ \omega_2 = 2\sin 3t,\ \omega_3 = 3\sin 2t$ *and* $\tilde{\theta}_{1,2,3} \to ?$

d) $\omega_1 = \sin(t),\ \omega_2 = 2\sin(t+\pi),\ \omega_3 = 3\sin(t+\pi/2)$ *and* $\tilde{\theta}_{1,2,3} \to ?$

e) $\omega_1 = \sin(2t),\ \omega_2 = 2\sin(t+\pi),\ \omega_3 = 3\sin(t+\pi/2)$ *and* $\tilde{\theta}_{1,2,3} \to ?$

## 6.1. Static Error Model

**Example 6.2**

For a given error model $\varepsilon = \tilde{\theta}_1\omega_1 + \tilde{\theta}_2\omega_2 + \tilde{\theta}_3\omega_3$ there are following scenarios:

a) $\omega_1 = \sin t,\ \omega_2 = 2\sin t,\ \omega_3 = 3\sin t$ *and* $\tilde{\theta}_{1,2,3} \to$ *not ness. to zero*

b) $\omega_1 = \sin t,\ \omega_2 = 2\sin t,\ \omega_3 = 3\sin 2t$ *and* $\tilde{\theta}_{1,2,3} \to$ *not ness. to zero*

c) $\omega_1 = \sin t,\ \omega_2 = 2\sin 3t,\ \omega_3 = 3\sin 2t$ *and* $\tilde{\theta}_{1,2,3} \to 0$

d) $\omega_1 = \sin(t),\ \omega_2 = 2\sin(t+\pi),\ \omega_3 = 3\sin(t+\pi/2)$ *and* $\tilde{\theta}_{1,2,3} \to$ *not n. to zero*

e) $\omega_1 = \sin(2t),\ \omega_2 = 2\sin(t+\pi),\ \omega_3 = 3\sin(t+\pi/2)$ *and* $\tilde{\theta}_{1,2,3} \to 0$

*Vector* $\omega \in {}^{\sim m}$ *has to contain at least* $m/2$ *different harmonics to provide identification properties*

## 6.1. Static Error Model

### Summary

**Properties of the closed-loop robust system:**

1. If $\omega$ is bounded, all signals in the system are bounded;
2. Error $\varepsilon$ approaches zero asymptotically fast;
3. Function $\|\tilde{\theta}\|^2$ is monotonically nonincreasing;
4. $\|\tilde{\theta}\|^2$ approaches zero asymptotically fast, if $\omega$ contains at least *m/2* harmonics and consists of linearly independent elements;

This property can be reformulated in terms of **Persistent Excitation Condition:**

$$\int_t^{t+T} \omega(\tau)\omega^T(\tau)d\tau \geq \alpha I \tag{6.3}$$

for some positive $\alpha$, $T$.

## 6.1. Static Error Model

**Example 6.3. The problem of identification reduced to Static Error Model**

Problem statement

Let a plant be described by

$$\ddot{y} + a_1\dot{y} + a_0 y = b_0 u \tag{6.4}$$

with unknown parameters $a_0$, $a_1$, $b_0$ and measurable input $u$ and output $y$.

The objective is to design such estimates $\hat{a}_0, \hat{a}_1, \hat{b}_0$ that obey equalities

$$\lim_{t\to\infty}(a_0 - \hat{a}_0) = \lim_{t\to\infty}(a_1 - \hat{a}_1) = \lim_{t\to\infty}(b_0 - \hat{b}_0) = 0. \tag{6.5}$$

## 6.1. Static Error Model

Solution

1. Apply transfer function

$$H(s) = \frac{1}{K(s)} = \frac{1}{s^2 + k_1 s + k_0}$$

with Hurwitz polynomial $K(s) = s^2 + k_1 s + k_0$ to the plant (6.4) assuming initial conditions $y(0), \dot{y}(0)$ equaled to zero:

$$\ddot{y} + a_1 \dot{y} + a_0 y = b_0 u$$

$$\Big\Downarrow H(s)[\bullet]$$

$$\frac{s^2}{K(s)}[y] + a_1 \frac{s}{K(s)}[y] + a_0 \frac{1}{K(s)}[y] = b_0 \frac{1}{K(s)}[u]$$

## 6.1. Static Error Model

Solution

$$\frac{s^2 \pm k_1 s \pm k_0}{K(s)}[y] + a_1 \frac{s}{K(s)}[y] + a_0 \frac{1}{K(s)}[y] = b_0 \frac{1}{K(s)}[u]$$

$$y + (a_1 - k_1)\frac{s}{K(s)}[y] + (a_0 - k_0)\frac{1}{K(s)}[y] = b_0 \frac{1}{K(s)}[u]$$

$$y = \underbrace{(k_1 - a_1)}_{\theta_1}\underbrace{\frac{s}{K(s)}[y]}_{\omega_1} + \underbrace{(k_0 - a_0)}_{\theta_2}\underbrace{\frac{1}{K(s)}[y]}_{\omega_2} + \underbrace{b_0}_{\theta_3}\underbrace{\frac{1}{K(s)}[u]}_{\omega_3}$$

*Parameterized plant*

$$y = \theta^T \omega \tag{6.6}$$

$$\theta = col(\theta_1, \theta_2, \theta_3), \quad \omega = col(\omega_1, \omega_2, \omega_3)$$

## 6.1. Static Error Model

Solution

2. Design of error

$$\varepsilon = y - \hat{\theta}^T \omega \qquad (6.7)$$

where $\hat{\theta}$ is the estimate of $\theta$.

$$y = \theta^T \omega$$

$$\Downarrow$$

*Error model* $\varepsilon = \tilde{\theta}^T \omega,$

$\tilde{\theta} = \theta - \hat{\theta}$ is parametric error vector.


$$\Downarrow$$

3. Adaptation algorithm design.

*Adaptation algorithm* $$\dot{\hat{\theta}} = \gamma \omega \varepsilon \qquad (6.8)$$

## 6.1. Static Error Model

Solution Summary

*Error*

$$\varepsilon = y - \hat{\theta}^T \omega = y - (k_1 - \hat{a}_1)\frac{s}{K(s)}[y] - (k_0 - \hat{a}_0)\frac{1}{K(s)}[y] - \hat{b}_0 \frac{1}{K(s)}[u]$$

*Adaptation Algorithms*

$$\dot{\hat{\theta}} = \gamma \omega \varepsilon \rightarrow$$

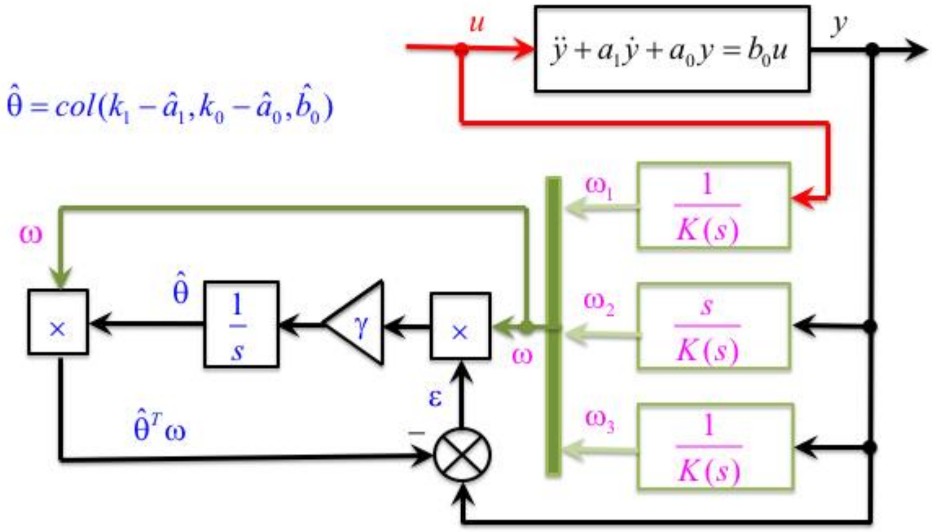
$$\dot{\hat{a}}_0 = -\gamma \frac{1}{K(s)}[y]\varepsilon$$

$$\dot{\hat{a}}_1 = -\gamma \frac{s}{K(s)}[y]\varepsilon$$

$$\dot{\hat{b}}_0 = \gamma \frac{1}{K(s)}[u]\varepsilon$$

## 6.1. Static Error Model

General Scheme

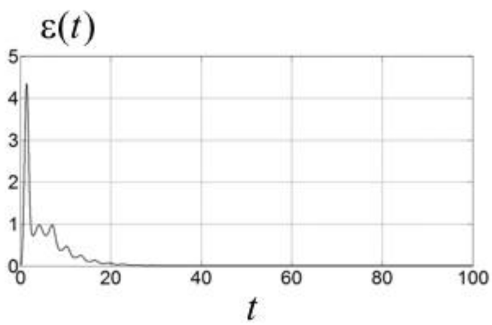


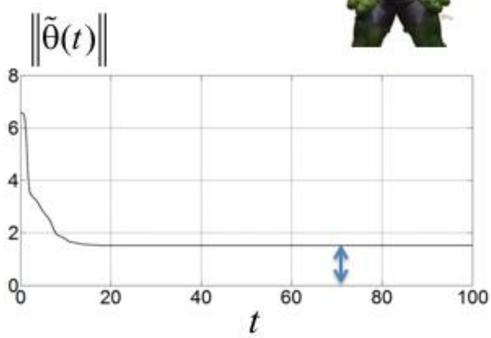

## 6.1. Static Error Model

Simulation results

| | | |
|---|---|---|
| *Plant* | $\ddot{y} + 2\dot{y} + y = 3u$ | |
| *Filters polynomial* | $K(s) = s^2 + 5s + 6$ | $\theta = col(3,\ 5,\ 3)$ |
| *Adaptation gain* | $\gamma = 1$ | |

$u(t) = 10\sin t$



## 6.1. Static Error Model

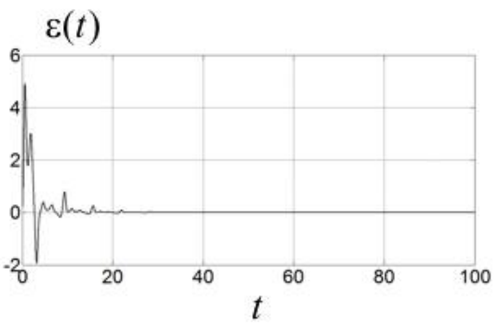
<u>Simulation results</u>

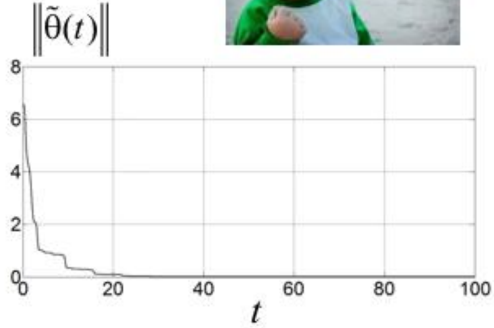
*Plant* $\ddot{y}+2\dot{y}+y=3u$

*Filters polynomial* $K(s)=s^2+5s+6$ $\theta = col(3,\ 5,\ 3)$

*Adaptation gain* $\gamma=1$

$u(t)=10\sin t+20\cos 2t$


## 6.1. Static Error Model

**M.I.T. rule as an alternative methodology to Lyapunov functions**

*Error model* $\varepsilon(t)=\tilde{\theta}^T(t)\omega(t),$

*Lyapunov function* | *M.I.T. rule*

*Function* | *Performance index*

$$V=\frac{1}{2\gamma}\tilde{\theta}^T\tilde{\theta} \qquad J(\varepsilon)=\frac{1}{2}\varepsilon^2$$

$$\dot{V}=-\varepsilon^2 \qquad \dot{\hat{\theta}}=\gamma\ \underset{\tilde{\theta}}{grad}\ J(\varepsilon)$$

*Adaptation algorithm* $\dot{\hat{\theta}}=\gamma\omega\varepsilon$

## 6.2. Dynamic error model with measurable state

$$\dot{e}(t) = Ae(t) + b\tilde{\theta}^T(t)\omega(t),$$
$$\varepsilon(t) = Ce(t) \tag{6.9}$$

where $e \in {\sim}^n$ is the state $\varepsilon$ is the output, $\tilde{\theta} \in {\sim}^m$ is the vector of parametric errors, $\omega \in {\sim}^m$ is the vector of measurable functions (regressor).

***Remark 6.2.*** **The model is widely used in the problems of state adaptive control (see example below).**

$$V = \frac{1}{2}e^T Pe + \frac{1}{2\gamma}\tilde{\theta}^T\tilde{\theta} \tag{6.10}$$

with a positive gain $\gamma$ and positively defined symmetric matrix $P = P^T \succ 0$ defined later.

## 6.2. Dynamic error model with measurable state

Time derivative:

$$\dot{e} = Ae + b\tilde{\theta}^T\omega \qquad \dot{\tilde{\theta}} = -\dot{\hat{\theta}}$$

$$\dot{V} = \frac{1}{2}\dot{e}^T Pe + \frac{1}{2}e^T P\dot{e} + \frac{1}{\gamma}\tilde{\theta}^T\dot{\tilde{\theta}} = \frac{1}{2}\left(Ae + b\tilde{\theta}^T\omega\right)^T Pe + \frac{1}{2}e^T P\left(Ae + b\tilde{\theta}^T\omega\right) - \frac{1}{\gamma}\tilde{\theta}^T\dot{\hat{\theta}} =$$

$$\frac{1}{2}e^T A^T Pe + \frac{1}{2}e^T PAe + b^T\tilde{\theta}^T\omega Pe - \frac{1}{\gamma}\tilde{\theta}^T\dot{\hat{\theta}} = \frac{1}{2}e^T\left(A^T P + PA\right)e + \tilde{\theta}^T\omega b^T Pe - \frac{1}{\gamma}\tilde{\theta}^T\dot{\hat{\theta}}$$

*Since matrix $A$ is Hurwitz, it is related to the matrix $P$ via Lyapunov equation* $A^T P + PA = -Q$ *with* $Q = Q^T \succ 0$

$$\dot{V} = -\frac{1}{2}e^T Qe + \tilde{\theta}^T\omega b^T Pe - \frac{1}{\gamma}\tilde{\theta}^T\dot{\hat{\theta}}$$

**Adaptation algorithm?**

## 6.2. Dynamic error model with measurable state

$$\dot{V} = -\frac{1}{2}e^T Qe + \tilde{\theta}^T \omega b^T Pe - \frac{1}{\gamma}\tilde{\theta}^T \dot{\hat{\theta}}$$

If $\dot{\hat{\theta}} = \gamma\omega b^T Pe$ ,

$$\dot{V} = -\frac{1}{2}e^T Qe < 0 \qquad (6.11)$$

## 6.2. Dynamic error model with measurable state

### Summary and Discussion

| | |
|---|---|
| *Error Model* | $\dot{e} = Ae + b\tilde{\theta}^T \omega$ |
| *Adaptation Algorithm* | $\dot{\hat{\theta}} = \gamma\omega b^T Pe$ |
| *Lyapunov function* | $V = \frac{1}{2}e^T Pe + \frac{1}{2\gamma}\tilde{\theta}^T \tilde{\theta}$ |
| *Its time derivative* | $\dot{V} = -\frac{1}{2}e^T Qe < 0$ |

} **What it means?**

## 6.2. Dynamic error model with measurable state

### Summary

**Properties of the closed-loop robust system:**

1. If $\omega$ is bounded, all signals in the system are bounded;
2. Error $\|e\|$ approaches zero asymptotically fast;
3. Function $\|\tilde{\theta}\|^2$ is monotonically nonincreasing;
4. $\|\tilde{\theta}\|^2$ approaches zero asymptotically fast, if $\omega$ contains at least *m/2* harmonics and consists of linearly independent elements;

This property can be reformulated in terms of **Persistent Excitation Condition:**

$$\int_t^{t+T} \omega(\tau)\omega^T(\tau)d\tau \geq \alpha I$$

for some positive $\alpha$, $T$.

## 6.2. Dynamic error model with measurable state

**Example 6.4. The problem of state adaptive control**

<u>Problem statement</u>

Let a plant be described by

$$\underbrace{\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix} 0 & 1 \\ -a_0 & -a_1 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}}_{x} + \underbrace{\begin{bmatrix} 0 \\ b_0 \end{bmatrix}}_{b} u \qquad (6.12)$$

with **unknown** parameters $a_0$, $a_1$ , known $b_0$ and measurable state input $u$ and output $y$.

The objective is to design a control $u$ such that

$$\lim_{t\to\infty} \|x_M - x\| = 0 \qquad (6.13)$$

## 6.2. Dynamic error model with measurable state

$x_M$ is the state of reference model

$$\underbrace{\begin{bmatrix}\dot{x}_{M1}\\ \dot{x}_{M2}\end{bmatrix}}_{\dot{x}_M}=\underbrace{\begin{bmatrix}0 & 1\\ -a_{M0} & -a_{M1}\end{bmatrix}}_{A_M}\underbrace{\begin{bmatrix}x_{M1}\\ x_{M2}\end{bmatrix}}_{x_M}+\underbrace{\begin{bmatrix}0\\ b_{M0}\end{bmatrix}}_{b_M}g \qquad (6.14)$$

with parameters $a_{M0}$, $a_{M1}$, $b_{M0}$ responsible for transient performance of the closed-loop system and reference signal $g$ .

**Main idea of solution is to reduce the problem to the error model.**

**Then to get the adaptation algorithm.**


## 6.2. Dynamic error model with measurable state

Solution

**1.** Let the parameters $a_0$, $a_1$ be known.

Form the error signal $e = x_M - x$ and take its derivative in amount of plant and reference model equations:

$$\dot{e} = \dot{x}_M - \dot{x} = A_M x_M + b_M g - Ax - bu$$

Let $\dot{e} \triangleq A_M e$ $\left(e(t) = \exp(A_M t)e(0) \to 0 \ \textit{exponentially fast}\right)$.

Then

$$A_M x_M + b_M g - Ax - bu \triangleq A_M e$$

$$\cancel{A_M x_M} + b_M g - Ax - bu \triangleq \cancel{A_M x_M} - A_M x$$

$$bu = \left(A_M - A\right)x + b_M g$$

## 6.2. Dynamic error model with measurable state

Solution

$$bu = (A_M - A)x + b_M g$$

$$\Downarrow$$

$$\begin{bmatrix} 0 \\ b_0 \end{bmatrix} u = \left( \begin{bmatrix} 0 & 1 \\ -a_{M0} & -a_{M1} \end{bmatrix} - \begin{bmatrix} 0 & 1 \\ -a_0 & -a_1 \end{bmatrix} \right) x + \begin{bmatrix} 0 \\ b_{M0} \end{bmatrix} g$$

$$\Downarrow$$

$$u = \frac{1}{b_0}\Big[ \underbrace{(a_0 - a_{M0})}_{\theta_1} x_1 + \underbrace{(a_1 - a_{M1})}_{\theta_2} x_2 + b_{M0} g \Big]$$

$$\Downarrow$$

*Nonadaptive control*

$$u = \frac{1}{b_0}\left[ \theta^T x + b_{M0} g \right] \tag{6.15}$$

## 6.2. Dynamic error model with measurable state

Solution

**2.** Let the parameters $a_0$, $a_1$ be unknown. Control

$$u = \frac{1}{b_0}\left[ \theta^T x + b_{M0} g \right]$$

is not implementable. Substitute estimate $\hat{\theta}$ for $\theta$ and obtain implementable adjustable control:

*Adjustable control*

$$u = \frac{1}{b_0}\left[ \hat{\theta}^T x + b_{M0} g \right] \tag{6.16}$$

Replace (6.16) in the plant equation $\dot{x} = Ax + bu$ :

$$\dot{x} = A\,x + b\frac{1}{b_0}\left[ \hat{\theta}^T x + b_{M0} g \right]$$

## 6.2. Dynamic error model with measurable state

Solution

Evaluate time derivative of error:

$$\dot{e} = \dot{x}_M - \dot{x} = A_M x_M + b_M g - Ax - b\frac{1}{b_0}\left[\hat{\theta}^T x + b_{M0} g\right] \pm A_M x$$

$$\Downarrow$$

$$\dot{e} = A_M e + \cancel{b_M g} + \left(A_M - A\right)x - b\frac{1}{b_0}\left[\hat{\theta}^T x + \cancel{b_{M0} g}\right]$$

$$\Downarrow$$

$$\begin{bmatrix}\dot{e}_1\\ \dot{e}_2\end{bmatrix} = \begin{bmatrix}0 & 1\\ -a_{M0} & -a_{M1}\end{bmatrix}\begin{bmatrix}e_1\\ e_2\end{bmatrix} +$$

$$\left(\begin{bmatrix}0 & 1\\ -a_{M0} & -a_{M1}\end{bmatrix} - \begin{bmatrix}0 & 1\\ -a_0 & -a_1\end{bmatrix}\right)\begin{bmatrix}x_1\\ x_2\end{bmatrix} - \begin{bmatrix}0\\ b_0\end{bmatrix}\frac{1}{b_0}\hat{\theta}^T x$$

## 6.2. Dynamic error model with measurable state

Solution

$$\begin{bmatrix}\dot{e}_1\\ \dot{e}_2\end{bmatrix} = \begin{bmatrix}0 & 1\\ -a_{M0} & -a_{M1}\end{bmatrix}\begin{bmatrix}e_1\\ e_2\end{bmatrix} + \begin{bmatrix}0\\ \underbrace{(a_0 - a_{M0})}_{\theta_1} + \underbrace{(a_1 - a_{M1})}_{\theta_2}\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} - \underbrace{\begin{bmatrix}0\\ 1\end{bmatrix}}_{k}\hat{\theta}^T x$$

$$\Downarrow$$

$$\dot{e} = A_M\, e + k\theta^T x - k\hat{\theta}^T x$$

$$\Downarrow$$

$$\dot{e} = A_M\, e + k\tilde{\theta}^T x \qquad (6.17)$$


with parametric error $\tilde{\theta} = \theta - \hat{\theta}$.

## 6.2. Dynamic error model with measurable state

Solution

*Error model* $\dot{e} = A_M\, e + k\tilde{\theta}^T x$

$$\Downarrow$$

*Adaptation Algorithm* $\dot{\hat{\theta}} = \gamma x k^T P e$ (6.18)

where $\gamma$ is a positive gain, $P = P^T \succ 0$ is the solution of Lyapunov equation

$$A_M^T P + P A_M = -Q \quad (6.19)$$

with preliminary selected $Q = Q^T \succ 0$.

## 6.2. Dynamic error model with measurable state

Solution Summary

*Adjustable control* $u = \frac{1}{b_0}\left[\hat{\theta}^T x + b_{M0} g\right]$ (6.16)

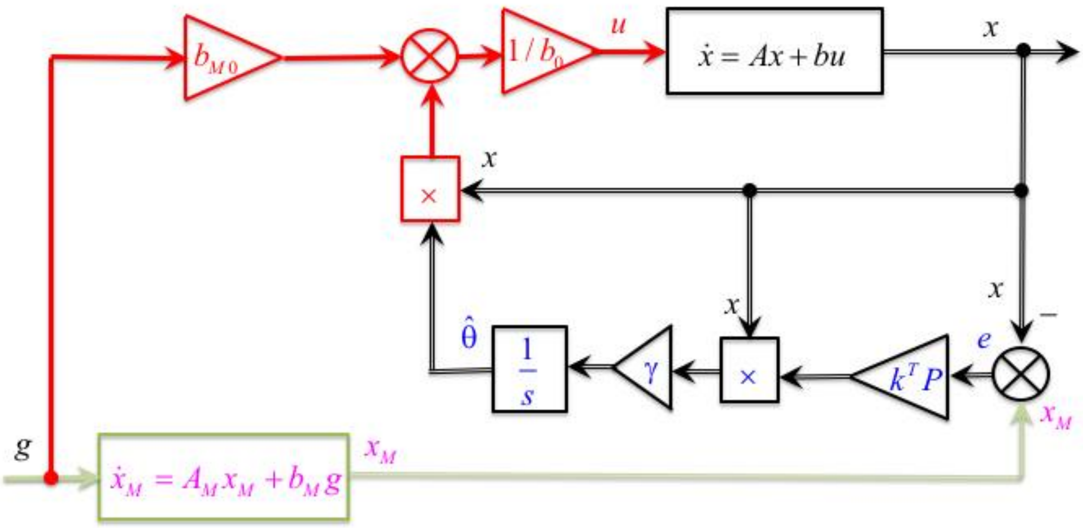
*Adaptation Algorithm* $\dot{\hat{\theta}} = \gamma x k^T P e$ (6.18)

*Error* $e = x_M - x$

*Lyapunov equation* $A_M^T P + P A_M = -Q$ (6.19)

## 6.2. Dynamic error model with measurable state

General Scheme




## 6.2. Dynamic error model with measurable state

Simulation results

*Plant (unstable)*

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -a_0 & -a_1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

*Unknown parameters*

$$a_0 = 1, \; a_1 = -2$$

*Reference model*

$$\begin{bmatrix} \dot{x}_{M1} \\ \dot{x}_{M2} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -6 & -5 \end{bmatrix} \begin{bmatrix} x_{M1} \\ x_{M2} \end{bmatrix} + \begin{bmatrix} 0 \\ 6 \end{bmatrix} g$$

*Adaptation gain*

$$\gamma = 100$$

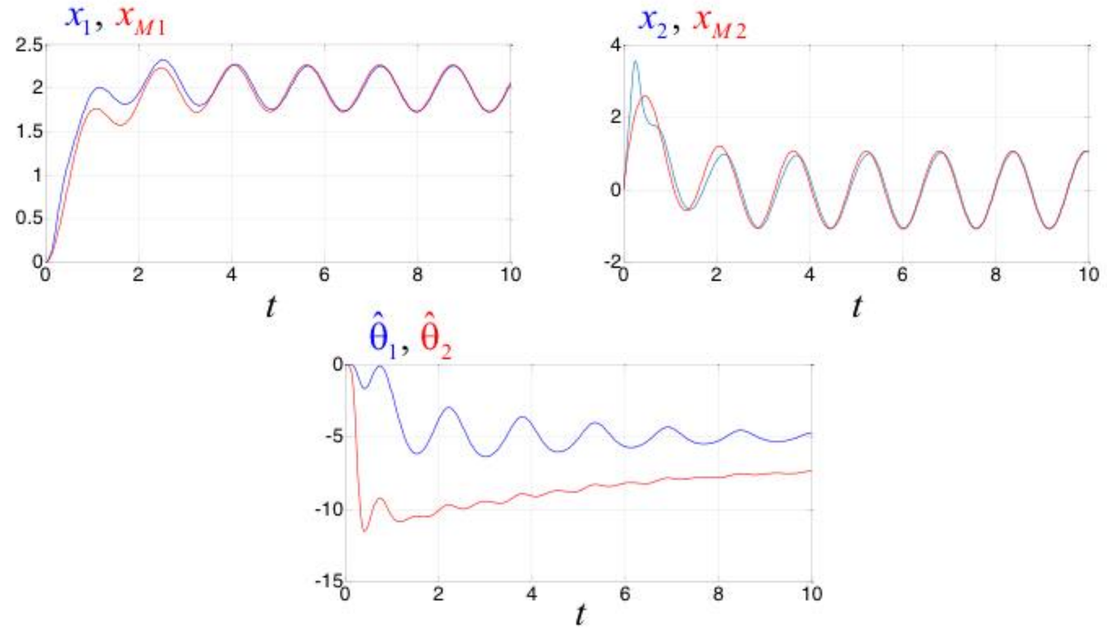
*Matrix P*

$$P = \begin{bmatrix} 1.1167 & 0.0833 \\ 0.0833 & 0.1167 \end{bmatrix}$$

*Reference* $g(t) = \sin 4t + 2$

## 6.2. Dynamic error model with measurable state

Simulation results

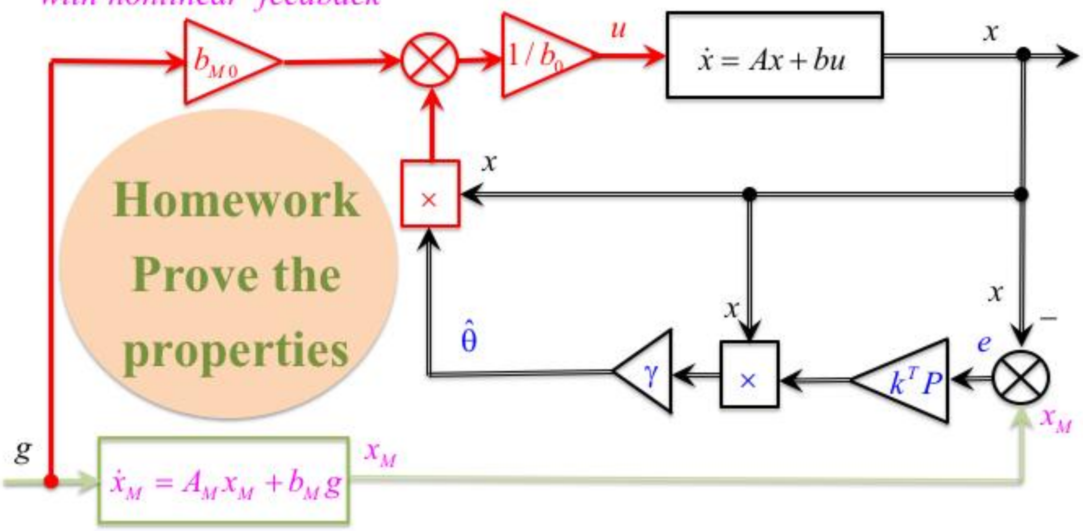



## 6.2. Dynamic error model with measurable state

**Robust modifications of adaptation algorithm**

*Modification with nonlinear feedback*

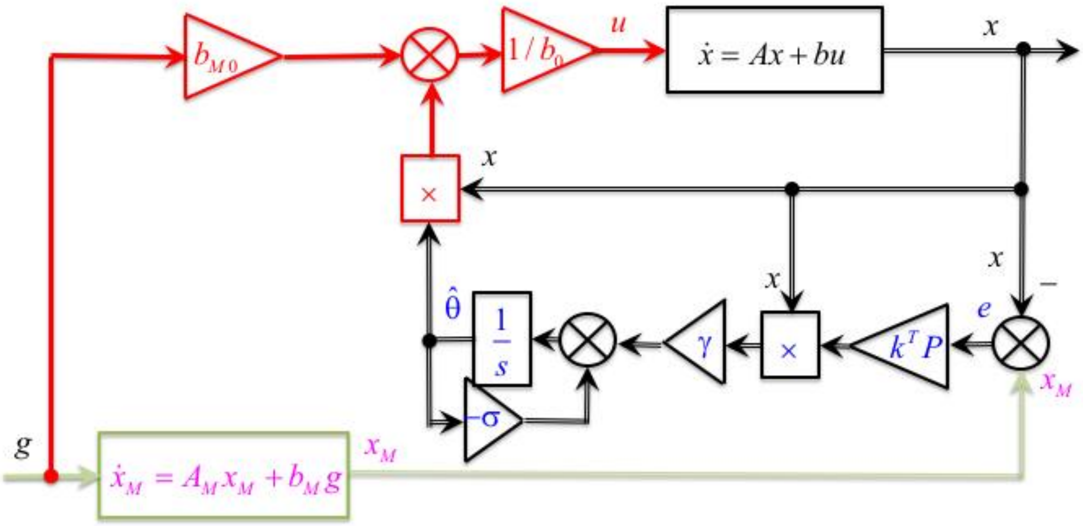
$$\dot{\hat{\theta}} = \gamma \omega b^T P e$$



## 6.2. Dynamic error model with measurable state

### Robust modifications of adaptation algorithm

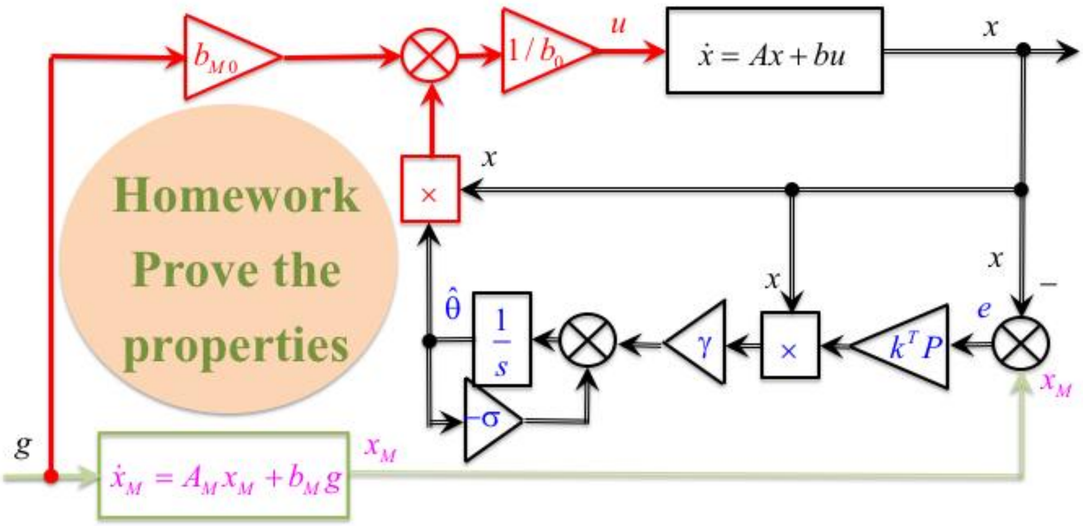
*σ – Modification* $\dot{\hat{\theta}} = -\sigma\hat{\theta} + \gamma\omega b^T Pe$




## 6.2. Dynamic error model with measurable state

### Robust modifications of adaptation algorithm

*σ – Modification* $\dot{\hat{\theta}} = -\sigma\hat{\theta} + \gamma\omega b^T Pe$



## 6.3. Dynamic error model with measurable output

$$\dot{e}(t) = Ae(t) + b\tilde{\theta}^T(t)\omega(t),$$
$$\varepsilon(t) = Ce(t) \qquad (6.20a)$$

where $e \in \sim^n$ is the unmeasurable state $\varepsilon$ is the output, $\tilde{\theta} \in \sim^m$ is the vector of parametric errors, $\omega \in \sim^m$ is the vector of measurable functions (regressor).

***Remark 6.3.  Since vector  is not measurable, the model (6.20a) can be presented in the "Input-Output" form***

$$\varepsilon(t) = W(s)\left[\tilde{\theta}^T(t)\omega(t)\right] \qquad (6.20b)$$

***with transfer function*** $W(s) = C(Is - A)^{-1}b.$

## 6.3. Dynamic error model with measurable output

***Remark 6.4.*** **The model is widely used in the problems of output adaptive control (see example below).**

**The problem is to design an adaptation algorithm/algorithms based on (6.20)**

## 6.3. Dynamic error model with measurable output

**Solution #1**

**Can we just apply adaptation algorithm**

$$\dot{\hat{\theta}} = \gamma\omega\varepsilon$$

**used for static error model?**

**IF YES, WHEN???**

## 6.3. Dynamic error model with measurable output

**Solution #1**

$$\dot{e} = Ae + b\tilde{\theta}^T\omega,$$
$$\varepsilon = Ce$$
$$\Downarrow$$
$$\dot{\hat{\theta}} = \gamma\,\omega b^T Pe$$
$$\Downarrow$$

*"e" unmeasurable*

If $b^T P = C$, adaptation algorithm becomes implementable, since

$$\dot{\hat{\theta}} = \gamma\,\omega b^T Pe = \gamma\,\omega\varepsilon \qquad (6.21)$$

## 6.3. Dynamic error model with measurable output

**Solution #1**

**Lemma (Yakubovich-Kalman-Popov):**

Matrix $P = P^T \succ 0$ obeys Lyapunov equation

$$A^T P + PA = -Q$$

and equation

$$b^T P = C$$

simultaniously, iff transfer function

$$W(s) = C(Is - A)^{-1} b.$$

is Strictly Positive Real (SPR).

## 6.3. Dynamic error model with measurable output

**Solution #1**

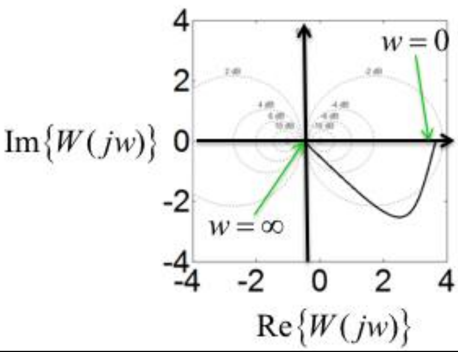
**Definition 6.1. Transfer function** $W(s) = C(Is - A)^{-1} b$ **is SPR if**

1. It is stable, i.e. polynomial of its denominator is Hurwitz (has all the roots in the left half plane of root locus);
2. Nyiqust plot is placed in the right half plane of the diagram.

$$\operatorname{Re}\{W(jw)\} > 0, \quad \forall w \in [0, \infty).$$

3. The limit equality hold

$$\lim_{w \to \infty} w^2 \operatorname{Re}\{W(jw)\} > 0$$



## 6.3. Dynamic error model with measurable output

**Example 6.5. SPR transfer function of first order block**

$$W(s) = \frac{K}{Ts+1}$$

with some positive constant parameters $K$ and $T$ .

**Verification**

1. Frequency transfer function

$$W(jw) = \frac{K}{Tjw+1} = \frac{K(-Tjw+1)}{(Tjw+1)(-Tjw+1)} = \frac{K}{T^2w^2+1} - j\frac{KTw}{T^2w^2+1}$$

2. The first condition: $Ts+1=0 \Rightarrow s_1 = -1/T \Rightarrow W(s)\ is\ Hurwitz$

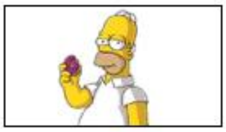
3. The second condition:

$$\mathrm{Re}\{W(jw)\} = \frac{K}{T^2w^2+1} > 0, \quad \forall w \in [0, \infty).$$

## 6.3. Dynamic error model with measurable output

4. The third condition:

$$\lim_{w\to\infty} w^2\,\mathrm{Re}\{W(jw)\} = \lim_{w\to\infty}\frac{Kw^2}{T^2w^2+1} = \frac{K}{T^2} > 0.$$



## 6.3. Dynamic error model with measurable output

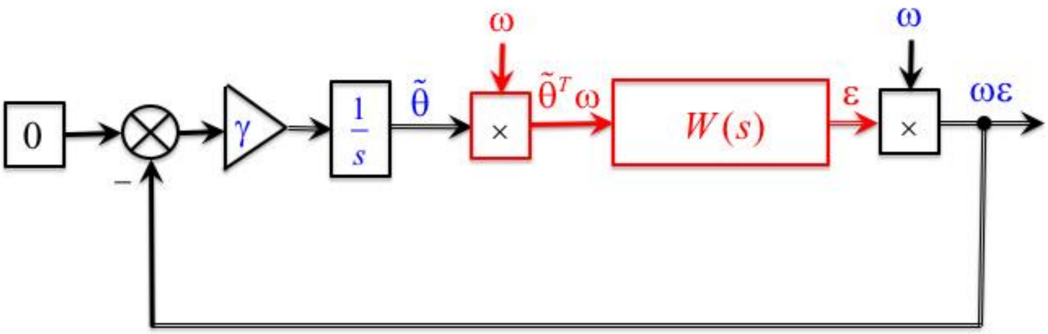
4. The third condition:

$$\lim_{w\to\infty} w^2 \operatorname{Re}\{W(jw)\} = \lim_{w\to\infty} \frac{Kw^2}{T^2w^2+1} = \frac{K}{T^2} > 0.$$

**SPR transfer function is a function with property of the first order block, i.e. relative degree less than 2 (0 or 1)**

## 6.3. Dynamic error model with measurable output

***Remark 6.5. One-syllable words about adaptation algorithm and SPR transfer functions***




*Error model* $\varepsilon(t) = W(s)\left[\tilde{\theta}^T(t)\omega(t)\right]$

*Adaptation algorithm* $\dot{\tilde{\theta}}(t) = -\dot{\hat{\theta}}(t) = -\gamma\,\omega(t)\varepsilon(t)$

## 6.3. Dynamic error model with measurable output

**Solution #1**

### Summary and Discussion

| | |
|---|---|
| *Error Model* | $\varepsilon = W(s)\left[\tilde{\theta}^T \omega\right],$ |
| *Adaptation Algorithm* | $\dot{\hat{\theta}} = \gamma \omega \varepsilon,$ |

where $W(s)$ is an SPR transfer function.

**SPR condition is quite restrictive and can narrow practical meaning of the problem**

## 6.3. Dynamic error model with measurable output

**Solution #2 Augmented error algorithm**

Consider error model

$$\varepsilon = W(s)\left[\tilde{\theta}^T \omega\right]$$

and introduce augmentation signal

$$\hat{\varepsilon} = \varepsilon - \hat{\theta}^T W(s)[\omega] + W(s)\left[\hat{\theta}^T \omega\right]. \tag{6.22}$$

Substitution of error model into (6.22) gives static error model **!!!**

$$\hat{\varepsilon} = \tilde{\theta}^T W(s)[\omega]. \tag{6.23}$$

Adaptation algorithm (see section **6.1. Static error model**)

$$\dot{\hat{\theta}} = \gamma W(s)[\omega]\hat{\varepsilon}. \tag{6.24}$$

## 6.3. Dynamic error model with measurable output

**Solution #2 Augmented error algorithm**

### Summary and Discussion

| | |
|---|---|
| *Error Model* | $\varepsilon = W(s)\left[\tilde{\theta}^T\omega\right],$ |
| *Augmented error* | $\hat{\varepsilon} = \varepsilon - \hat{\theta}^T W(s)[\omega] + W(s)\left[\hat{\theta}^T\omega\right],$ |
| *Adaptation Algorithm* | $\dot{\hat{\theta}} = \gamma W(s)[\omega]\hat{\varepsilon}.$ |

Proved by the Swapping lemma:

$$W(s)\left[\hat{\theta}^T\omega\right] = \hat{\theta}^T W(s)[\omega] - W_C(s)\left[W_b(s)\left[\omega^T\right]\dot{\hat{\theta}}\right]$$

with $W_C(s) = C(Is - A)^{-1}$, $W_b(s) = (Is - A)^{-1} b$ are the transfer matrices.

## 6.3. Dynamic error model with measurable output

**Example 6.6. The problem of output adaptive control**

<u>Problem statement</u>

Let a plant be described by

$$\ddot{y} + a_1\dot{y} + a_0 y = b_0 u \tag{6.25}$$

with **unknown** parameters $a_0, a_1$, known $b_0$ and unmeasurable state $\dot{y}$, known input $u$ and output $y$.

The objective is to design a control $u$ such that

$$\lim_{t\to\infty}\|y_M - y\| = 0, \tag{6.26}$$

where $y_M$ is the output of reference model

$$\ddot{y}_M + a_{M1}\dot{y}_M + a_{M0}y_M = b_{M0}g \tag{6.27}$$

with reference signal $g$.

## 6.3. Dynamic error model with measurable output

Solution

1. Obstacle of unmeasurable state.

   Apply first order (*n-1*th) filter

$$\frac{1}{s+k}, \quad k>0$$

to the plant equation:

$$\frac{1}{s+k}\left[\ddot{y}+a_1\dot{y}+a_0y\right]=b_0\frac{1}{s+k}[u]$$

$$\Downarrow$$

$$\frac{s \pm k}{s+k}[\dot{y}]+a_1\frac{s \pm k}{s+k}[y]+a_0\frac{1}{s+k}[y]=b_0\frac{1}{s+k}[u]$$

$$\dot{y}=(k-a_1)y+\left(a_1k-k^2-a_0\right)\frac{1}{s+k}[y]+b_0\frac{1}{s+k}[u]$$

## 6.3. Dynamic error model with measurable output

Solution

$$\dot{y}=\underbrace{(k-a_1)}_{\theta_1^*}\underbrace{y}_{\omega_1}+\underbrace{\left(a_1k-k^2-a_0\right)}_{\theta_2^*}\underbrace{\frac{1}{s+k}[y]}_{\omega_2}+\underbrace{b_0}_{\theta_3^*}\underbrace{\frac{1}{s+k}[u]}_{\omega_3}$$

$$\Downarrow$$

$$\dot{y}=\theta^{*T}\omega$$

**The derivative $\dot{y}$ is still not accessible, however presentable in the useful form of linear regression**

## 6.3. Dynamic error model with measurable output

Solution

2. Obstacle of unknown parameters.

**Main idea of solution is to reduce the problem to the error model.**

**Then to get the adaptation algorithm.**


## 6.3. Dynamic error model with measurable output

Solution

2. Obstacle of unknown parameters.

Evaluate the **second** time derivative of error $\varepsilon = y_M - y$ in view of plant $\ddot{y} + a_1\dot{y} + a_0 y = b_0 u$

and reference model $\ddot{y}_M + a_{M1}\dot{y}_M + a_{M0}y_M = b_{M0}g$ :

$$\ddot{\varepsilon} = \ddot{y}_M - \ddot{y} = -a_{M1}\dot{y}_M - a_{M0}y_M + b_{M0}g + a_1\dot{y} + a_0 y - b_0 u$$

$$\ddot{\varepsilon} = -a_{M1}\left(\dot{y}_M \pm \dot{y}\right) - a_{M0}\left(y_M \pm y\right) + b_{M0}g + a_1\dot{y} + a_0 y - b_0 u =$$

$$-a_{M1}\dot{\varepsilon} - a_{M1}\dot{y} - a_{M0}\varepsilon - a_{M0}y + b_{M0}g + a_1\dot{y} + a_0 y - b_0 u$$

$$\Downarrow$$

$$\varepsilon = \frac{1}{s^2 + a_{M1}s + a_{M0}}\left[-a_{M1}\dot{y} - a_{M0}y + b_{M0}g + a_1\dot{y} + a_0 y - b_0 u\right]$$

## 6.3. Dynamic error model with measurable output

Solution

$$\varepsilon = \underbrace{\frac{1}{s^2 + a_{M1}s + a_{M0}}}_{W_M(s)} \left[ -a_{M1}\dot{y} - a_{M0}y + b_{M0}g + a_1\dot{y} + a_0 y - b_0 u \right]$$

$$\Downarrow$$

$$\varepsilon = W_M(s)\left[ (a_1 - a_{M1})\dot{y} + (a_0 - a_{M0})y + b_{M0}g - b_0 u \right]$$

$$\varepsilon = W_M(s)\left[ (a_1 - a_{M1})\dot{y} + (a_0 - a_{M0})y + b_{M0}g - b_0 u \right]$$

$$\Downarrow \qquad \dot{y} = \theta^{*T}\omega$$

$$\varepsilon = W_M(s)\left[ \theta^T\omega + b_{M0}g - b_0 u \right]$$

## 6.3. Dynamic error model with measurable output

Solution

$$\varepsilon = W_M(s)\left[ \theta^T\omega + b_{M0}g - b_0 u \right]$$

where $\omega = col\left( y, \ \frac{1}{s+k}[y], \ \frac{1}{s+k}[u] \right)$

$$\theta = col\left( k - a_1 + a_0 - a_{M0}, \ a_1 k - k^2 - a_0, \ b_0 \right)$$

*Adjustable control* $$u = \frac{1}{b_0}\left[ \hat{\theta}^T\omega + b_{M0}g \right] \qquad (6.28)$$

*Error model* $$\varepsilon = W_M(s)\left[ \tilde{\theta}^T\omega \right] \qquad (6.29)$$


with parametric errors $\tilde{\theta} = \theta - \hat{\theta}$.

## 6.3. Dynamic error model with measurable output

Solution

$$\varepsilon = W_M(s)\left[\tilde{\theta}^T\omega\right]$$

$$\Downarrow$$

where

*Augmented error* $$\hat{\varepsilon} = \varepsilon - \hat{\theta}^T W_M(s)[\omega] + W_M(s)\left[\hat{\theta}^T\omega\right], \qquad (6.30)$$

*Adaptation Algorithm* $$\dot{\hat{\theta}} = \gamma W_M(s)[\omega]\hat{\varepsilon}. \qquad (6.31)$$

## 6.3. Dynamic error model with measurable output

Solution Summary

*Adjustable control* $$u = \frac{1}{b_0}\left[\hat{\theta}^T\omega + b_{M0}g\right] \qquad (6.28)$$

*Augmented error* $$\hat{\varepsilon} = \varepsilon - \hat{\theta}^T W_M(s)[\omega] + W_M(s)\left[\hat{\theta}^T\omega\right], \qquad (6.30)$$

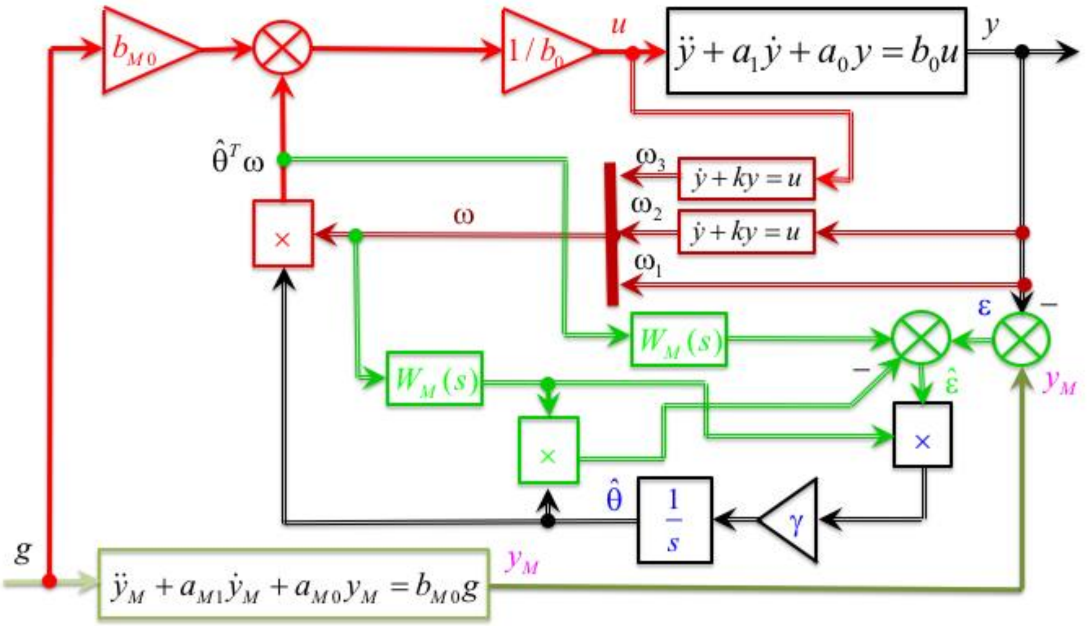
*Adaptation Algorithm* $$\dot{\hat{\theta}} = \gamma W_M(s)[\omega]\hat{\varepsilon}. \qquad (6.31)$$

*Error* $$\varepsilon = y_M - y$$

*Regressor with filters* $$\omega = col\left(y, \frac{1}{s+k}[y], \frac{1}{s+k}[u]\right)$$

## 6.3. Dynamic error model with measurable output

General Scheme




## 6.3. Dynamic error model with measurable output

Simulation results

*Plant*

$$\ddot{y}+a_1\dot{y}+a_0y=u$$

*Unknown parameters*

$$a_0=1,\ a_1=2$$

*Reference model*

$$\ddot{y}_M+5\dot{y}_M+6y_M=6g$$

*Adaptation gain*

$$\gamma=1000$$

*Reference transfer function (with unity denominator)*

$$W_M(s)=\frac{1}{s^2+5s+6}$$

*Reference* $g(t)=\sin 4t+2$

## 6.3. Dynamic error model with measurable output

Simulation results

*Regressor*

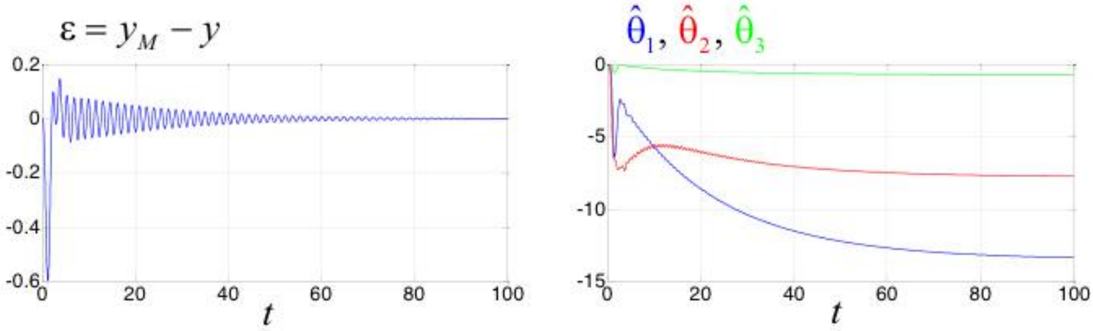
$$\omega = col\left( y, \frac{1}{s+8}[y], \frac{1}{s+8}[u] \right)$$

*Augmented error* $\hat{\varepsilon} = \varepsilon - \hat{\theta}^T \frac{1}{s^2+5s+6}[\omega] + \frac{1}{s^2+5s+6}\left[\hat{\theta}^T \omega\right],$

*Error* $\varepsilon = y_M - y$

## 6.3. Dynamic error model with measurable output

Simulation results



Digital twins

# Cyber-physical systems Digital twins

Alexey Margun
alexeimargun@gmail.com

## Digital twins

**Digital Twin** is a software analogue of a physical device that simulates internal processes, technical characteristics and behavior of a real object under the influence of the environment.


- Online copy of a real technical system (digital shadow)
- Offline modeling of technical systems

# Digital twins

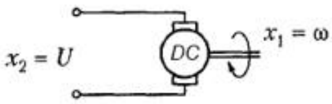
**Problems:**

- Unknown parameters
- Parameters changing
- Absence of sensors
- External noises and disturbances

**Possible solutions:**

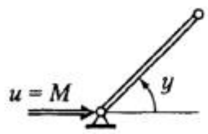
- Identification of unknown parameters
- Observers instead of sensors

# Input-output model




$T\dot{x}_1(t) + x_1(t) = kx_2(t)$

$x_1(t) = k\left(1 - e^{-t/T}\right)x_2$




$J\ddot{y}(t) = u(t)$

Laplace transformation ($p = \frac{d}{dt}$): $\dot{x}_1 = px_1, \int x = \frac{x}{p}$.

Plant model:

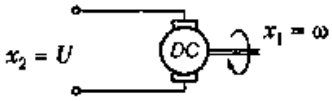
$$a(p)y(t) = b(p)u(t),$$

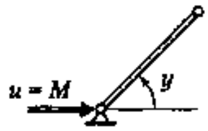
$a(p) = a_o p^n + a_1 p^{n-1} + \cdots + a_{n-1}p + a_n$ is a characteristic polynomial

$b(p) = b_o p^m + b_1 p^{m-1} + \cdots + b_{m-1}p + b_m$.

# Input-output model

$y(t) = W(p)u(t), W(p) = \frac{b(p)}{a(p)}$ is a transfer function




$$y(t) = \frac{k}{Tp+1}u(t)$$

$$\rightarrow \boxed{\frac{k}{Tp+1}} \rightarrow y(t)$$

$$y(t) = \frac{1}{p^2}u(t)$$

$$u(t) \rightarrow \boxed{\frac{1}{p^2}} \rightarrow y(t)$$

# State space model

All linear differential equations ca be written as

$$\dot{x}_1 = a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n + b_1u,$$
$$\dot{x}_2 = a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n + b_2u,$$
$$\dots$$
$$\dot{x}_n = a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n + b_nu,$$
$$y(t) = c_1x_1(t) + c_2x_2(t) + \dots + c_nx_n(t).$$

In matrix representation:

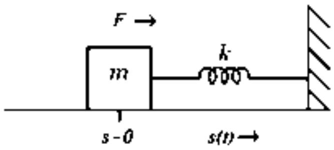
$$\dot{x} = Ax + Bu,$$
$$y = Cx$$

$$x = \begin{vmatrix} x_1 \\ \dots \\ x_n \end{vmatrix}, A = \begin{vmatrix} a_{11} & \dots & a_{1n} \\ \dots & \dots & \dots \\ a_{n1} & \dots & a_{nn} \end{vmatrix}, \mathrm{B} = \begin{vmatrix} b_1 \\ \dots \\ b_n \end{vmatrix}, C = [c_1 \quad \dots \quad c_n]$$

## State space model. Example

**Dynamic equations:**

$$\dot{s} = v$$
$$m\dot{v} = F - ks - hv$$




**State space model:**

$$\dot{x} = Ax + Bu,$$
$$y = Cx$$

Let us choose state vector as $x = \begin{vmatrix} s \\ v \end{vmatrix}$.

$$A = \begin{vmatrix} 0 & 1 \\ -k/m & -h/m \end{vmatrix}, B = \begin{vmatrix} 0 \\ 1/m \end{vmatrix}, C = |1 \quad 0|.$$

$$x(t) = x_{free} + x_{forced} = e^{At}x_0 + \int_0^t e^{A(t-\tau)}Bu(\tau)d\tau$$

## State space model. Change of coordinates

Consider new state vector:

$$x^* = Px$$

$P$ is a transformation matrix, $\det P \neq 0$.

Inverse transformation:

$$x = P^{-1}x^*$$

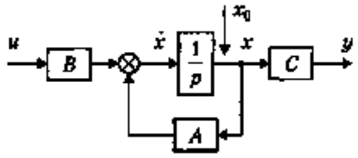
Model in new coordinates:

$$\begin{matrix} \dot{x}^* = A^*x^* + B^*u \\ y^* = C^*x^* \end{matrix}, A^* = PAP^{-1}, B^* = PB, C^* = CP^{-1}$$

Characteristic polynomial and poles of the system don't changes.

# State space model

**Modeling scheme**




**Transformation to input-output form:**

$$W(p) = C(pI - A)^{-1}B,$$

$$I = \begin{vmatrix} 1 & 0 & 0 \\ 0 & \dots & 0 \\ 0 & 0 & 1 \end{vmatrix}$$

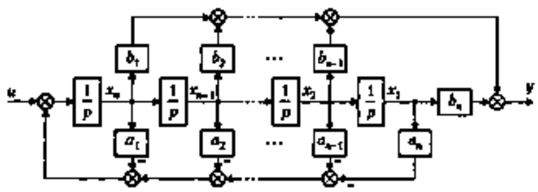
**Characteristic polynomial:**

$$\det(pI - A) = 0$$

# State space model

Transformation to state space model:

$$W(p) = \frac{b_1 p^{n-1} + \dots + b_{n-1}p + b_n}{p^n + a_1 p^{n-1} + \dots + b_{n-1}p + b_n}$$



Canonical controlled form:

$\dot{x}_1 = x_2$

$\dot{x}_2 = x_3$

...

$\dot{x}_n = -a_n x_1 - a_{n-1} x_2 - \dots - a_1 x_n + u$

$y = b_n x_1 + b_{n-1} x_2 + \dots + b_1$

$$A^* = \begin{vmatrix} 0 & | & I \\ & a^T & \end{vmatrix} = \begin{vmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 \\ -a_n & -a_{n-1} & \dots & & -a_1 \end{vmatrix}, B^* = \begin{vmatrix} 0 \\ 0 \\ \dots \\ 0 \\ 1 \end{vmatrix}, C^{*T} = \begin{vmatrix} b_n \\ b_{n-1} \\ \dots \\ b_2 \\ b_1 \end{vmatrix}$$

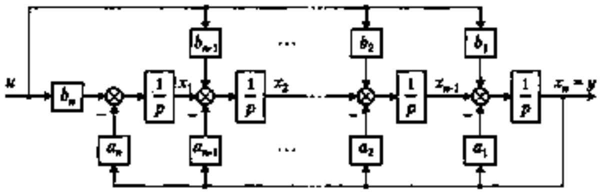
Transformation matrix: $P = U^* U^{-1}$ , $U$, $U^*$ are controllability matrices of canonical and original model

$$U = [B | AB | \dots | A^{n-1}B]$$

# State space model

Transformation to state space model:

$$W(p) = \frac{b_1 p^{n-1} + \dots + b_{n-1}p + b_n}{p^n + a_1 p^{n-1} + \dots + b_{n-1}p + b_n}$$




Canonical observed form:

$\dot{x}_1 = -a_n x_n + b_n u$

$\dot{x}_2 = x_1 - a_{n-1} x_n + b_n u$

$\dot{x}_n = x_{n-1} - a_1 x_n + b_1 u$

$y = x_n$

$$A^* = \begin{vmatrix} 0^T | \\ \dots \ | \ a \\ I | \end{vmatrix} = \begin{vmatrix} 0 & 0 & \dots & 0 & -a_n \\ 1 & 0 & \dots & 0 & -a_{n-1} \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 & -a_2 \\ 0 & 0 & \dots & 1 & -a_1 \end{vmatrix}, B^* = \begin{vmatrix} b_n \\ b_{n-1} \\ \dots \\ b_2 \\ b_1 \end{vmatrix}, C^{*T} = \begin{vmatrix} 0 \\ 0 \\ \dots \\ 0 \\ 1 \end{vmatrix}$$

Transformation matrix: $P = (Q^*)^{-1} Q$ , $Q, Q^*$ are observability matrices of canonical and original model

$$Q^T = [C \,|\, CA \,|\, \dots \,|\, CA^{n-1}]$$

# Identification. Scalar example

**Identification** is a set of methods for constructing mathematical models of a dynamic systems from observational data.

Consider plant:

$$y(t) = \theta^* u(t)$$

$u(t)$ is a scalar input,
$y(t)$ is a scalar output,
$\theta^*$ is an unknown parameter.
The obvious solution :

$$\theta = \frac{y(t)}{u(t)}$$

## Identification. Scalar example

Consider plant:

$$y(t) = \theta^* u(t)$$

$u(t)$ is a scalar input,
$y(t)$ is a scalar output,
$\theta^*$ is an unknown parameter.
The obvious solution :

$$\theta = \frac{y(t)}{u(t)}$$

Doesn't work if $u = 0$.
Hardly calculated if $u \to 0$.
High influence of noises.

## Online estimation

Let $\theta$ is an estimate of $\theta^*$.

Parallel model:

$$\hat{y}(t) = \theta u(t)$$

Error:

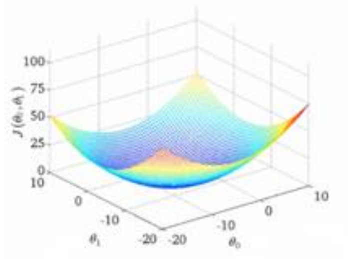
$$e = y - \hat{y} = y - \theta u$$

Consider functional:

$$J(\theta) = \frac{e^2}{2} = \frac{(y - \theta u)^2}{2}$$

Goal: minimize $J(\theta)$

# Online estimation



Let denote:

$$\nabla f(x) = \begin{bmatrix} \frac{\partial f(x)}{\partial x_1} \\ \dots \\ \frac{\partial f(x)}{\partial x_n} \end{bmatrix}$$ is a gradient of $f(x)$.

Lemma. If $J \in C^1$ and is convex on $R^n$ than $\theta^*$ is a global minimum if

$$\nabla J(\theta^*) = 0$$

Therefore, we need to solve equation $\nabla J(\theta^*) = 0$ with respect to the $\theta^*$

# Gradient search. Discrete

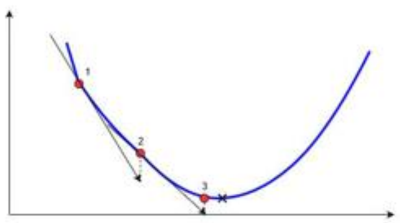
The search for the minimum is in the direction of reducing the function $d_k = -\nabla J(\theta_k)$

Identification algorithm:

$$\theta_{k+1} = \theta_k + \lambda_k d_k = \theta_k - \lambda_k \nabla J(\theta_k),$$
$$k = 0, 1, 2, \dots$$

$\lambda_k$ is a step size

$\theta_k$ is an estimate of $\theta$ on $k$-th step.

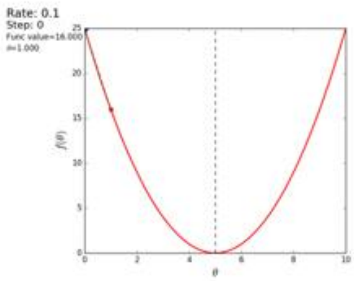


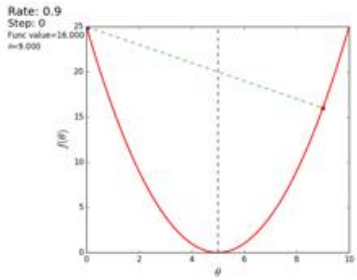
# Gradient search. Discrete

Example:

$$y = (\theta - 5)^2$$
$$\frac{\partial y}{\partial \theta} = 2(\theta - 5)$$

Initial value $\theta = 0$.




# Gradient search. Continuous

Rewrite algorithm as:

$$\frac{\theta_{k+1} - \theta_k}{\lambda_k} = -\nabla J(\theta)$$

If step is infinite small: $\lim_{\lambda_k \to 0} \frac{\theta_{k+1} - \theta_k}{\lambda_k} = \dot{\theta}$

Algorithm takes the form:

$$\dot{\theta} = -\gamma \nabla J(\theta)$$

$\gamma > 0$ is a coefficient that regulates convergence speed

For scalar case

$$\dot{\theta} = -\gamma \nabla J(\theta) = \gamma(y - \theta u)u = \gamma e u, \theta(0) = \theta_0$$

# Gradient search. Continuous

Consider estimation error:

$$\tilde{\theta} = \theta^* - \theta$$

Error transient:

$$\tilde{\theta}(t) = e^{-\gamma \int_0^t u^2(\tau)d\tau}\tilde{\theta}(0)$$

If $u = 0$ or $u = e^{-t}$, $\tilde{\theta}(t)$ will not converges to zero.

If $u^2 = \frac{1}{1+t}$, $\tilde{\theta}(t)$ asymptotically converges to zero.

$\tilde{\theta}(t)$ exponentially converges to zero if persistent excitation condition holds:

$$\int_t^{t+T_0} u^2(\tau)d\tau \geq \alpha_0 T_0, \forall t \geq 0$$

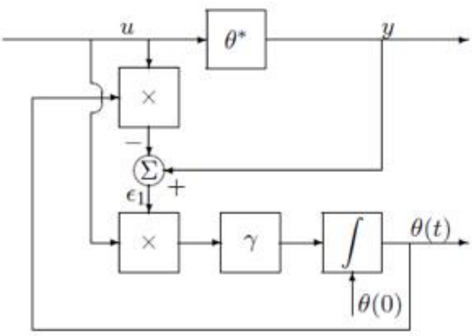
where $\alpha_0, T_0 > 0$

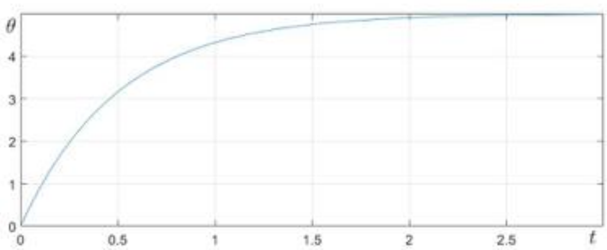
# Gradient search. Example

Consider the system $y = \theta u, \theta = 5$

Identification algorithm: $\dot{\theta} = -\gamma \nabla J(\theta) = \gamma(y - \theta u)u = \gamma e u$

$\theta(0) = \theta_0, \gamma = 2$



# Gradient search. Normalization

For system $y(t) = \theta^* u(t)$ with unbounded $y$ and $u$ problem

$$\min_{\theta} J = \min_{\theta} \frac{(y - \theta u)^2}{2}$$

can become hard for computing.

Solution is a normalization

$$\bar{y}(t) = \theta^* \bar{u}(t),$$

$$\bar{y}(t) = \frac{y}{m}, \bar{u}(t) = \frac{u}{m}, m^2 = 1 + u^2$$

Gradient search:

$$\dot{\theta} = \gamma \bar{e} \bar{u}, \gamma > 0.$$

In origin coordinates:

$$\dot{\theta} = \frac{\gamma e u}{m^2}$$

# Gradient search. Two unknown

Consider system

$$\dot{x} = -ax + bu, x(0) = x_0,$$

$$\dot{x} = \theta^T \phi, \theta = [a \quad b]^T, \phi = [-x \quad u]$$

where $a > 0$ and $b$ are unknown constants to be identified.

Parallel model:

$$\dot{\hat{x}} = -\hat{a}\hat{x} + \hat{b}\hat{u}, \hat{x}(0) = \hat{x}_0$$

Error:

$$e = x - \hat{x}$$

Functional:

$$J(\theta) = \frac{e^2}{2}$$

$$\dot{\theta} = \gamma \nabla J(\theta), \dot{\hat{a}} = -\gamma_1 e x, \dot{\hat{b}} = \gamma_2 e u$$

# Gradient search. Two unknown

If $\dot{x}$ is unmeasured.

Rewrite system:

$$\dot{x} = -a_m x + (a_m - a)x + bu \text{ или } x = \frac{1}{p+a_m}[(a_m - a)x + bu]$$

$a_m > 0$ is chosen by developer.

$$x = \theta^{*T}\phi,$$

$$\theta^* = [b, a_m - a]^T, \phi = \left[\frac{1}{p+a_m}u, \frac{1}{p+a_m}x\right]^T$$

Error:

$$e = x - \hat{x}$$

Serial-parallel model:

$$\dot{\hat{x}} = -a_m\hat{x} + (a_m - \hat{a})x + \hat{b}u \text{ или } \hat{x} = \frac{1}{p+a_m}[(a_m - \hat{a})x + \hat{b}u]$$

$$J = \frac{e^2}{2}$$

$$\theta = [a \quad b]^T$$

$$\dot{\theta} = \gamma\nabla J(\theta), \dot{\hat{a}} = -\gamma_1 ex, \dot{\hat{b}} = \gamma_2 eu$$

# Gradient search. Two unknown

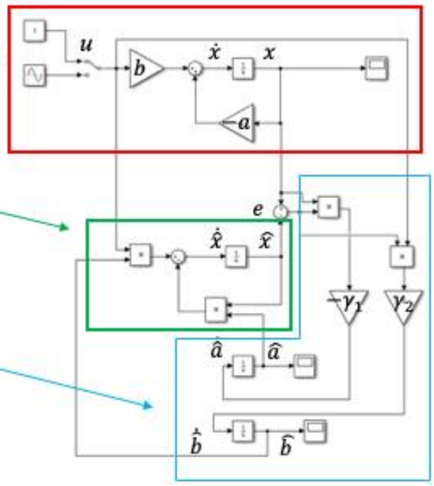
System: $\dot{x} = -ax + bu$.

Parallel model: $\dot{\hat{x}} = -a\hat{x} + \hat{b}u$.

Identification algorithm:

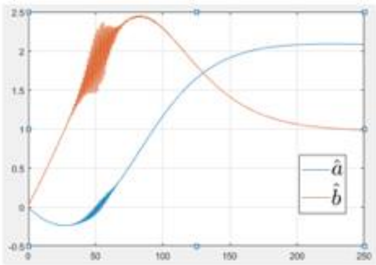
$\dot{\hat{a}} = -\gamma_1 ex$,

$\dot{\hat{b}} = \gamma_2 eu$

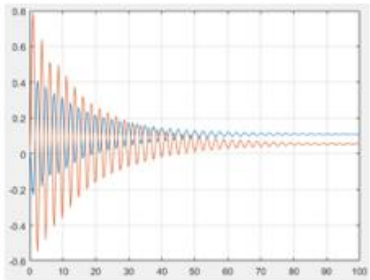


# Gradient search. Two unknown

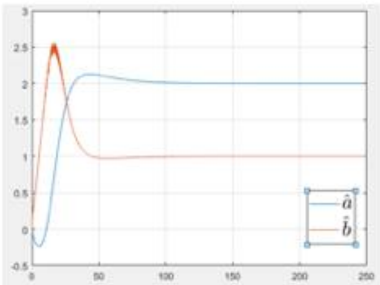
$u = \sin 5t$
$\gamma_{1,2} = 1$



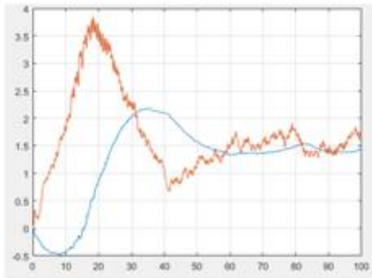

$u = 1(t)$
$\gamma_{1,2} = 5$



$\gamma_{1,2} = 5$




Белый шум



# Gradient search. Linear dynamic system

**Linear dynamic system:**

$$\dot{x} = A \quad x + B \quad u$$
$$A \quad \in R^{nxn}, B \quad \in R^{n}, x \in R^{n}$$

**Error:**

$$e = x - \hat{x}$$

**Functional:**

$$J = \frac{e^T e}{2}$$

**Parallel model:**

$$\dot{\hat{x}} = \hat{A} \quad \hat{x} + \hat{B} \quad u, \hat{x} \in R^{n}$$
$$\dot{\hat{A}} = \gamma_1 e x^T, \dot{\hat{B}} = \gamma_2 e u^T$$

**or serial-parallel model**

$$\dot{x} = A_m x + (A \quad - A_m)x + B \quad u, A_m \in R^{nxn}$$
$$\dot{\hat{x}} = A_m \hat{x} + (\hat{A} \quad - A_m)\hat{x} + \hat{B} \quad u$$
$$\dot{\hat{A}} = \gamma_1 e x^T, \dot{\hat{B}} = \gamma_2 e u^T$$

# System parametrization

Consider plant:

$$y^{(n)} + a_{n-1}y^{(n-1)} + \dots + a_0 y = b_{n-1}u^{(n-1)} + b_{n-2}u^{(n-2)} + \dots + b_0 u$$

Rewrite all parameters as vector

$$\theta^* = [b_{n-1}, b_{n-2}, \dots, b_0, a_{n-1}, a_{n-2}, \dots, a_0]^T$$

Rewrite input/output signals and their derivatives:

$$Y = \left[u^{(n-1)}, u^{(n-2)}, \dots, u, -y^{(n-1)}, -y^{(n-2)}, \dots, -y\right]^T =$$
$$= [\alpha_{n-1}^T(p)u, -\alpha_{n-1}^T(p)y]^T, \alpha_i(p) = \left[p^i, p^{i-1}, \dots, 1\right]^T$$

Therefore, we can rewrite system equation:

$$y^{(n)} = \theta^{*T} Y$$

# System parametrization

If derivatives $y^{(n)} = \theta^{*T} Y$ are unmeasured

Apply stable filter $\frac{1}{\Lambda(p)}$ for both parts of equation, $\Lambda(p)$ is a Hurwitz polynomial:

$$z = \theta^{*T}\phi,$$

$$z = \frac{p^n}{\Lambda(p)} y, \phi = \left[\frac{\alpha_{n-1}^T(p)}{\Lambda(p)} u, -\frac{\alpha_{n-1}^T(p)}{\Lambda(p)} y\right]$$

$$\Lambda(p) = p^n + \lambda_{n-1}p^{n-1} + \dots + \lambda_0$$

All signals of filtered model are measured.

# System parametrization

Consider $\Lambda(p)$ as $\Lambda(p) = p^n + \lambda^T \alpha_{n-1}(p), \lambda = [\lambda^{n-1}, \dots, \lambda_0]^T$
In this case:

$$z = \frac{p^n}{\Lambda(p)} y = \frac{\Lambda(p) - \lambda^T \alpha_{n-1}(p)}{\Lambda(p)} y = y - \lambda^T \frac{\alpha_{n-1}(p)}{\Lambda(p)} y$$

$$y = z + \lambda^T \frac{\alpha_{n-1}(p)}{\Lambda(p)} y$$

$$z = \theta^{*T} \phi = \theta_1^{*T} \phi_1 + \theta_2^{*T} \phi_2, \theta_1^{*T} = [b_{n-1}, \dots, b_0], \theta_2^{*T} = [a_{n-1}, \dots, a_0],$$

$$\phi_1 = \frac{\alpha_{n-1}(p)}{\Lambda(p)} u, \phi_2 = -\frac{\alpha_{n-1}(p)}{\Lambda(p)} \mathrm{y}$$

$$y = \theta_1^{*T} \phi_1 + \theta_2^{*T} \phi_2 - \lambda^T \phi_2$$

$$y = \theta_\lambda^{*T} \phi, \qquad \theta_\lambda^{*T} = [\theta_1^{*T}, \theta_2^{*T}, -\lambda^T]$$

# State observers

**Observer** is an algorithm that allows to estimate the unmeasurable variables of the state vector.
Consider linear dynamic model:

$$\dot{x} = Ax + Bu,$$
$$y = C^T x$$

Parameters of system are known. Vector $x$ is unmeasured.
If $x_0$ is known, than algorithm

$$\dot{\hat{x}} = A\hat{x} + Bu, \hat{x}(0) = x_0$$

provide $\hat{x}(t) = x(t) \forall t \geq 0$.

## State observers

If $x_0$ is unknown and matrix $A$ is stable we can use observer:

$$\dot{\hat{x}} = A\hat{x} + Bu, \hat{x}(0) = \hat{x}_0$$

Consider observation error:

$$\tilde{x} = x - \hat{x}$$

Its dynamics satisfy equation:

$$\dot{\tilde{x}} = A\tilde{x}, \tilde{x}(0) = x(0) - \hat{x}(0)$$

Solution of error dynamic equation:

$$\tilde{x}(t) = e^{At}\tilde{x}(0)$$

Because of $A$ is stable $\tilde{x}$ exponentially converges to zero

## Luenberger observer

If $x_0$ is unknown and matrix $A$ is unstable or we need increase speed of convergence:

$$\dot{\hat{x}} = A\hat{x} + Bu + K(y - \hat{y}), \hat{x}(0) = \hat{x}_0,$$
$$\hat{y} = C^T\hat{x},$$

where $K$ is chosen by developer.

Dynamics of estimation error:

$$\dot{\tilde{x}} = (A - KC^T)\tilde{x}, \tilde{x}(0) = x(0) - \hat{x}(0)$$

So;ution of error dynamics equation:

$$\tilde{x}(t) = e^{(A-KC^T)t}\tilde{x}(0)$$

By tuning $K$ we ensure the stability of the error model and adjust its transient (overshoot, transient time, etc.)

# Luenberger observer. Example

**System:**

$$\dot{x} = \begin{bmatrix} -4 & 1 \\ -4 & 0 \end{bmatrix} x + \begin{bmatrix} 1 \\ 3 \end{bmatrix} u, x(0) = \begin{bmatrix} 14 \\ 0,5 \end{bmatrix}$$

$$y = [1 \quad 0] x.$$

Luenberger observer

$$\dot{\hat{x}} = \begin{bmatrix} -4 & 1 \\ -4 & 0 \end{bmatrix} \hat{x} + \begin{bmatrix} 1 \\ 3 \end{bmatrix} u + \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} (y - \hat{y}),$$

$$\hat{y} = [1 \quad 0] \hat{x}.$$

# Luenberger observer. Example

Denote $A_0 = A - KC^T = \begin{bmatrix} -4 & 1 \\ -4 & 0 \end{bmatrix} - \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} [1 \quad 0] = \begin{bmatrix} -4 - k_1 & 1 \\ -4 - k_2 & 0 \end{bmatrix}$.

Let we need speed of convergence faster than $e^{-5t}$.

In this case real part of $A_0$ eigenvalues should be less than $-5$.
Let $\lambda_1 = -6, \lambda_2 = -8$.

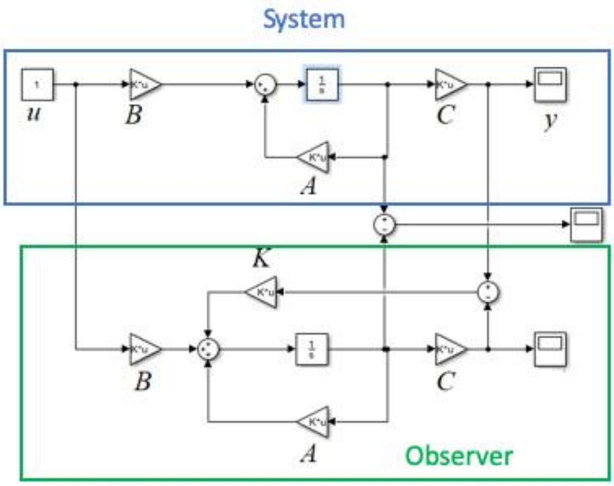
Therefore:

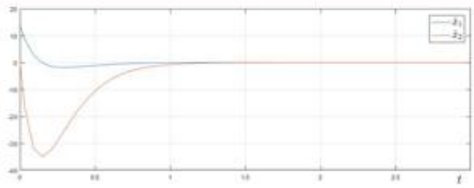
$$\det(pI - A_0) = p^2 + (4 + k_1)p + 4 + k_2 = (p + 6)(p + 8)$$

We can find:

$$k_1 = 10, k_2 = 44$$

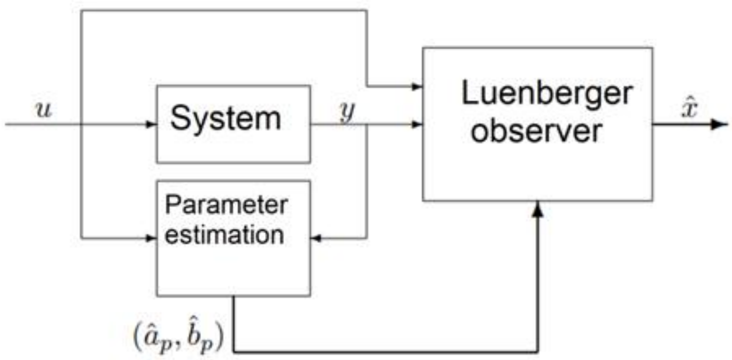
# Luenberger observer. Example



# Adaptive Luenberger observer

State vector is unmeasured.
System parameters are unknown.
Solution: simultaneously use the observer and the parameter estimation algorithm.



# Adaptive Luenberger observer

In state space form we need to estimate $n^2 + 2n$ parameters.
In input output form we need to estimate $n + m + 1 \leq 2n$ parameters.
Obtain transfer function:

$$C^T(pI - A)^{-1}B = \frac{b_{n-1}p^{n-1} + \ldots + b_1 p + b_0}{p^n + a_{n-1}p^{n-1} + \cdots + a_0}$$

Rewrite system in canonical observable form:

$$\dot{x}_\alpha = \begin{bmatrix} -a_p & \vdots & I_{n-1} \\ & \vdots & \cdots \\ & \vdots & 0 \end{bmatrix} x_\alpha + b_p u, y = [1 \ 0 \ \ldots 0] x_\alpha$$

$$a_p = [a_{n-1}, a_{n-2}, \ldots, a_0]^T, b_p = [b_{n-1}, b_{n-2}, \ldots, b_0]^T$$

# Adaptive Luenberger observer

Observer:

$$\dot{\hat{x}} = \hat{A}\hat{x} + \hat{b}_p u + K(y - \hat{y}), \hat{x}(0) = \hat{x}_0,$$

$$\hat{y} = [1\ 0 \ldots 0]\hat{x},$$

$$\hat{A} = \begin{bmatrix} -\hat{a}_p & \vdots & I_{n-1} \\ & \vdots & \cdots \\ & \vdots & 0 \end{bmatrix}, K = a^* - \hat{a}_p$$

$a^*$ is chosen such that

$$A^* = \begin{bmatrix} -a^* & \vdots & I_{n-1} \\ & \vdots & \cdots \\ & \vdots & 0 \end{bmatrix}$$

is stable, i.e. roots of $\det(pI - A^*) = 0$ have negative real part.

# Digital twins

It is necessary for development of digital twin:

- Build mathematical model of system
- Estimate unknown parameters with identification algorithm
- Build observer for state vector estimation
- Run obtained model in real time with the same input signal as a real system

Nonlinear control systems

# Nonlinear Control Systems

Zimenko Konstantin

## Nonlinear versus linear systems

### Linear systems

- Huge body of work in analysis and control of linear systems
- Most models currently available are linear (but most real systems are nonlinear...)


### Nonlinear systems

- Dynamics of linear systems are not rich enough to describe many commonly observed phenomena

Nonlinear systems can (sometime) be approximated by linear systems.
Nonlinear systems can (sometime) be "transformed" into linear systems. 2

# State-space model

State equation

$$\dot{x} = f(t, x, u)$$

Output equation

$$y = h(t, x, u)$$

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad u = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix}, \quad f(t, x, u) = \begin{bmatrix} f_1(t, x, u) \\ f_2(t, x, u) \\ \vdots \\ f_n(t, x, u) \end{bmatrix}$$

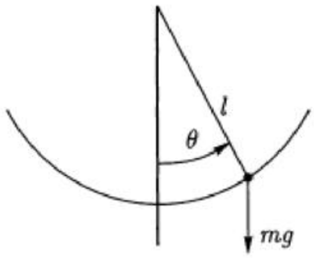
where $x \in \mathbb{R}^n$ is the state variable, $u \in \mathbb{R}^m$ is the input signal, and $y \in \mathbb{R}^q$ the output signal. The symbol $\dot{x} = \frac{dx}{dt}$ denotes the derivative of $x$ with respect to time $t$.

3

# Nonlinear systems: Example

Pendulum equation (equation of motion in the tangential direction)

$$ml\ddot{\theta} = -mg\sin\theta - kl\dot{\theta}.$$




State equations ($x_1 = \theta, x_2 = \dot{\theta}$)

$$\dot{x}_1 = x_2,$$
$$\dot{x}_2 = -\frac{g}{l}\sin x_1 - \frac{k}{m}x_2$$

Equilibrium points $(n\pi;0)$, $n = 0, \pm 1, \pm 2, \ldots$

4

# Nonlinear systems: Example

State equations (frictional resistance is neglected)

$$\dot{x}_1 = x_2,$$
$$\dot{x}_2 = -\frac{g}{l}\sin x_1$$

Equilibrium points $(n\pi;0)$, $n = 0, \pm 1, \pm 2, \ldots$

State equations (with friction and applied torque)

$$\dot{x}_1 = x_2,$$
$$\dot{x}_2 = -\frac{g}{l}\sin x_1 - \frac{k}{m}x_2 + \frac{1}{ml^2}T$$

where $T$ is the torque.

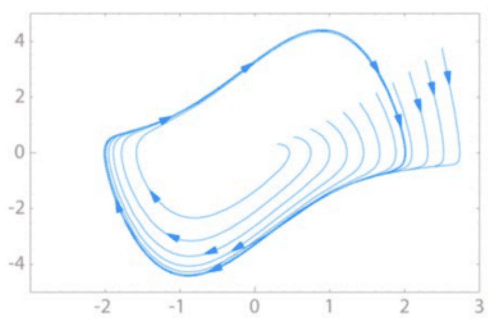
Equilibrium points $(\arcsin(T/mgl);0)$

5

# Nonlinear systems: Example

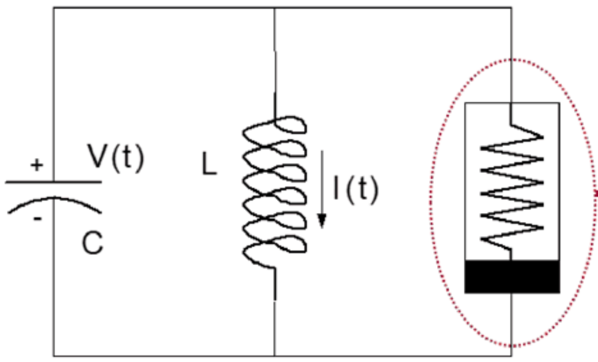
Robust oscillation

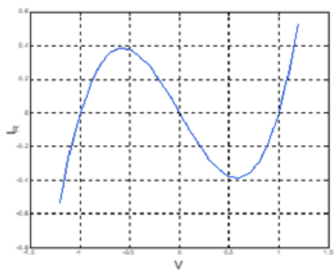
$$\dot{x}_1 = x_2,$$
$$\dot{x}_2 = -x_1 + \epsilon(1 - x_1^2)x_2$$




Van der Pol oscillator

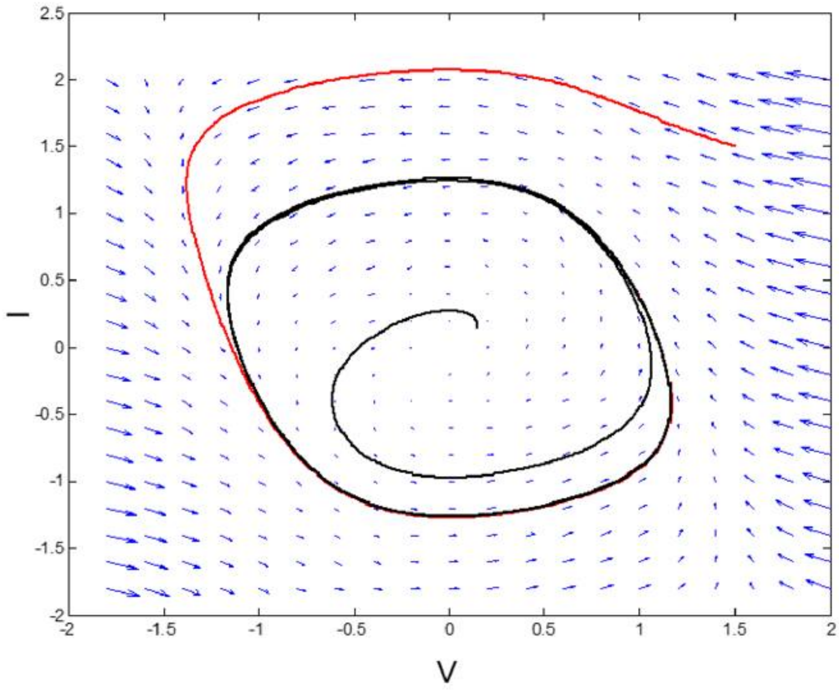
$$L\dot{I} = V$$
$$C\dot{V} = -I - I_R(V)$$



# Nonlinear systems: Example

Van der Pol oscillator: phase portrait



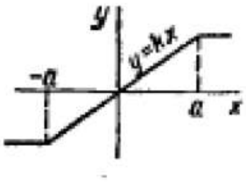

7

# Nonlinear phenomena

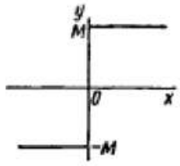
- Finite escape time
  (the state of unstable linear system goes to infinity as $t \to \infty$)
- Nonasimptotic stability (e.g. finite-time stability)
  (linear systems – infinite time of convergence)
- Multiple isolated equilibria
  (linear systems – only one isolated equilibrium point)
- Limit cycles
  (linear systems – system oscilates iff there is a pair of eigenvalues on the imaginary axis, which is a nonrobust condition)
- Subharmonic, harmonic, or almost-periodic oscillations
  (stable linear system under periodic input produces an output of the same frequency)
- Chaos
  (More complicated steady-state behavior)
- Multiple modes of behavior

8

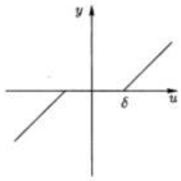
# Common nonlinearities



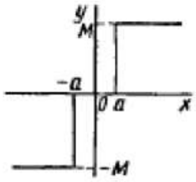
Saturation



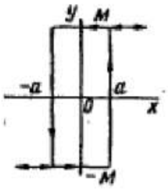
Relay



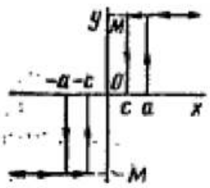
Dead zone



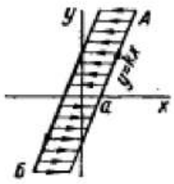
Relay with dead zone



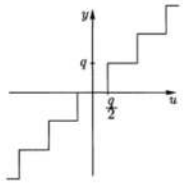
Relay with hysteresis



Three-position relay



Backslash



Quantization

9

# Qualitative behavior of linear systems

Linear second order system

$$\dot{x} = Ax, \;\; x \in \mathbb{R}^2, A \in \mathbb{R}^{2\times 2}$$

Apply a similarity transformation $M$ to $A$:

$$M^{-1}AM = J, \;\; M \in \mathbb{R}^{2\times 2}$$

where $J$ is the real *Jordan form* of $A$, which depending on the eigenvalues of $A$ may take one of the three forms

$$\begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}, \quad \begin{bmatrix} \lambda & k \\ 0 & \lambda \end{bmatrix}, \quad \begin{bmatrix} \alpha & -\beta \\ \beta & \alpha \end{bmatrix}$$

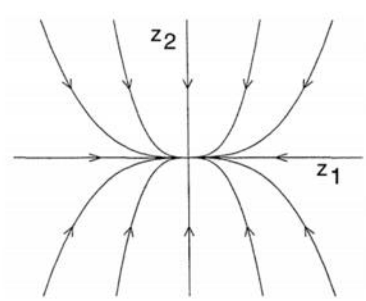
with $k$ being either 0 or 1.

Present a change of coordinates

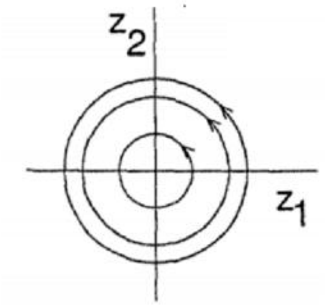
$$z = M^{-1}x$$
$$\dot{z} = M^{-1}\dot{x}$$

10

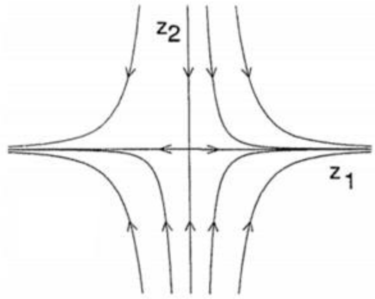
# Qualitative behavior of linear systems



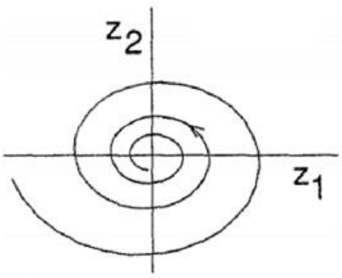

Stable node ($\lambda_{1,2} < 0$)



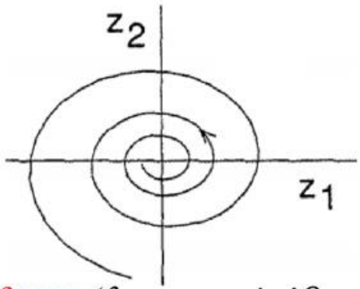

Center ($\lambda_{1,2} = \pm j\beta$)




Saddle point ($\lambda_2 < 0 < \lambda_1$)




Stable focus ($\lambda_{1,2} = a \pm j, a < 0$)

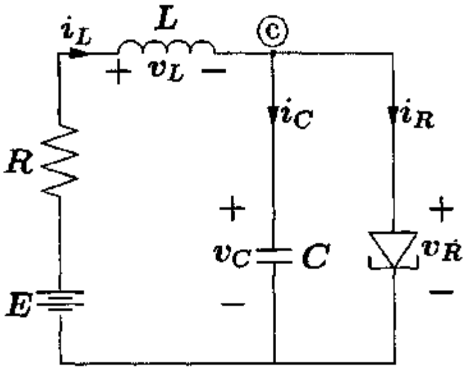



Stable focus ($\lambda_{1,2} = a \pm j\beta, a > 0$)

11

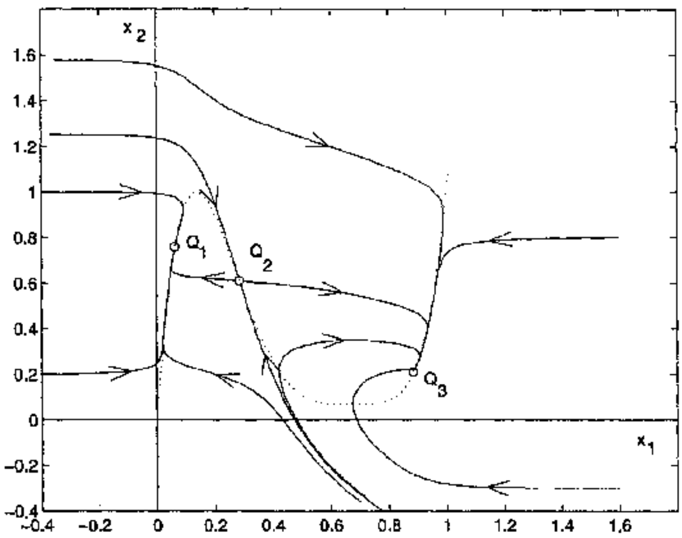
# Multiple equilibrium points

Tunnel-diode circuit




$$\dot{x}_1 = \frac{1}{C}[-h(x_1) - x_2]$$

$$\dot{x}_2 = \frac{1}{L}[-x_1 - Rx_2 + u]$$

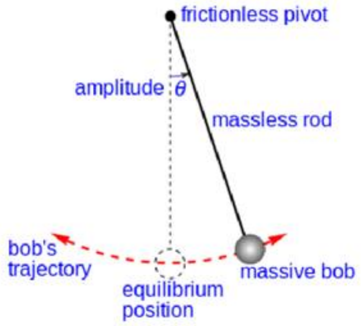


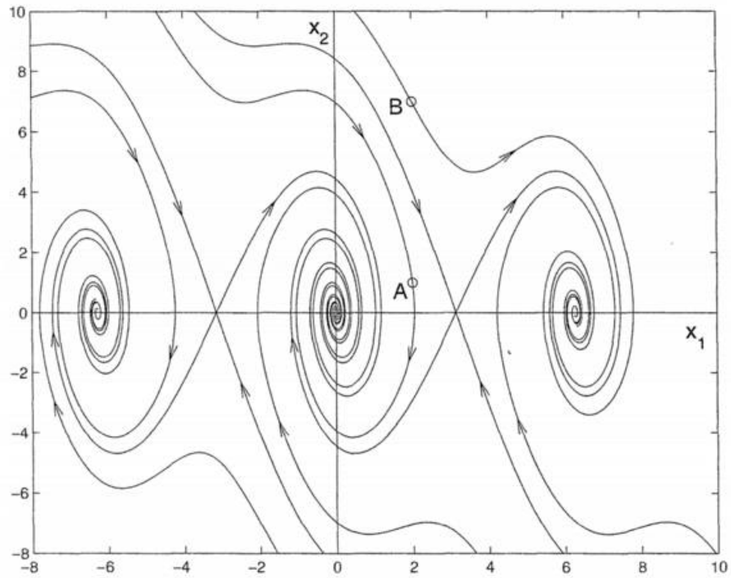

Phase portrait

12

# Multiple equilibrium points

Pendulum




Phase portrait of the pendulum equation

13

# Qualitative behavior near equilibrium

Consider autonomous system

$$\dot{x}_1 = f_1(x_1, x_2),$$
$$\dot{x}_2 = f_2(x_1, x_2)$$

where $f_1(x_1, x_2)$, $f_2(x_1, x_2)$ are continuously differeniable.

Let $p = (p_1, p_2)$ is the equilibrium point. Expanding the right-hand side into its Taylor series about the point $p$, obtain

$$\dot{x}_1 = f_1(p_1, p_2) + a_{11}(x_1 - p_1) + a_{12}(x_2 - p_2) + HOT,$$
$$\dot{x}_2 = f_2(p_1, p_2) + a_{21}(x_1 - p_1) + a_{22}(x_2 - p_2) + HOT,$$

where $HOT$ denotes high order terms and

$$a_{11} = \left.\frac{\partial f_1(x_1, x_2)}{\partial x_1}\right|_{x_1=p_1, x_2=p_2}, \quad a_{12} = \left.\frac{\partial f_1(x_1, x_2)}{\partial x_2}\right|_{x_1=p_1, x_2=p_2}$$
$$a_{21} = \left.\frac{\partial f_2(x_1, x_2)}{\partial x_1}\right|_{x_1=p_1, x_2=p_2}, \quad a_{22} = \left.\frac{\partial f_2(x_1, x_2)}{\partial x_2}\right|_{x_1=p_1, x_2=p_2}$$

Since $p = (p_1, p_2)$ is an equilibrium point

$$f_1(p_1, p_2) = f_2(p_1, p_2) = 0$$

14

## Qualitative behavior near equilibrium

Define

$$y_1 = x_1 - p_1, \quad \text{и} \quad y_2 = x_2 - p_2$$

and rewrite the state equation as

$$\dot{y}_1 = \dot{x}_1 = a_{11}y_1 + a_{12}y_2 + HOT$$
$$\dot{y}_2 = \dot{x}_2 = a_{21}y_1 + a_{22}y_2 + HOT$$

*HOT* is negligible in a small neighborhood of equilibrium point:

$$\dot{y}_1 = \dot{x}_1 = a_{11}y_1 + a_{12}y_2$$
$$\dot{y}_2 = \dot{x}_2 = a_{21}y_1 + a_{22}y_2$$

Rewriting in a vector form, obtain

$$\dot{y} = Ay$$

where

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \left.\begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix}\right|_{x=p} = \left.\frac{\partial f}{\partial x}\right|_{x=p}$$

15

## Example 1

Pendulum equation

$$\dot{x}_1 = x_2,$$
$$\dot{x}_2 = -10\sin x_1 - x_2.$$

Equilibrium points (0;0) и (π;0)

Jacobian

$$\frac{\partial f}{\partial x} = \begin{bmatrix} 0 & 1 \\ -10\cos x_1 & -1 \end{bmatrix}$$

Jacobian evaluated at the equilibrium point

$$A_1 = \begin{bmatrix} 0 & 1 \\ -10 & -1 \end{bmatrix}, \qquad \lambda_{1,2} = -0.5 \pm j3/12$$

$$A_2 = \begin{bmatrix} 0 & 1 \\ 10 & -1 \end{bmatrix}, \qquad \lambda_{1,2} = -3.7, 2.7$$

Equilibrium point (0;0) is a stable focus, equilibrium point (π;0) is a saddle point

16

# Example 2

Consider the system

$$\dot{x}_1 = -x_2 - \mu x_1(x_1^2 + x_2^2)$$
$$\dot{x}_2 = x_1 - \mu x_2(x_1^2 + x_2^2)$$

Jacobian at (0;0) has eigenvalues $\pm j$.

Transition to polar coordinates:

$$x_1 = r\cos\theta \quad \text{и} \quad x_2 = r\sin\theta$$

The system in polar coordinates

$$\dot{r} = -\mu r^3 \quad \text{и} \quad \dot{\theta} = 1$$

For $\mu > 0$ the equilibrium point (0;0) is a stable focus, for $\mu < 0$ is a unstable focus.

17

# Lyapunov function

Consider the system

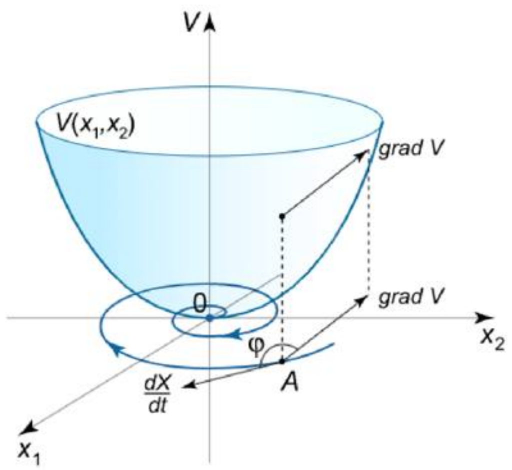
$$\dot{x} = f(x),$$

where $f : D \to R^n$ is locally Lipschitz

Let $p = 0$ is equilibrium point and $D \subset R^n$ is an open set, which contains $p$. Let $V : D \to R$ is a continuously differentiable function such, that

$$V(0) = 0 \text{ and } V(x) > 0 \text{ for } D\setminus\{0\}$$

If $\dot{V}(x) \le 0$ for $x \in D$, then $p = 0$ is stable.




If $\dot{V}(x) < 0$ for $x \in D\setminus\{0\}$, then $p = 0$ is asymptotically stable.

18

# Lyapunov function

Consider the system

$$\dot{x} = f(x), f(0) = 0.$$

Expanding the right-hand side into its Taylor series

$$\dot{x} = f(0) + \left.\frac{\partial f}{\partial x}\right|_{x=0} x + g(x) = Ax + g(x),$$

where

$$A = \left.\frac{\partial f}{\partial x}\right|_{x=0}.$$

Choose the candidate Lyapunov function in the form

$$V(x) = x^T P x, \quad P > 0$$

Then

$$\dot{V}(x) = \dot{x}^T P x + x^T P \dot{x} = \left[x^T A^T + g^T(x)\right] P x + x^T P[Ax + g(x)] = x^T \left(A^T P + PA\right) x + 2x^T P g(x) =$$
$$= -x^T Q x + 2x^T P g(x),$$

where $Q>0$ such, that

$$A^T P + PA = -Q \qquad \text{Lyapunov equation}$$

19

# Lyapunov function

Let

$$|g(x)| < \gamma |x|,$$

where $\gamma > 0$.

Since

$$x^T Q x \geq \lambda_{\min}(Q) x^T x = \lambda_{\min}(Q) |x|^2,$$

where $\lambda_{\min}(Q)$ is the smallest eigenvalue of the matrix $Q$, then

$$\dot{V}(x) \leq -\lambda_{\min}(Q)|x|^2 + 2\gamma \|P\| |x|^2 = -\left[\lambda_{\min}(Q) - 2\gamma \|P\|\right] x^2.$$

Lyapunov function derivative is negative if

$$\lambda_{\min}(Q) - 2\gamma\|P\| > 0 \Leftrightarrow \gamma < \frac{\lambda_{\min}(Q)}{2\|P\|}.$$

20

# Example

Consider the system

$$\dot{x} = ax^3$$

Linearization:

$$A = \left.\frac{\partial f}{\partial x}\right|_{x=0} = 3ax^2\big|_{x=0} = 0.$$

Choose the Lyapunov function

$$V(x) = x^2.$$

Then

$$\dot{V}(x) = 2ax^4.$$

The equilibrium point is:

1) stable, if $a = 0$;
2) asymptotically stable, if $a < 0$;
3) unstable, if $a > 0$.

21

# Stabilization: steady-state control

Consider the system

$$\dot{x} = f(x, u)$$

with desired equilibrium point $x = x_{ss}$

Steady-State Problem: Find steady-state control $u_{ss}$ s.t.

$$0 = f(x_{ss}, u_{ss})$$

$$x_\delta = x - x_{ss}, \qquad u_\delta = u - u_{ss}$$

$$\dot{x}_\delta = f(x_{ss} + x_\delta, u_{ss} + u_\delta) \stackrel{\text{def}}{=} f_\delta(x_\delta, u_\delta)$$

$$f_\delta(0, 0) = 0$$

$$u_\delta = \gamma(x_\delta) \quad \Rightarrow \quad u = u_{ss} + \gamma(x - x_{ss})$$

22

## State feedback stabilization

Nonlinear system

$$\dot{x} = f(x,u) \qquad [f(0,0)=0]$$
$$u = \gamma(x) \qquad [\gamma(0)=0]$$

Problem: stabilize the system at the origin

$$\dot{x} = f(x,\gamma(x))$$

where $f$ and $\gamma$ are locally Lipschitz functions

23

## Stabilization: linearization approach

$$\dot{x} = Ax + Bu$$

$$A = \frac{\partial f}{\partial x}(x,u)\bigg|_{x=0,u=0} ; \quad B = \frac{\partial f}{\partial u}(x,u)\bigg|_{x=0,u=0}$$

Closed-loop system:

$$\dot{x} = f(x,-Kx)$$

$$\begin{aligned}\dot{x} &= \left[\frac{\partial f}{\partial x}(x,-Kx) + \frac{\partial f}{\partial u}(x,-Kx)\,(-K)\right]_{x=0} x \\ &= (A-BK)x\end{aligned}$$

$(A-BK)$ is Hurwitz $\Rightarrow$ the origin is an exponentially stable equilibrium point

24

# Example: pendulum equation

$$\ddot{\theta} = -a \sin\theta - b\dot{\theta} + cT$$

Stabilize the pendulum at $\theta = \delta$

$$0 = -a \sin\delta + cT_{\text{ss}}$$

$$x_1 = \theta - \delta, \quad x_2 = \dot{\theta}, \quad u = T - T_{\text{ss}}$$

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -a[\sin(x_1 + \delta) - \sin\delta] - bx_2 + cu \end{aligned}$$

$$A = \left[ \begin{array}{cc} 0 & 1 \\ -a\cos(x_1 + \delta) & -b \end{array} \right]_{x_1=0} = \left[ \begin{array}{cc} 0 & 1 \\ -a\cos\delta & -b \end{array} \right]$$

25

# Example: pendulum equation

$$A = \left[ \begin{array}{cc} 0 & 1 \\ -a\cos\delta & -b \end{array} \right]; \quad B = \left[ \begin{array}{c} 0 \\ c \end{array} \right]$$

$$K = \left[ \begin{array}{cc} k_1 & k_2 \end{array} \right]$$

$$A - BK = \left[ \begin{array}{cc} 0 & 1 \\ -(a\cos\delta + ck_1) & -(b + ck_2) \end{array} \right]$$

$$k_1 > -\frac{a\cos\delta}{c}, \quad k_2 > -\frac{b}{c}$$

$$T = \frac{a\sin\delta}{c} - Kx = \frac{a\sin\delta}{c} - k_1(\theta - \delta) - k_2\dot{\theta}$$

26

## Feedback linearization

Consider the nonlinear system

$$\dot{x} = f(x) + G(x)u$$

$$f(0) = 0, \quad x \in R^n, \quad u \in R^m$$

Suppose there is a change of variables $z = T(x)$, defined for all $x \in D \subset R^n$, that transforms the system into the controller form

$$\dot{z} = Az + B\gamma(x)[u - \alpha(x)]$$

where $(A, B)$ is controllable and $\gamma(x)$ is nonsingular for all $x \in D$

$$u = \alpha(x) + \gamma^{-1}(x)v \quad \Rightarrow \quad \dot{z} = Az + Bv$$

27

## Feedback linearization

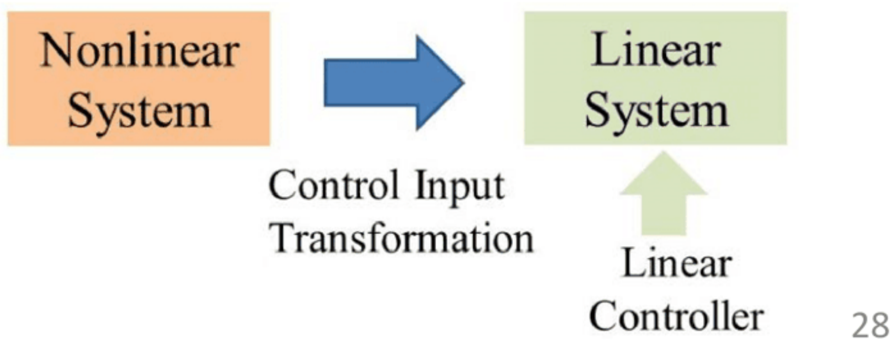
$$v = -Kz$$

Design $K$ such that $(A - BK)$ is Hurwitz

$$u = \alpha(x) - \gamma^{-1}(x)KT(x)$$

Closed-loop system in the $x$-coordinates:

$$\dot{x} = f(x) + G(x)\left[\alpha(x) - \gamma^{-1}(x)KT(x)\right]$$




28

# Feedback linearization

Closed-loop system:

$$u = \hat{\alpha}(x) - \hat{\gamma}^{-1}(x)K\hat{T}(x)$$

$$\dot{z} = (A - BK)z + B\delta(z)$$

$$\delta = \gamma[\hat{\alpha} - \alpha + \gamma^{-1}KT - \hat{\gamma}^{-1}K\hat{T}]$$

where $\hat{\alpha}, \hat{\gamma}, \hat{T}$ are nominal models of $\alpha, \gamma$ and $T$.

$$V(z) = z^T Pz, \quad P(A - BK) + (A - BK)^T P = -I$$

If $\|\delta(z)\| \le k\|z\|$ for all $z$, where

$$0 \le k < \frac{1}{2\|PB\|}$$

then the origin is globally exponentially stable

29

# Example: pendulum equation

$$\ddot{\theta} = -a\sin\theta - b\dot{\theta} + cT$$

$$x_1 = \theta - \delta, \quad x_2 = \dot{\theta}, \quad u = T - T_{ss} = T - \frac{a}{c}\sin\delta$$

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -a[\sin(x_1 + \delta) - \sin\delta] - bx_2 + cu \end{aligned}$$

$$u = \frac{1}{c}\{a[\sin(x_1 + \delta) - \sin\delta] - k_1x_1 - k_2x_2\}$$

$$A - BK = \begin{bmatrix} 0 & 1 \\ -k_1 & -(k_2 + b) \end{bmatrix}$$ is Hurwitz

30

# Example: pendulum equation

$$T = u + \frac{a}{c}\sin\delta = \frac{1}{c}\left[a\sin(x_1+\delta) - k_1x_1 - k_2x_2\right]$$

Let $\hat{a}$ and $\hat{c}$ be nominal models of $a$ and $c$

$$T = \frac{1}{\hat{c}}\left[\hat{a}\sin(x_1+\delta) - k_1x_1 - k_2x_2\right]$$

$$\dot{x} = (A - BK)x + B\delta(x)$$

$$\delta(x) = \left(\frac{\hat{a}c - a\hat{c}}{\hat{c}}\right)\sin(x_1+\delta_1) - \left(\frac{c-\hat{c}}{\hat{c}}\right)(k_1x_1 + k_2x_2)$$

31

# Example: pendulum equation

$$\delta(x) = \left(\frac{\hat{a}c - a\hat{c}}{\hat{c}}\right)\sin(x_1+\delta_1) - \left(\frac{c-\hat{c}}{\hat{c}}\right)(k_1x_1 + k_2x_2)$$

$$|\delta(x)| \le k\|x\| + \varepsilon$$

$$k = \left|\frac{\hat{a}c - a\hat{c}}{\hat{c}}\right| + \left|\frac{c-\hat{c}}{\hat{c}}\right|\sqrt{k_1^2 + k_2^2}, \quad \varepsilon = \left|\frac{\hat{a}c - a\hat{c}}{\hat{c}}\right| |\sin\delta_1|$$

$$P = \begin{bmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{bmatrix}, \quad PB = \begin{bmatrix} p_{12} \\ p_{22} \end{bmatrix}$$

$$k < \frac{1}{2\sqrt{p_{12}^2 + p_{22}^2}}$$

$$\sin\delta_1 = 0 \;\Rightarrow\; \varepsilon = 0$$

32

## Backstepping

$$\begin{aligned}\dot{\eta} &= f(\eta) + g(\eta)\xi \\ \dot{\xi} &= u, \quad \eta \in R^n, \ \xi, \ u \in R\end{aligned}$$

Stabilize the origin using state feedback

View $\xi$ as "virtual" control input to

$$\dot{\eta} = f(\eta) + g(\eta)\xi$$

Suppose there is $\xi = \phi(\eta)$ that stabilizes the origin of

$$\dot{\eta} = f(\eta) + g(\eta)\phi(\eta)$$

$$\frac{\partial V}{\partial \eta}[f(\eta) + g(\eta)\phi(\eta)] \le -W(\eta), \quad \forall \, \eta \in D$$

33

## Backstepping

$$z = \xi - \phi(\eta)$$

$$\begin{aligned}\dot{\eta} &= [f(\eta) + g(\eta)\phi(\eta)] + g(\eta)z \\ \dot{z} &= u - \frac{\partial \phi}{\partial \eta}[f(\eta) + g(\eta)\xi]\end{aligned}$$

$$u = \frac{\partial \phi}{\partial \eta}[f(\eta) + g(\eta)\xi] + v$$

$$\begin{aligned}\dot{\eta} &= [f(\eta) + g(\eta)\phi(\eta)] + g(\eta)z \\ \dot{z} &= v\end{aligned}$$

34

# Backstepping

$$V_c(\eta, \xi) = V(\eta) + \tfrac{1}{2}z^2$$

$$\begin{aligned} \dot{V}_c &= \frac{\partial V}{\partial \eta}[f(\eta) + g(\eta)\phi(\eta)] + \frac{\partial V}{\partial \eta}g(\eta)z + zv \\ &\le -W(\eta) + \frac{\partial V}{\partial \eta}g(\eta)z + zv \end{aligned}$$

$$v = -\frac{\partial V}{\partial \eta}g(\eta) - kz, \quad k > 0$$

$$\dot{V}_c \le -W(\eta) - kz^2$$

35

# Backstepping

$$\begin{aligned} \dot{x} &= f_0(x) + g_0(x)z_1 \\ \dot{z}_1 &= f_1(x, z_1) + g_1(x, z_1)z_2 \\ \dot{z}_2 &= f_2(x, z_1, z_2) + g_2(x, z_1, z_2)z_3 \\ &\vdots \\ \dot{z}_{k-1} &= f_{k-1}(x, z_1, \ldots, z_{k-1}) + g_{k-1}(x, z_1, \ldots, z_{k-1})z_k \\ \dot{z}_k &= f_k(x, z_1, \ldots, z_k) + g_k(x, z_1, \ldots, z_k)u \end{aligned}$$

$$g_i(x, z_1, \ldots, z_i) \neq 0 \quad \text{for } 1 \le i \le k$$

36

# Example

$$\dot{x}_1 = x_1^2 - x_1^3 + x_2, \qquad \dot{x}_2 = u$$

$$\dot{x}_1 = x_1^2 - x_1^3 + x_2$$

$$x_2 = \phi(x_1) = -x_1^2 - x_1 \quad \Rightarrow \quad \dot{x}_1 = -x_1 - x_1^3$$

$$V(x_1) = \tfrac{1}{2}x_1^2 \quad \Rightarrow \quad \dot{V} = -x_1^2 - x_1^4, \quad \forall\, x_1 \in R$$

$$z_2 = x_2 - \phi(x_1) = x_2 + x_1 + x_1^2$$

$$\begin{aligned} \dot{x}_1 &= -x_1 - x_1^3 + z_2 \\ \dot{z}_2 &= u + (1 + 2x_1)(-x_1 - x_1^3 + z_2) \end{aligned}$$

37

# Example

$$V_c(x) = \tfrac{1}{2}x_1^2 + \tfrac{1}{2}z_2^2$$

$$\begin{aligned} \dot{V}_c &= x_1(-x_1 - x_1^3 + z_2) \\ &\quad + z_2[u + (1 + 2x_1)(-x_1 - x_1^3 + z_2)] \end{aligned}$$

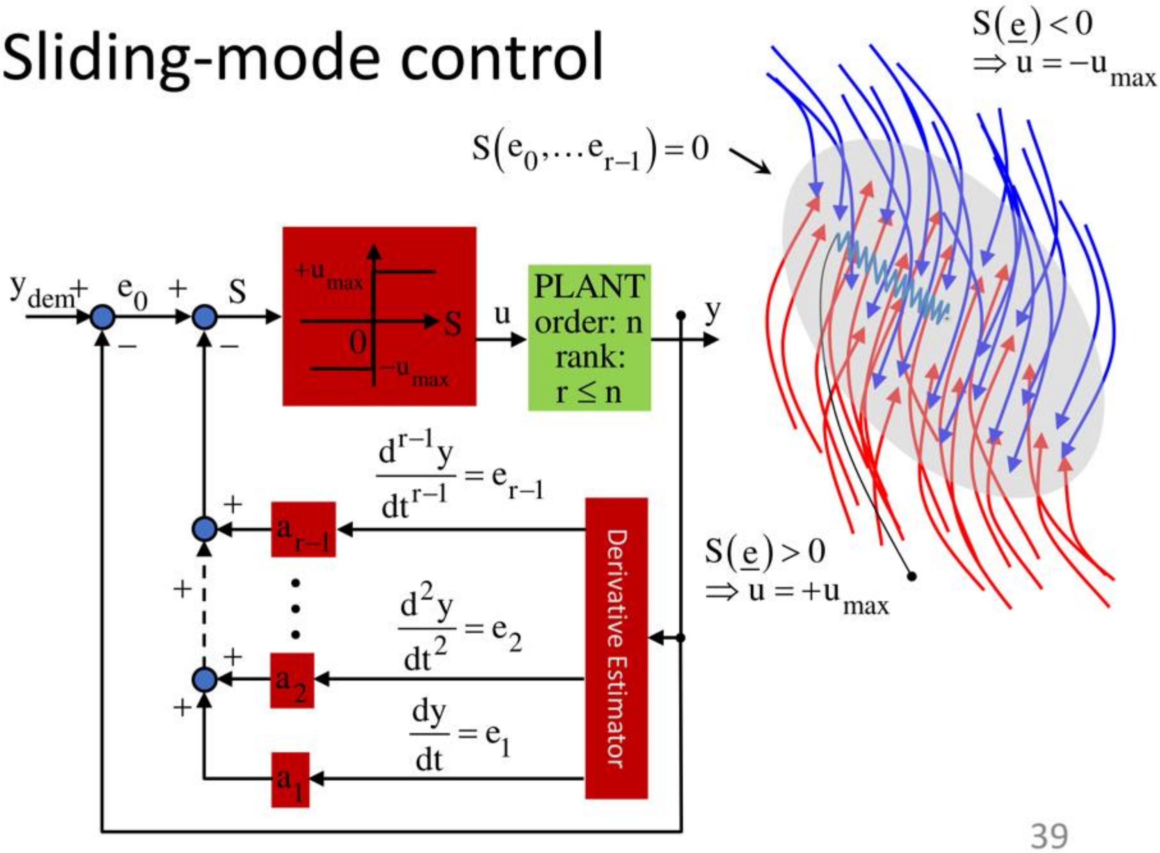
$$\begin{aligned} \dot{V}_c &= -x_1^2 - x_1^4 \\ &\quad + z_2[x_1 + (1 + 2x_1)(-x_1 - x_1^3 + z_2) + u] \end{aligned}$$

$$u = -x_1 - (1 + 2x_1)(-x_1 - x_1^3 + z_2) - z_2$$

$$\dot{V}_c = -x_1^2 - x_1^4 - z_2^2$$

$$\dot{V}_c = -x_1^2 - x_1^4 - (x_2 + x_1 + x_1^2)^2$$

# Sliding-mode control




39

# Sliding-mode control

$$\dot{x}_1 = x_2 \qquad \dot{x}_2 = h(x) + g(x)u, \quad g(x) \geq g_0 > 0$$

Sliding Manifold (Surface):

$$s = a_1 x_1 + x_2 = 0$$

$$s(t) \equiv 0 \quad \Rightarrow \quad \dot{x}_1 = -a_1 x_1$$

$$a_1 > 0 \quad \Rightarrow \quad \lim_{t \to \infty} x_1(t) = 0$$

40

# Sliding-mode control

$$\dot{s} = a_1\dot{x}_1 + \dot{x}_2 = a_1 x_2 + h(x) + g(x)u$$

Suppose

$$\left| \frac{a_1 x_2 + h(x)}{g(x)} \right| \leq \varrho(x)$$

$$V = \tfrac{1}{2}s^2$$

$$\dot{V} = s\dot{s} = s[a_1 x_2 + h(x)] + g(x)su \leq g(x)|s|\varrho(x) + g(x)su$$

$$\beta(x) \geq \varrho(x) + \beta_0, \quad \beta_0 > 0$$

$$s > 0, \quad u = -\beta(x)$$

$$\dot{V} \leq g(x)|s|\varrho(x) - g(x)\beta(x)|s|$$

$$\dot{V} \leq g(x)|s|\varrho(x) - g(x)(\varrho(x) + \beta_0)|s| = -g(x)\beta_0|s|$$

41

# Sliding-mode control

$$s < 0, \quad u = \beta(x)$$

$$\dot{V} \leq g(x)|s|\varrho(x) + g(x)su = g(x)|s|\varrho(x) - g(x)\beta(x)|s|$$

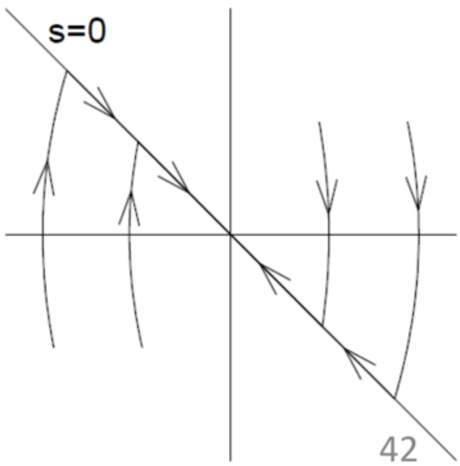
$$\dot{V} \leq g(x)|s|\varrho(x) - g(x)(\varrho(x) + \beta_0)|s| = -g(x)\beta_0|s|$$

$$\text{sgn}(s) = \begin{cases} 1, & s > 0 \\ -1, & s < 0 \end{cases}$$

$$u = -\beta(x)\,\text{sgn}(s)$$

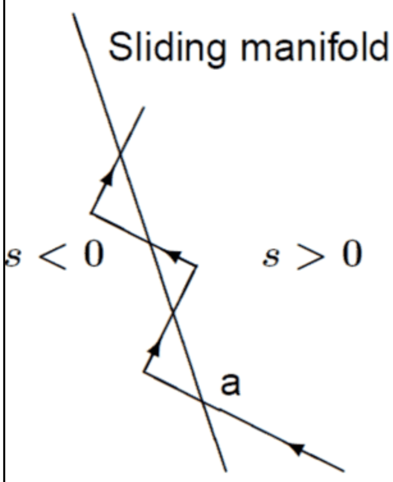
$$\dot{V} \leq -g(x)\beta_0|s| \leq -g_0\beta_0|s|$$

$$\dot{V} \leq -g_0\beta_0\sqrt{2V}$$



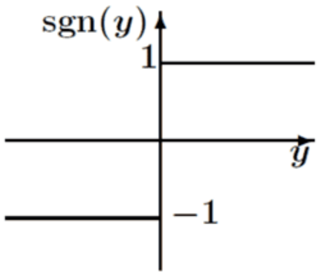

42

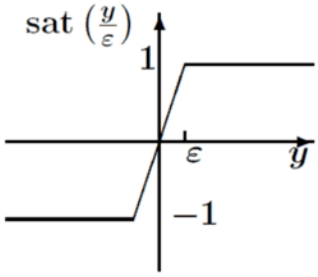
# Sliding-mode control: chattering




$$u = -\beta(x)\,\mathrm{sat}\left(\frac{s}{\varepsilon}\right)$$

$$\mathrm{sat}(y) = \begin{cases} y, & \text{if } |y| \le 1 \\ \mathrm{sgn}(y), & \text{if } |y| > 1 \end{cases}$$




43

Homogeneity of nonlinear systems

# Homogeneity: analysis and control design for dynamical systems

Zimenko Konstantin

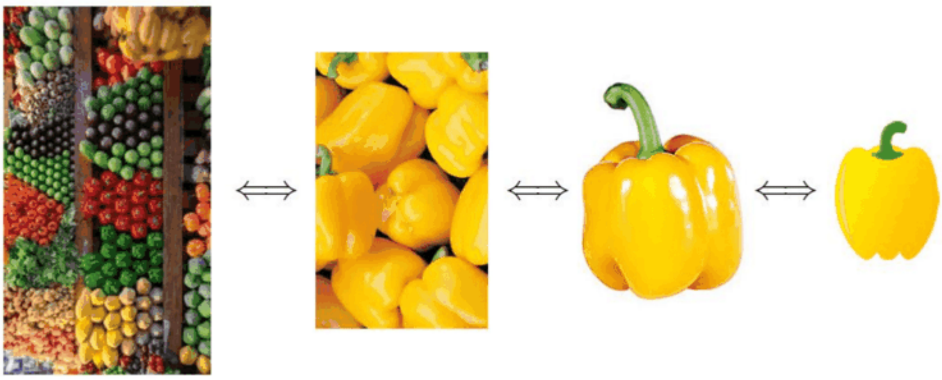
## Homogeneity and heterogeneity

https://en.wikipedia.org/wiki/Homogeneity_and_heterogeneity:
*Homogeneity* and *heterogeneity* are concepts often used in the sciences and statistics relating to the uniformity in a substance or organism. A material or image that is homogeneous is uniform in composition or character; one that is heterogeneous is distinctly nonuniform in one of these qualities.

2

## Beauty of homogeneity

- We are living in a heterogeneous world.
- Homogeneous object/system is an idealization.
- Studying of idealized case in order to deal with the general one.



Homogeneity in mathematics is a kind of symmetry.

3

# History of the subject

The homogeneity is a property of dynamical systems: state rescaling does not change the system behavior (Euler; Zubov, 1958; Rothschild and Stein, 1976; Hermes, 1986).

**Applications:**

- stability analysis (Andrieu, Praly, Astolfi, 2008; Bacciotti and Rosier, 2001; Hermes, 1991a; 1991b; Rosier, 1992);
- systems approximation (Hermes, 1991a);
- stabilization (Bhat and Bernstein, 2005; Grüne, 2000; Kawski, 1991; Moulay and Perruquetti, 2006; Sepulchre and Aeyels, 1996);
- observation (Andrieu, Praly, Astolfi, 2008).

**Extensions:**

- coordinate-free homogeneity (Khomenuk, 1961; Kawski, 1995);
- homogeneity in the bi-limit (Andrieu, Praly, Astolfi, 2008);
- local homogeneity (Efimov and Perruquetti, 2010);
- time-delay systems (Efimov and Perruquetti, 2011; Efimov et al., 2014; 2015), differential inclusions (Filippov, 1988; Bernuau et al., 2013).

4

# Mathematical definition of homogeneity

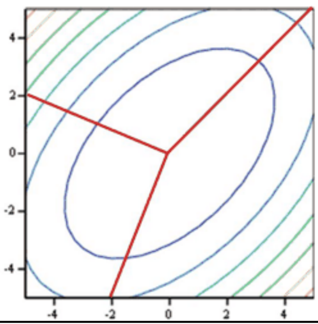
**Definition** For a function $f : \mathbb{R}^n \to \mathbb{R}$, if for any (positive) constant $\lambda$ and all $x \in \mathbb{R}^n$

$$f(\lambda x) = \lambda^\nu f(x),$$

then the function $f$ is called (positively) homogeneous with degree $\nu$.

***Theorem*** *(Euler's theorem on homogeneous functions)*
Let $f : \mathbb{R}^n \to \mathbb{R}$ be a $C^1$ homogeneous function of degree $\nu$, then

$$\frac{df(x)}{dx} x = \nu f(x).$$




5

# Examples of homogeneous functions

- A polynomial function of degree $\nu = 2$:

$$f(x) = x_1^2 + x_1 x_2 + x_2^2, \quad f(\lambda x) = \lambda^2 x_1^2 + \lambda^2 x_1 x_2 + \lambda^2 x_2^2 = \lambda^2 f(x),$$

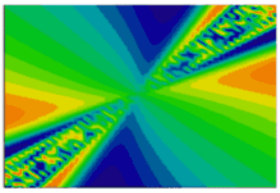
$$\frac{df(x)}{dx} x = (2x_1 + x_2)x_1 + (2x_2 + x_1)x_2 = 2(x_1^2 + x_1 x_2 + x_2^2) = 2f(x).$$

- Functions of degree $\nu = 0$: $f(x) = 1$

$$f(x) = \mathrm{sign}(x_1^2 - x_2^2),\ f(\lambda x) = \mathrm{sign}(\lambda^2 x_1^2 - \lambda^2 x_2^2) = \mathrm{sign}(x_1^2 - x_2^2) = f(x);$$

$$f(x) = \frac{x_1 + x_2}{x_1 - x_2},\ f(\lambda x) = \frac{\lambda x_1 + \lambda x_2}{\lambda x_1 - \lambda x_2} = \frac{x_1 + x_2}{x_1 - x_2} = f(x).$$

- A combination of degree $\nu = 0.5$: $f(x) = \sin\left(\frac{x_1 + x_2}{x_1 - x_2}\right)(x_1^2 + x_1 x_2 + x_2^2)^{\frac{1}{4}}$



6

# Homogeneity for dynamical systems

$$\dot{x} = f(x),\ x \in \mathbb{R}^n,\ f(0) = 0. \tag{1}$$

**Definition** For a function $f : \mathbb{R}^n \to \mathbb{R}^n$, if for any $\lambda > 0$ and all $x \in \mathbb{R}^n$

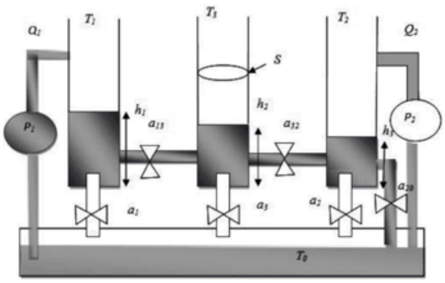
$$f(\lambda x) = \lambda^{\nu+1} f(x),$$

then the function $f$ is called homogeneous with degree $\nu$.

- Linear systems with degree $\nu = 0$:

$$f(x) = Ax, \quad f(\lambda x) = \lambda A x = \lambda f(x).$$

- Nonlinear hydraulic three tank system with $\nu = -0.5$:



$$\begin{aligned} \dot{h}_1 &= -\frac{a_{13}}{S}\lceil h_1 - h_3 \rfloor^{0.5} + \frac{1}{S}Q_1, \quad \lceil s \rfloor^{\alpha} = |s|^{\alpha}\mathrm{sign}(s), \\ \dot{h}_2 &= \frac{a_{32}}{S}\lceil h_3 - h_2 \rfloor^{0.5} - \frac{a_{20}}{S}\lceil h_2 \rfloor^{0.5} + \frac{1}{S}Q_2, \\ \dot{h}_3 &= \frac{a_{13}}{S}\lceil h_1 - h_3 \rfloor^{0.5} - \frac{a_{32}}{S}\lceil h_3 - h_2 \rfloor^{0.5}. \end{aligned}$$

7

# Stability definitions

Denote solution to (1) with initial condition $x_0 \in \mathbb{R}^n$ as $X(t, x_0)$, $0 \in \Omega \subset \mathbb{R}^n$.

**Definition 1** At equilibrium $x = 0$ the system (1) is said to be

(a) *Lyapunov stable* if $\forall x_0 \in \Omega$ the solution $X(t, x_0)$ is defined $\forall t \geq 0$, and $\forall \epsilon > 0\ \exists \delta > 0$ s.t. $\forall x_0 \in \Omega$: $\|x_0\| \leq \delta \Rightarrow \|X(t, x_0)\| \leq \epsilon\ \forall t \geq 0$;

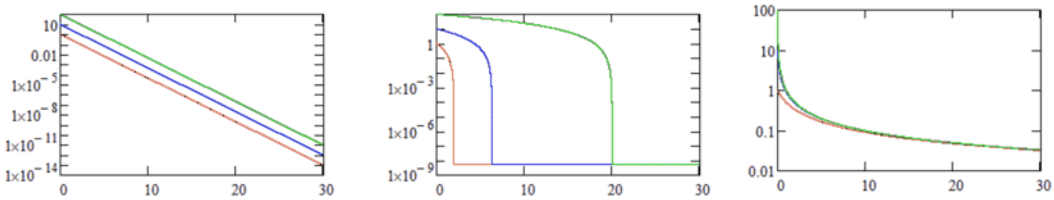
(b) *asymptotically stable* if it is Lyapunov stable and $\forall \kappa > 0$ and $\forall \epsilon > 0$ $\exists T(\kappa, \epsilon) \geq 0$ s.t. $\forall x_0 \in \Omega$: $\|x_0\| \leq \kappa \Rightarrow \|X(t, x_0)\| \leq \epsilon\ \forall t \geq T(\kappa, \epsilon)$;

(c) *finite-time stable* if it is Lyapunov stable and finite-time converging from $\Omega$: $\forall x_0 \in \Omega\ \exists 0 \leq T_0(x_0) < +\infty$ s.t. $X(t, x_0) = 0\ \forall t \geq T_0(x_0)$;

(d) *fixed-time stable* if it is finite-time stable and $\sup_{x_0 \in \Omega} T_0(x_0) < +\infty$.

The set $\Omega$ is called the *domain* of stability/attraction.

If $\Omega = \mathbb{R}^n$, then these properties are called *global*.



8

# Rate of convergence and homogeneous systems

$$\dot{x} = -a\lceil x \rfloor^{\alpha},\ x \in \mathbb{R},\ a > 0,\ \alpha \geq 0.$$

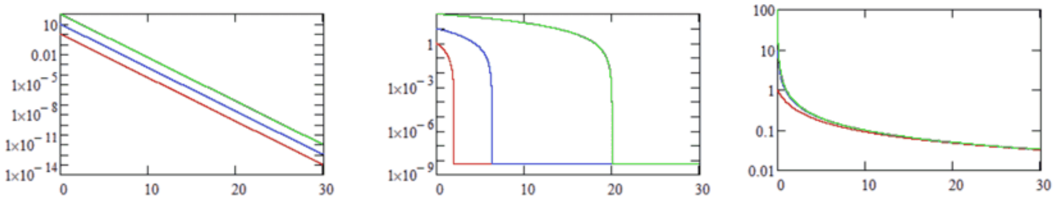
- The system is Lyapunov stable: $V(x) = 0.5x^2$ and $\dot{V} = -a|x|^{\alpha+1} < 0$.
- The system is homogeneous of degree $\nu = \alpha - 1$.

Solutions for $x(0) = x_0$ and $\beta = a(1 - \alpha)$:

$$X(t, x_0) = \begin{cases} \sqrt[-\nu]{|x_0|^{-\nu} - \beta t}\,\mathrm{sign}(x_0) & \alpha < 1 \\ e^{-at} x_0 & \alpha = 1 \\ \frac{x_0}{\sqrt[\nu]{1 - |x_0|^{\nu}\beta t}} & \alpha > 1 \end{cases};$$

- finite-time stability for $\nu < 0$ ($\alpha \in [0, 1)$) with $T_0(x_0) = \frac{|x_0|^{-\nu}}{\beta}$,
- exponential (asymptotic) stability for $\nu = 0$ ($\alpha = 1$),
- fixed-time stable with respect to unit ball for $\nu > 0$ ($\alpha > 1$) :

$$T_1(x_0) = \frac{1 - |x_0|^{-\nu}}{|\beta|},\ \lim_{x_0 \to +\infty} T_1(x_0) = \frac{1}{|\beta|}.$$



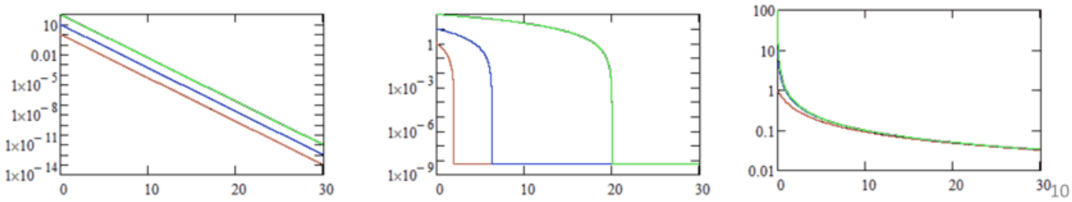
9

# Scaling of trajectories

If $X(t, x_0)$ is a solution of (1) with initial condition $x_0 \in \mathbb{R}^n$, and (1) is homogeneous of degree $\nu$, then $Y(t, y_0) = \lambda X(\lambda^\nu t, \lambda^{-1} y_0)$ for any $\lambda > 0$ is solution of (1) for initial condition $y_0 = \lambda x_0$:

$$\begin{aligned}
\frac{d}{dt} Y(t, y_0) &= \lambda \frac{d}{dt} X(\lambda^\nu t, \lambda^{-1} y_0) \\
&= \lambda^{\nu+1} \frac{d}{d\lambda^\nu t} X(\lambda^\nu t, \lambda^{-1} y_0) \\
&= \lambda^{\nu+1} f(X(\lambda^\nu t, \lambda^{-1} y_0)) = f(\lambda X(\lambda^\nu t, \lambda^{-1} y_0)) \\
&= f(Y(t, y_0)).
\end{aligned}$$

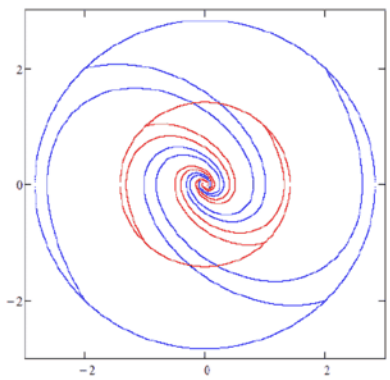
"Homogeneity" of solutions: $X(t, \lambda x_0) = \lambda X(\lambda^\nu t, x_0)$.



# Local ⇒Global

Denote $y \in \mathbb{S} = \{x \in \mathbb{R}^n : \|x\| = 1\}$, then

$$\forall x \in \mathbb{R}^n \ \exists y \in \mathbb{S} : \ x = \lambda y, \ \lambda = \|x\|.$$



- Behavior of all trajectories initiated on a sphere ⇒ Behavior in $\mathbb{R}^n$.
- Local stability ⇒ Global stability.
- Attractivity ⇒ Stability.

11

# Lyapunov functions

Let (1) be *homogeneous* of degree $\nu$, *asymptotically stable* and have a homogeneous Lyapunov function $V$ of degree $\mu$:

$$f(\lambda x) = \lambda^{\nu+1} f(x), \quad V(\lambda x) = \lambda^{\mu} V(x).$$

- $\exists$ a Lyapunov function $\Longleftrightarrow$ $\exists$ a homogeneous Lyapunov function.
- The Lyapunov function is positive definite:

$$c_1 = \inf_{y \in \mathbb{S}} V(y), \; c_2 = \sup_{y \in \mathbb{S}} V(y), \; V(x) = \|x\|^{\mu} V(y),$$

$$c_1 > 0, \; c_2 > 0 \Longrightarrow c_1 \|x\|^{\mu} \le V(x) \le c_2 \|x\|^{\mu}.$$

- Let $\sup_{y \in \mathbb{S}} \frac{\partial}{\partial y} V(y) f(y) = -a, \; a > 0$, then for any $x \in \mathbb{R}^n$ exists $y \in \mathbb{S}$ such that $x = \lambda y$ with $\lambda = \|x\|$:

$$\frac{\partial}{\partial x} V(x) f(x) = \frac{\partial}{\partial \lambda y} V(\lambda y) f(\lambda y) = \lambda^{\nu+\mu} \frac{\partial}{\partial y} V(y) f(y) \le -a \|x\|^{\nu+\mu} \le -\frac{a}{c_2} V^{1+\frac{\nu}{\mu}}(x).$$

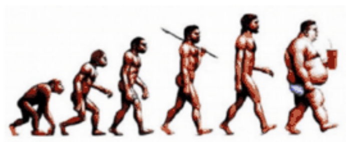
12

# Role of homogeneity

Homogeneity is an algebraic property $\Rightarrow$ It can be easily checked.

Linear systems $\in$ Homogeneous systems $\in$ Nonlinear systems:

| *Linear systems* | *Homogeneous systems* | *Nonlinear systems* |
|---|---|---|
| Scalability of trajectories | Scalability of trajectories | ? |
| Local = Global | Local = Global | Local $\neq$ Global |
| Attractivity $\Rightarrow$ Stability | Attractivity $\Rightarrow$ Stability | Attractivity $\not\Rightarrow$ Stability |
| Quadratic LF | Homogeneous LF | ? |
| Exponential convergence | Degree dependent | ? |
| 0-GAS $\Rightarrow$ ISS | 0-GAS $\Rightarrow$ ISS via degree | 0-GAS $\not\Rightarrow$ ISS |
| Robustness to delay | Robustness to delay | ? |



Conventional (Euler) $\Rightarrow$ Weighted $\Rightarrow$ Local $\Rightarrow$ Geometric/Coordinate-free

13

# Weighted homogeneity

$$\dot{x} = f(x), \ x \in \mathbb{R}^n. \qquad (2)$$

- For any $r_i > 0$, $i = \overline{1, n}$ define vector of weights $r = [r_1 \dots r_n]^T$.
- For any $r$ and $\lambda > 0$ define dilation matrix $\Lambda_r = \text{diag}\{\lambda^{r_i}\}_{i=1}^n$.
- A homogeneous norm can be defined for any $r$:

$$|x|_r = \left( \sum_{i=1}^{n} |x_i|^{\rho / r_i} \right)^{1/\rho}, \rho > 0 \implies |\Lambda_r x|_r = \lambda |x|_r.$$

For $x \in \mathbb{R}^n$, its Euclidean norm $|x|$ is related with $|x|_r$:

$$\underline{\sigma}_r(|x|_r) \le |x| \le \bar{\sigma}_r(|x|_r), \ \underline{\sigma}_r, \bar{\sigma}_r \in K_\infty.$$

- The homogeneous sphere $S_r = \{x \in \mathbb{R}^n : |x|_r = 1\}$.

14

# Weighted homogeneity

**Definition 2**

*Function $g : \mathbb{R}^n \to \mathbb{R}$ is called $r$-homogeneous if*

$$\exists d \ge 0 \ : \ g(\Lambda_r x) = \lambda^d g(x) \quad \forall x \in \mathbb{R}^n \ \forall \lambda > 0.$$

*The function $f : \mathbb{R}^n \to \mathbb{R}^n$ is called $r$-homogeneous if*

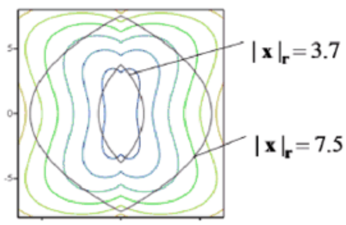
$$\exists d \ge -\min_{1 \le i \le n} r_i \ : \ f(\Lambda_r x) = \lambda^d \Lambda_r f(x) \quad \forall x \in \mathbb{R}^n \ \forall \lambda > 0.$$

*$d$ is called degree of homogeneity.*

**Homogeneous function:**

$$g(x_1, x_2) = \frac{x_1^2 + x_2^4}{|x_1| + |x_2|^2}, \ r = [2\ 1]^T, \ d = 2;$$

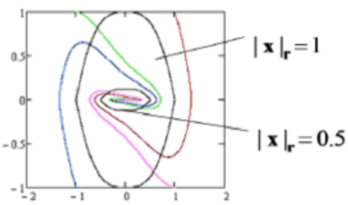
$$g(\Lambda_r x) = g(\lambda x_1, \lambda x_2) = \lambda^2 g(x_1, x_2).$$




**Homogeneous system:**

$$\mathbf{f}(x_1, x_2) = [\sqrt[3]{x_2} \ \ -x_1^3 - x_2]^T, \ r = [1\ 3]^T, \ d = 0.$$

$$\mathbf{f}(\Lambda_\mathbf{r} \mathbf{x}) = \mathbf{f}(\lambda x_1, \lambda^3 x_2) = \Lambda_\mathbf{r} \mathbf{f}(\mathbf{x}).$$




15

# Weighted homogeneity

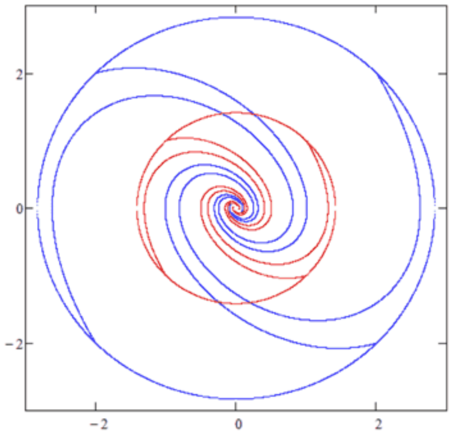
Local attractivity

$\Leftrightarrow$

Global asymptotic stability

$+$

Homogeneous Lyapunov function



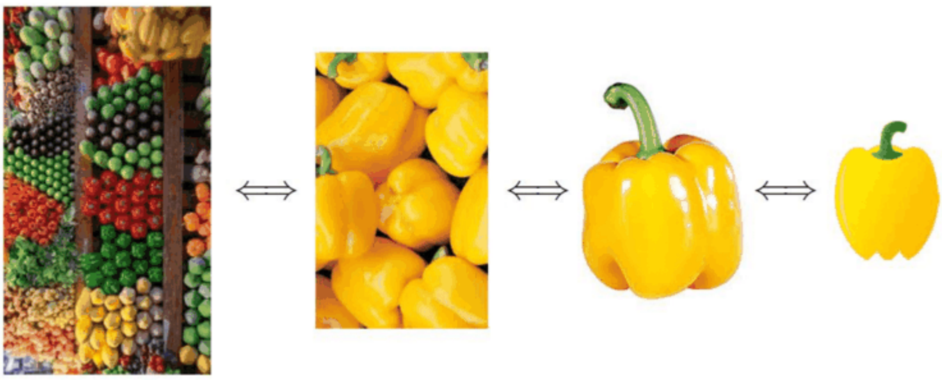

***Proposition 1*** Let (2) be a $r$-homogeneous system with degree $d$ and $x(t)$ be a trajectory with initial condition $x_0$. The curve $t \mapsto \Lambda_r x(\lambda^d t)$ is a trajectory of the system with initial condition $\Lambda_r x_0$ for all $\lambda > 0$:

$$\tfrac{d}{dt}\left(\Lambda_r x(\lambda^d t)\right) \;=\; \lambda^d \Lambda_r f(x(\lambda^d t)) = f(\Lambda_r x(\lambda^d t)).$$

16

# Homogeneity & Heterogeneity

- Homogeneous systems have global behaviors
  - no limit cycles
  - no isolated equilibria
- Homogeneous systems have many useful properties
  - analysis
  - synthesis
- How to apply the theory of homogeneous systems in a heterogeneous world?



17

# Local homogeneity

**Definition 3** *Function $g : \mathbb{R}^n \to \mathbb{R}$ is called $(r,\lambda_0,g_0)$-homogeneous ($g_0 : \mathbb{R}^n \to \mathbb{R}$) if*

$$\exists d_0 \geq 0 \ : \ \lim_{\lambda \to \lambda_0} \lambda^{-d_0} g(\Lambda_r x) = g_0(x) \quad \forall x \in S_r.$$

*Function $f : \mathbb{R}^n \to \mathbb{R}^n$ is called $(r,\lambda_0,f_0)$-homogeneous ($f_0 : \mathbb{R}^n \to \mathbb{R}^n$) if*

$$\exists d_0 \geq -\min_{1 \leq i \leq n} r_i \ : \ \lim_{\lambda \to \lambda_0} \lambda^{-d_0} \Lambda_r^{-1} f(\Lambda_r x) = f_0(x) \quad \forall x \in S_r.$$

- Bi-limit homogeneity in (Andrieu, Praly, Astolfi, 2008) for $\lambda_0 \in \{0, +\infty\}$ (the limit has to be uniform on $S_r$).
- The approximating functions $g_0, f_0$ for $0 < \lambda_0 < +\infty$ can be chosen homogeneous:

$$g_0(x) = |x|_r^d \lambda_0^{-d_0} g(\Lambda_{r,0}\Lambda_{|x|}^{-1}x), \quad f_0(x) = |x|_r^d \lambda_0^{-d_0} \Lambda_{r,0}^{-1} f(\Lambda_{r,0}\Lambda_{|x|}^{-1}x), \qquad (3)$$
$$\Lambda_{r,0} = \mathrm{diag}\{\lambda_0^{r_i}\}_{i=1}^n, \ \Lambda_{|x|} = \mathrm{diag}\{|x|_r^{r_i}\}_{i=1}^n.$$

- Linearization $\neq$ Local homogeneity: $f(x) = -x^3 + x^5 \Rightarrow f_0(x) = -x^3$.

18

# Stability analysis

Relations between $\dot{x} = f(x)$ and $\dot{x} = f_0(x)$ (Zubov, 1958; Rosier, 1992; Andrieu, Praly, Astolfi, 2008):

- $f_0$ is GAS for $\lambda_0 = 0 \Longrightarrow f$ is LAS at the origin;
- $f_0$ is GAS for $\lambda_0 = +\infty \Longrightarrow f$ is Lagrange stable.

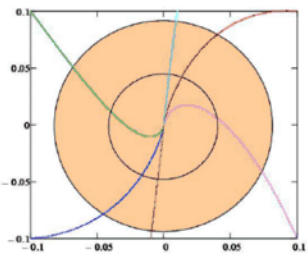
$$\mathbf{f}(x_1,x_2) = \begin{bmatrix} -x_1 + x_1x_2^4 - x_1^5 + 2x_1^2 - 2x_1^2x_2 \\ -x_2 + x_1 - x_2^5 - x_1^2x_2^3 - x_1x_2^2 - x_2^2 \end{bmatrix}$$

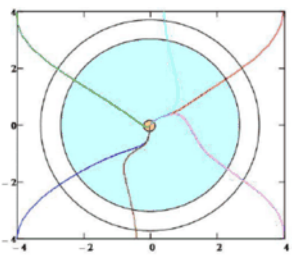
$\lambda_1 = 0, \mathbf{r}_1 = [1\ 1]^T, d_1 = 0$

$$\mathbf{f}_1(x_1,x_2) = \begin{bmatrix} -x_1 \\ -x_2 + x_1 \end{bmatrix}$$

$\lambda_2 = +\infty, \mathbf{r}_2 = [1\ 1]^T, d_2 = 4$

$$\mathbf{f}_2(x_1,x_2) = \begin{bmatrix} x_1x_2^4 - x_1^5 \\ -x_2^5 - x_1^2x_2^3 \end{bmatrix}$$

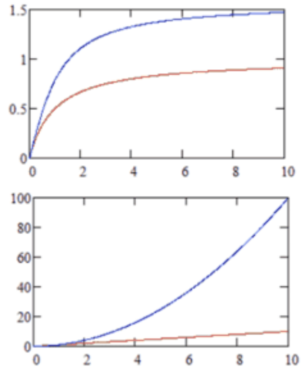



19

# Functions of classes $\mathcal{K}$, $\mathcal{K}_\infty$ and $\mathcal{KL}$

- A $C^0$ function $\alpha : R_+ \to R_+$ belongs to the class $\mathcal{K}$ if $\alpha(0) = 0$ and it is strictly increasing:
  - $\alpha(s) = as^2$ for $a > 0$,
  - $\alpha(s) = \arctan(s)$,
  - $\alpha(s) = \frac{s}{1+s}$.
- The function $\alpha : R_+ \to R_+$ belongs to the class $\mathcal{K}_\infty$ if $\alpha \in \mathcal{K}$ and it is increasing to $\infty$:
  - $\alpha(s) = as^\gamma$ for $a > 0$ and $\gamma > 0$,
  - $\alpha(s) = \ln(1 + s)$.




- A $C^0$ function $\beta : \mathbb{R}_+ \times \mathbb{R}_+ \to \mathbb{R}_+$ is of class $\mathcal{KL}$, if $\beta(\cdot, t) \in \mathcal{K}$ for any $t \in \mathbb{R}_+$, and $\beta(s, \cdot)$ is strictly decreasing to 0 for any fixed $s \in \mathbb{R}_+$:
  - $\beta(s, t) = ae^{-rt}s$ for $r > 0$ and $a > 0$,
  - $\beta(s, t) = \frac{bs}{\sqrt[\nu]{1+as^\nu t}}$ for $b > 0$, $a > 0$ and $\nu > 1$.

20

# ISS property

$$\dot{x}(t) = f(x(t), d(t)), \ t \geq 0, \tag{4}$$

where $x(t) \in \mathbb{R}^n$ is the state, $d(t) \in \mathbb{R}^m$ is the external input, $d \in \mathcal{L}_\infty$.

**Definition 4** The system (6) is called *input-to-state practically stable (ISpS)*, if $\forall d \in \mathcal{L}_\infty$ and $\forall x_0 \in \mathbb{R}^n$ $\exists$ $\beta \in \mathcal{KL}$, $\gamma \in \mathcal{K}$ and $c \geq 0$ such that

$$\|X(t, x_0, d)\| \leq \beta(\|x_0\|, t) + \gamma(||d||_{[0,t)}) + c \quad \forall t \geq 0.$$

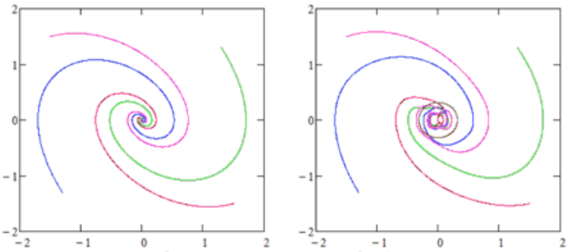
The system is called *ISS* if $c = 0$.

**Definition 5** The system (6) is called *integral ISS* (iISS), if $\forall d \in \mathcal{L}_\infty$ and $\forall x_0 \in \mathbb{R}^n$ $\exists$ $\alpha \in \mathcal{K}_\infty$, $\gamma \in \mathcal{K}$ and $\beta \in \mathcal{KL}$ such that

$$\alpha(\|X(t, x_0, d)\|) \leq \beta(\|x_0\|, t) + \int_0^t \gamma(\|d(s)\|)\, ds \quad \forall t \geq 0.$$

21

# ISS property



$d = 0$ $\quad$ $d \neq 0$

**Definition 6** A $C^\infty$ function $V : \mathbb{R}^n \to \mathbb{R}_+$ is called *ISpS*-Lyapunov function if
(i) $\exists \alpha_1, \alpha_2 \in \mathcal{K}_\infty$ s.t. $\forall x \in \mathbb{R}^n$:

$$\alpha_1(|x|) \leqslant V(x) \leqslant \alpha_2(|x|);$$

(ii) $\exists \sigma, \alpha_3 \in \mathcal{K}_\infty$ and a constant $c \geqslant 0$ s.t. $\forall x \in \mathbb{R}^n$ and $\forall d \in \mathbb{R}^m$:

$$\frac{\partial}{\partial x} V(x) f(x, d) \leqslant -\alpha_3(|x|) + \sigma(|d|) + c.$$

It is called ISS-Lyapunov function if $c = 0$.

It is called iISS-Lyapunov function if $c = 0$ and $\alpha_3$ is a positive definite function.

22

# ISS/iISS for homogeneous systems

Define $\tilde{f}(x, d) = [f(x, d)^T \ 0_m]^T \in \mathbb{R}^{n+m}$, where $0_m$ is a zero vector.

***Theorem 2*** Let $\tilde{f}$ be homogeneous with the weights $\mathbf{r} = [r_1, \ldots, r_n] > 0$, $\tilde{\mathbf{r}} = [\tilde{r}_1, \ldots, \tilde{r}_m] \geq 0$ with a degree $\nu \geq -\min_{1 \leq j \leq n} r_j$, i.e. $f(\Lambda_r x, \Lambda_{\tilde{r}} d) = \lambda^\nu \Lambda_r f(x, d)$.
Let (6) be GAS for $d = 0$, then the system (6) is
ISS *if* $\tilde{r}_{\min} > 0$, *where* $\tilde{r}_{\min} = \min_{1 \leq j \leq m} \tilde{r}_j$;
iISS *if* $\tilde{r}_{\min} = 0$ *and* $\nu \leq 0$.

***Corollary 1*** Let $f_0 : \mathbb{R}^n \to \mathbb{R}^n$ be $r$-homogeneous with a degree $\nu$ and GAS.
If $f(x, d) = f_0(x) + d$, then (6) is ISS for $\nu > -r_{\min}$, and iISS for $\nu = -r_{\min}$.
If $f(x, d) = f_0(x + d)$, then (6) is always ISS.

Example: $\mathbf{r} = [1 \ 3]$, $\tilde{\mathbf{r}} = [2 \ 2]$, $\nu = 2$

$$\begin{aligned} \dot{x}_1 &= -x_1^3 + x_2^{1/3} d_1, \\ \dot{x}_2 &= -x_2^{5/3} + x_1^3 d_2. \end{aligned}$$

23

# ISS/iISS for homogeneous systems

Analysis of ISS/iISS/*ISpS* $\Longleftrightarrow$ Find an ISS/iISS/*ISpS* LF for $\dot{x} = f(x, d)$.

Analysis of ISS/iISS via homogeneity

$\Longleftrightarrow$

Algebraic operations + a LF for $\dot{x} = f(x, 0)$.

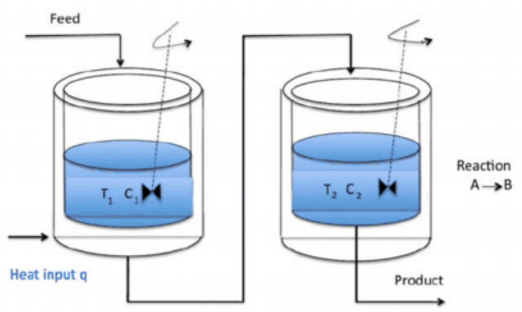
Analysis of *ISpS* via homogeneity

$\Longleftrightarrow$

Algebraic operations + a LF for $\dot{x} = f_{\infty}(x, 0)$.

24

# Homogeneity & delays




**Time delay systems arise in:**

- Chemical processing (transportation delays)
- Remote control (delays from communication links)
- Economics (delayed effects of economic polices)
- ⋮

**BUT delays may induce:** Poor performances; Instability; Difficulties in control design.

25

# Homogeneity & delays

Denote by $C^n[a, b]$, $0 \leq a < b \leq +\infty$ the Banach space of continuous functions $\phi : [a, b] \to \mathbb{R}^n$ with the uniform norm $||\phi|| = \sup_{a \leq \zeta \leq b} |\phi(\zeta)|$.

Autonomous functional differential equation of retarded type:

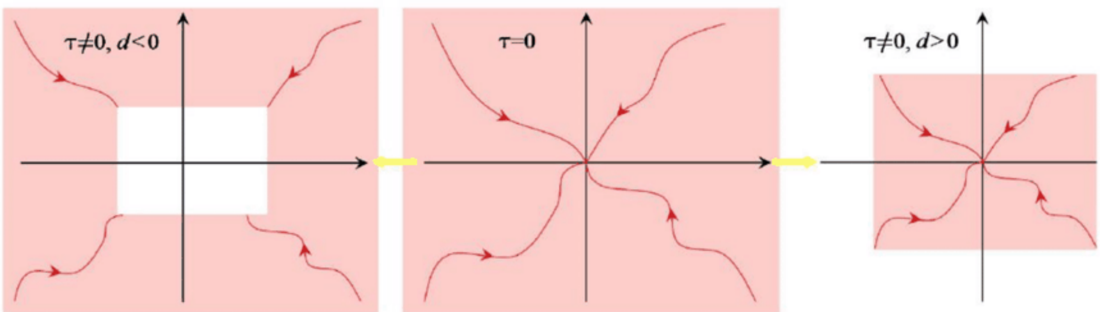
$$dx(t)/dt = f(x_t), \; t \geq 0, \qquad (5)$$

- $x \in \mathbb{R}^n$ and $x_t \in C^n[-\tau, 0]$ is the state function;
- $x_t(s) = x(t + s)$, $-\tau \leq s \leq 0$;
- $f : C^n[-\tau, 0] \to \mathbb{R}^n$ is locally Lipschitz continuous, $f(0) = 0$.

26

# Homogeneity & delays

***Lemma 1*** Let $f(x_t) = F[x(t), x(t-\tau)]$ in (5) be $r$-homogeneous with degree $d > 0$ ($d < 0$) and GAS for $\tau = 0$, then $\forall \rho \; \exists 0 < \tau_0 < +\infty$ such that (2) is LAS in $B_\rho^\tau$ (GAS with respect to $B_\rho^\tau$) $\forall \rho \; \exists 0 < \tau < \tau_0$.




***Theorem 3*** Let the system (1) be $(r, +\infty, f_0)$-homogeneous with $d_0 < 0$, $f_0(x_t) = F_0[x(t), x(t-\tau)]$ and the origin for the approximating system (3) be *GAS* for $\tau = 0$. Then (1) has bounded trajectories *IOD*.

27

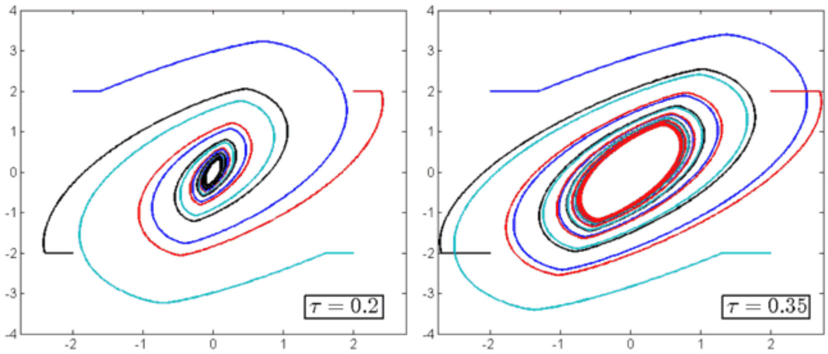
# Homogeneity & delays

Consider the following system

$$\begin{cases} \dot{x}_1 = x_2 - l_1|x_1|^{\alpha}\mathrm{sign}(x_1), \\ \dot{x}_2 = -l_2|x_1|^{2\alpha-1}\mathrm{sign}(x_1), \end{cases}$$

where $l_1 > 0$, $l_2 > 0$, $\alpha \in \left(\frac{1}{2}, 1\right)$. The system is homogeneous for $r = [1, \alpha]^T$ with degree $\mu = \alpha - 1 < 0$.
The state $x_1$ is available with delay $0 \leq \tau \leq \tau_0 < +\infty$.

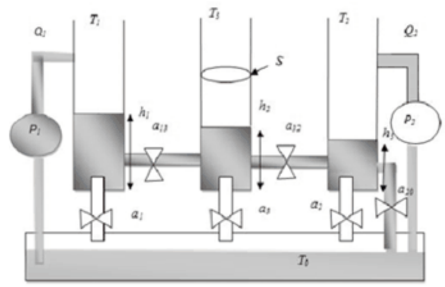
The results of simulation for $\alpha = 0.7$, $l_1 = 1$, $l_2 = 2$:

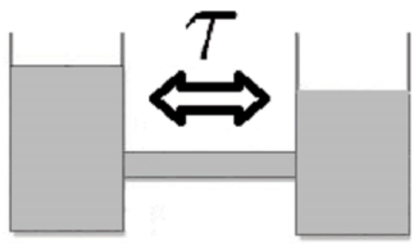



28

# Homogeneity & delays

Liquid flows between the tanks with delays $\tau_i \in (0, \tau_{\max})$, $i = \overline{1, 2}$, $0 < \tau_{\max} \leq \tau_0 < +\infty$:



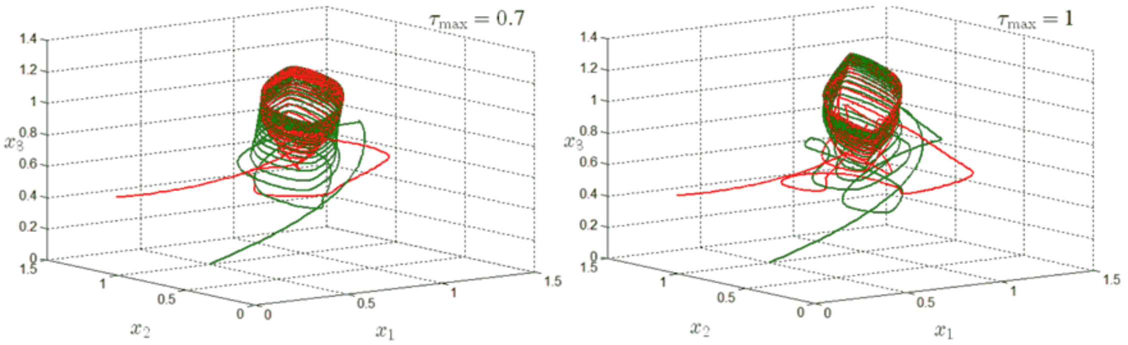



$$\begin{cases} \dot{x}_1(t) = -\frac{a_{13}}{S}\left\lceil x_1(t) - x_3(t-\tau_1)\right\rfloor^{0.5} + \frac{1}{S}u_1(t), \\ \dot{x}_2(t) = \frac{a_{32}}{S}\left\lceil x_3(t-\tau_2) - x_2(t)\right\rfloor^{0.5} - \frac{a_{20}}{S}\left\lceil x_2(t)\right\rfloor^{0.5} + \frac{1}{S}u_2(t), \\ \dot{x}_3(t) = \frac{a_{13}}{S}\left\lceil x_1(t-\tau_1) - x_3(t)\right\rfloor^{0.5} - \frac{a_{32}}{S}\left\lceil x_3(t) - x_2(t-\tau_2)\right\rfloor^{0.5}, \end{cases}$$

For $u_1 = const$, $u_2 = const$ the system is $(r, +\infty, f_0)$-homogeneous with $d_0 = -0.5$. Then by Theorem 1 the system has bounded trajectories *IOD*.

29

# Homogeneity & delays




30

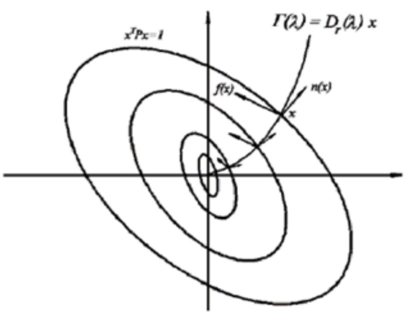
# Control design example

Consider the following **system:**

$$\dot{x}(t) = Ax(t) + bu(t) + d(t,x), \quad (3)$$

where $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}$ is the control input, $d(t,x)\colon \mathbb{R}^{n+1} \to \mathbb{R}^n$ describes the system uncertainties and disturbances,

$$A = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix} \quad \text{and} \quad b = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}$$




Introduce **homogenenous ILF**

$$Q(V,x) := x^T D(V^{-1}) P D(V^{-1}) x - 1, \quad (4)$$

where $P = P^T \in \mathbb{R}^{n \times n} : P > 0$, $D(\lambda)$ is the **dilation matrix** of the form $D(\lambda) = diag\{\lambda^{1+(n-i)\mu}\}_{i=1}^{n}$ for $0 < \mu \leq 1$.

31

# Control design example

***Theorem 4*** The system $\dot{x} = Ax + bu + d(t, x)$ is finite-time robustly stable if:

1) for $\mu \in (0, 1]$, $\alpha, \beta, \gamma, c \in \mathbb{R}_+ : \alpha > \beta$ the following system of inequalities is feasible

$$\begin{cases} AX + XA^T + by + y^T b^T + \alpha X + \beta I_n \leq 0, \\ -\nu X \leq XH_\mu + H_\mu X < 0, \quad X > 0 \end{cases}$$

2) the control has the form

$$u(V, x) = V^{1-\mu} kD(V^{-1})x,$$

where $V \in \mathbb{R}_+ : Q(V, x) = 0$ and $Q(V, x)$ presented by (4) with $P = X^{-1}$;
3) the disturbance function $d(t, x)$ satisfies the following inequality

$$d^T(t, x)D^2(V^{-1})d(t, x) \leq \beta^2 V^{-2\mu}.$$

32

# Control design example

***Corollary 2***

If $\tilde{d} \equiv 0$ then the system (12), (17) is $r$-homogeneous of degree $-\mu$ with $r = (1 + (k-1)\mu, 1 + (k-2)\mu, ..., 1)$. The Implicit Lyapunov Function $V(x)$ is $r$-homogeneous of degree 1.

*Proof*
Obviously, we have $Q(V, D_r(\lambda)s) = Q(\lambda^{-1}V, s)$, i.e. $V(D_r(\lambda)s) = \lambda V(s)$. Now, we derive

$$\begin{aligned} \tilde{u}(D_r(\lambda)s) &= V^{1-\mu}(D_r(\lambda)s)KD_r(V^{-1}(D_r(\lambda)s))D_r(\lambda)s \\ &= \lambda^{1-\mu}V^{1-\mu}(s)KD_r(\lambda^{-1}V^{-1}(s))D_r(\lambda)s = \lambda^{1-\mu}\tilde{u}(x) \end{aligned}$$

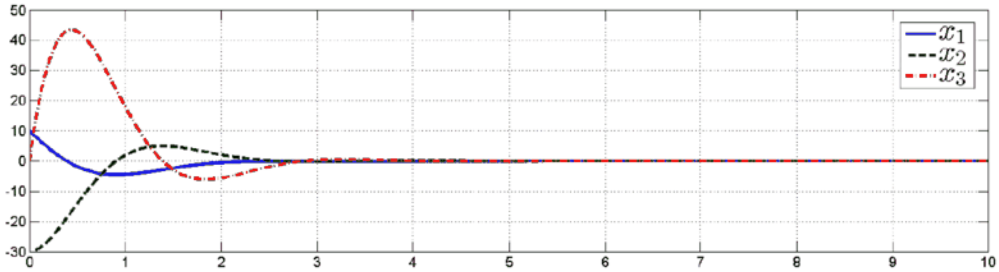
and $\tilde{A}D_r(\lambda)s + \tilde{B}\tilde{u}(D_r(\lambda)s) = \lambda^{-\mu}D_r(\lambda)(\tilde{A}s + \tilde{B}\tilde{u}(s))$. □

**Homogeneity** imply:

- robustness abilities to external perturbations, e.g. **Input-to-State Stability**;
- robustness abilities to **time-delays**;
- it allows to reject some **non-Lipschitz disturbances** in the case of non-zero homogeneity degree;

# Control design example

Finite-time control ($\mu = 1$)
Disturbances: $d_1 = d_2 = 0, \quad d_3 = \text{sign} x_2$.




34

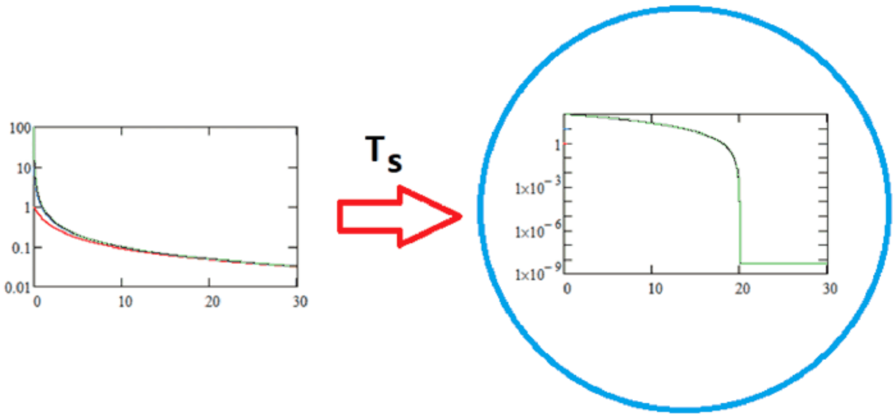
# Control design example

- Fixed-time convergence can be achieved by changing the homogeneity degree in hybrid control algorithm

$$Q_1(V, s) := s^T D_{r_\mu}(V^{-1}) P D_{r_\mu}(V^{-1}) s - 1,$$
$$Q_2(V, s) := s^T D_{r_\nu}(V^{-1}) P D_{r_\nu}(V^{-1}) s - 1,$$

$$D_1(\lambda) = \{\lambda^{1+(n-i)\mu}\}_{i=1}^{n} \text{ and } D_2(\lambda) = \{\lambda^{1+(i-1)\nu}\}_{i=1}^{n}$$

$$u = \begin{cases} V^{1+\nu} k D_2 \left(V^{-1}\right) x & \text{for} \quad t \le T_s \Longrightarrow \textbf{fixed-time attr. of the ball}, \\ V^{1-\mu} k D_1 \left(V^{-1}\right) x & \text{for} \quad t > T_s \Longrightarrow \textbf{finite-time stab. of the origin} \end{cases}$$




35

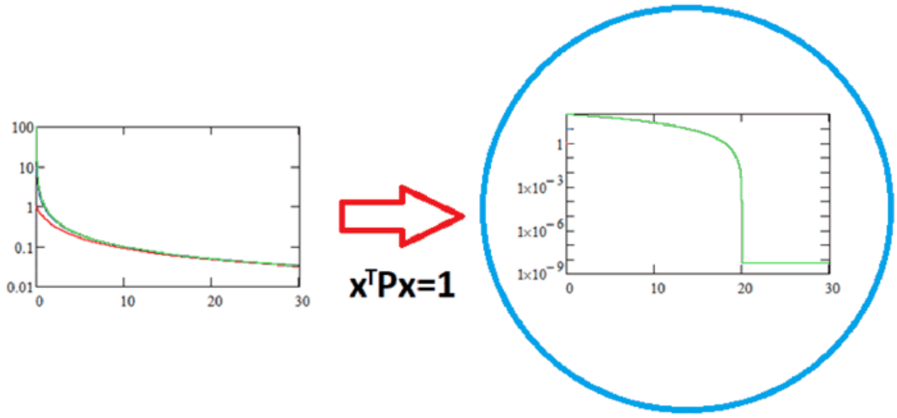
# Control design example

- Fixed-time convergence can be achieved by changing the homogeneity degree in hybrid control algorithm

$$Q_1(V,s) := s^T D_{r_\mu}(V^{-1}) P D_{r_\mu}(V^{-1}) s - 1,$$
$$Q_2(V,s) := s^T D_{r_\nu}(V^{-1}) P D_{r_\nu}(V^{-1}) s - 1,$$

$$D_1(\lambda) = \left\{\lambda^{1+(n-i)\mu}\right\}_{i=1}^{n} \text{ and } D_2(\lambda) = \left\{\lambda^{1+(i-1)\nu}\right\}_{i=1}^{n}$$

$$u = \begin{cases} V^{1+\nu} k D_2\left(V^{-1}\right) x & \text{for} \quad x^T P x \geq 1 \implies \textbf{fixed-time attr. of the ball}, \\ V^{1-\mu} k D_1\left(V^{-1}\right) x & \text{for} \quad x^T P x < 1 \implies \textbf{finite-time stab. of the origin} \end{cases}$$




36

# Conclusions

- Verification of homogeneity: algebraic operations.
- An "*intermediate*" class of systems between linear and non-linear: local ≡ global.
- Local homogeneity: stability/instability in large $\iff$ analysis at the origin of a simplified system.
- Robustness: ISpS, ISS and iISS $\iff$ GAS + degree constraints.
- Robustness to delays.
- Control design with time constraints

That's All

37

## Thank you for your attention to homogeneity:

38

Fault-Tolerant control
Robust Detection of Actuator Faults in Nonlinear Systems

# Robust Detection of Actuator Faults in Nonlinear Systems

Anton Zhilenkov

## Outline

- Introduction
- Fault Diagnosis Methodologies
- Robust Observer-Based Fault Diagnosis: An Overview
- Detection and Isolation of Actuator Faults

2

# Introduction

- A fault can be defined as an unexpected deviation of at least one characteristic property, called the feature of the system, from the normal condition which tends to degrade the overall performance of a system and leads to undesirable but still tolerable behavior of the system.

3

# Common types of faults:

- Actuator faults, such as damage in the bearings, deficiencies in force and momentum, defects in the gears, aging effects, and stuck faults. Actuators are used to generate the desired inputs to control the process to behave normally. When actuator faults occur, the faulty actuators are no longer able to generate the desired control inputs.
- Sensor faults, such as scaling errors, drifts, dead zones, short cuts, and contact failures. Sensors are used to provide measurements that are needed for monitoring the system and computing the desired inputs. When sensor faults occur, the faulty sensors are no longer able to provide accurate measurements which are needed to generate the control inputs.
- Abnormal parameter variations in the system. When some components of the plant are faulty, the original process is changed into a different process so that the controller designed for the original process is no longer able to achieve the expected system performance.
- Construction defects such as cracks, ruptures, fractures, leaks, and loose parts etc.
- External obstacles such as collisions and clogging of outflows.

4

# Common types of faults:

- We will focus on the type of faults which can be modelled as additive changes appearing in actuators or sensors.
- A faulty system with actuator and sensor faults is depicted in fig. 1.1.

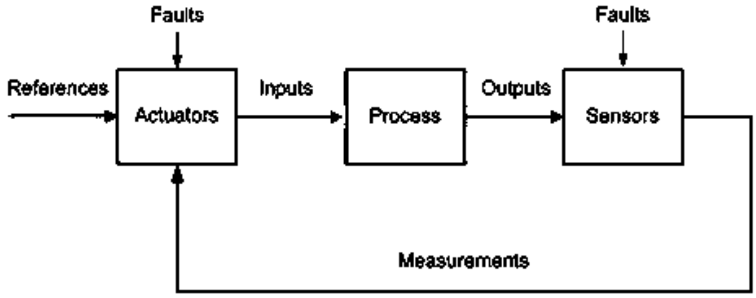



**Fig. 1.1** A faulty system which is subject to actuator faults and sensor faults

5

# Fault Diagnosis Methodologies

- Knowledge-based Fault Detection and Isolation (FDI) methods
- Signal-based FDI methods
- **Model-based FDI methods**

We will focus on model-based fault diagnosis methods. More specifically, observer-based fault diagnosis methods will be the main concern.

6

# FDI methods

Model-based FDI methods comprise two principal steps:

- residual generation,
- residual evaluation.

Corresponding to different residual generation techniques, model-based FDI methods can be divided into three groups:

(1) Parity-equation approach

(2) Parameter-estimation approach

**(3) Observer-based approach**

7

# Robust Observer-Based Fault Diagnosis: An Overview

- A major downside of the model-based fault diagnosis methods is that they require an accurate mathematical model of the considered system.
- The system parameters often vary during the process, that can cause a misleading alarm and therefore make the model-based fault diagnosis system ineffective.
- A robust fault diagnosis system should have the ability to be sensitive to fault signals but insensitive to other signals.

8

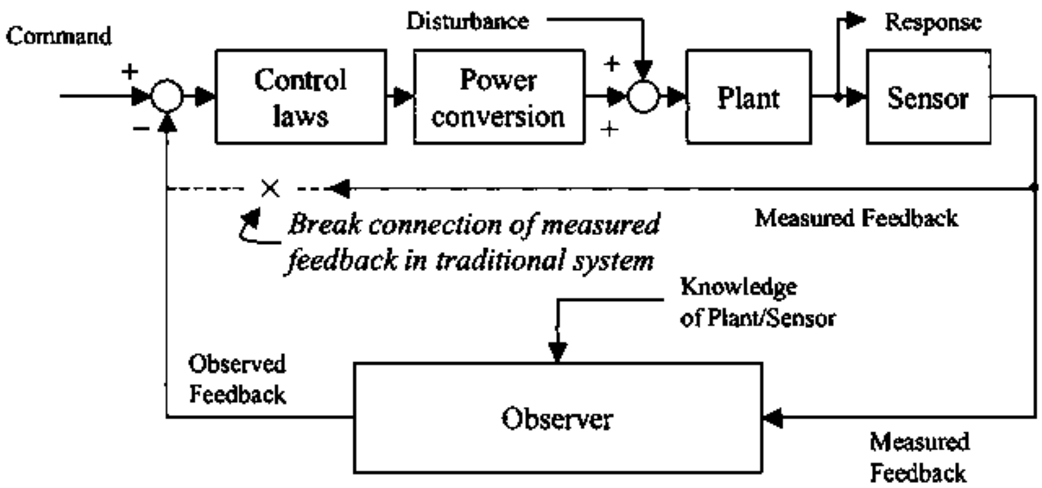
# Robust Observer-Based Fault Diagnosis: An Overview

- Some of robust observer-based fault diagnosis methods:

1. Beard–Jones fault detection filter (BJFDF)-based fault diagnosis.
2. Unknown-input observer (UIO)-based fault diagnosis.
3. Adaptive observer (AO)-based fault diagnosis.
4. **Sliding-mode observer (SMO)-based fault diagnosis.**

9

# Role of an observer in a control system




10

# Detection of Actuator Faults

- Problem formulation
- The general form of the Luenberger observer
- Robust observer-based FD system
- Simulation Results

11

# Problem Formulation

Consider a nonlinear system described by

$$\begin{cases} \dot{x}(t) = Ax(t) + f(x,t) + Bu(t) + Df_a(t) + E\Delta\psi(t), \\ y(t) = Cx(t), \end{cases} \qquad (1.1)$$

where $x \in \mathcal{R}^n$ - vector of state variables;

$u \in \mathcal{R}^m$ - vector of inputs; $y \in \mathcal{R}^p$ - vector of outputs;

$f_a \in \mathcal{R}^h$ - vector of unknown actuator faults;

$\Delta\psi \in \mathcal{R}^r$ - lumped uncertainties and disturbances;

$f(x, t)$ - known nonlinear continuous term.

$A \in \mathcal{R}^{n\times n}$, $B \in \mathcal{R}^{n\times m}$, $C \in \mathcal{R}^{p\times n}$, $D \in \mathcal{R}^{n\times h}$ and $E \in \mathcal{R}^{n\times r}$ are known constant matrices with $C$ and $E$ both being of full rank.

12

# Problem Formulation

*Note that*

A nonlinear system of the form

$$\dot{x}(t) = \Omega(x, u, t)$$

can be expressed as

$$\dot{x}(t) = Ax(t) + f(x, t)$$

if $\Omega(x, u, t)$ is continuously differentiable with respect to $x$.

13

# Problem Formulation

*Remark 1.1*

It is assumed in this example that the fault distribution matrix $D$ is known.

We assume that the actuator faults could occur in each input channel, and therefore we have

$$D = B$$

and

$$f_a \in \mathcal{R}^h.$$

14

# Problem Formulation

- **Assumption 1.1**

$$rank(CE) = rank(E).$$

- **Assumption 1.2** For every complex number $s$ with nonnegative real part

$$rank\begin{bmatrix} sI - A & E \\ C & 0 \end{bmatrix} = n + rank(E)$$

15

# Problem Formulation

- **Assumption 1.3** The nonlinear continuous term $f(x,t)$ is assumed to be known and Lipschitz about the state $x$ uniformly, i.e.,

$$|| f(x, t) - f(\hat{x}, t) || \leq \mathcal{L}_f || x - \hat{x} ||, \quad \forall x, \hat{x} \in \mathcal{R}^n$$

where $\mathcal{L}_f$ is the known Lipschitz constant.

16

# Problem Formulation

- **Assumption 1.4** The actuator fault vector $f_a$ and uncertainty vector $\Delta\psi$ satisfies the following constraint:

$$||f_a|| \le \rho_a \text{ and } ||\Delta\psi|| \le \xi, \qquad (1.4)$$

where $\rho_a$ and $\xi$ are two known positive constants.

17

# Problem Formulation

- **Lemma 1.1** *Under Assumption 1.1, there exist state and output transformations*:

$$z = Tx = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}, \quad w = Sy = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \qquad (1.5)$$

*such that in the new coordinate, the system matrices become,*

$$TAT^{-1} = \begin{bmatrix} A_1 & A_2 \\ A_3 & A_4 \end{bmatrix}, \ TB = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}, \qquad (1.6)$$
$$TE = \begin{bmatrix} E_1 \\ 0 \end{bmatrix}, \ SCT^{-1} = \begin{bmatrix} C_1 & 0 \\ 0 & C_4 \end{bmatrix}$$

where $E_1$ *and* $C_1$ *are invertible.*

18

# Problem Formulation

- *After introducing the state and output transformations (1.5), system (1.1) is expressed as,*

$$\dot{z} = TAT^{-1}z + Tf(T^{-1}z,t) + TB(u+f_a) + TE\Delta\psi$$
$$y = CT^{-1}z \qquad (1.8)$$

*Using the relations in (1.6), system (1.8) is converted into two subsystems as*

$$\begin{cases} \dot{z}_1 = A_1 z_1 + A_2 z_2 + f_1(T^{-1}z,t) + B_1(u+f_a) + E_1\Delta\psi \\ w_1 = C_1 z_1 \end{cases} \qquad (1.9)$$

$$\begin{cases} \dot{z}_2 = A_3 z_1 + A_4 z_2 + f_2(T^{-1}z,t) + B_2(u+f_a) \\ w_2 = C_4 z_2 \end{cases} \qquad (1.10)$$

where $f_1(T^{-1}z, t) = T_1 f(T^{-1}z, t)$ and $f_2(T^{-1}z, t) = T_2 f(T^{-1}z, t)$.

19

# Problem Formulation

**Lemma 1.2** *The pair ($A_4$,$C_4$) is detectable if and only if Assumption 1.2 holds.*

It follows from Lemma 1.2 that there exists a matrix $L \in \mathcal{R}^{(n-r)\times(p-r)}$ such that $A_4 - LC_4$ is stable, and thus for any $Q_2 > 0$, the Lyapunov equation,

$$(A_4 - LC_4)^T P_2 + P_2 (A_4 - LC_4) = -Q_2, \qquad (1.11)$$

has a unique solution $P_2 > 0$.

20

# Problem Formulation

***Remark 1.2*** It is seen from Lemma 1.1 that the satisfaction of Assumption 1.1 ensures the existence of coordinate transformations T and S, such that in the new coordinate, the subsystem-1, formulated in (1.9), is prone to both actuator faults and system uncertainties, while the subsystem-2, formulated in (1.10), is only prone to actuator faults but free from system uncertainties.

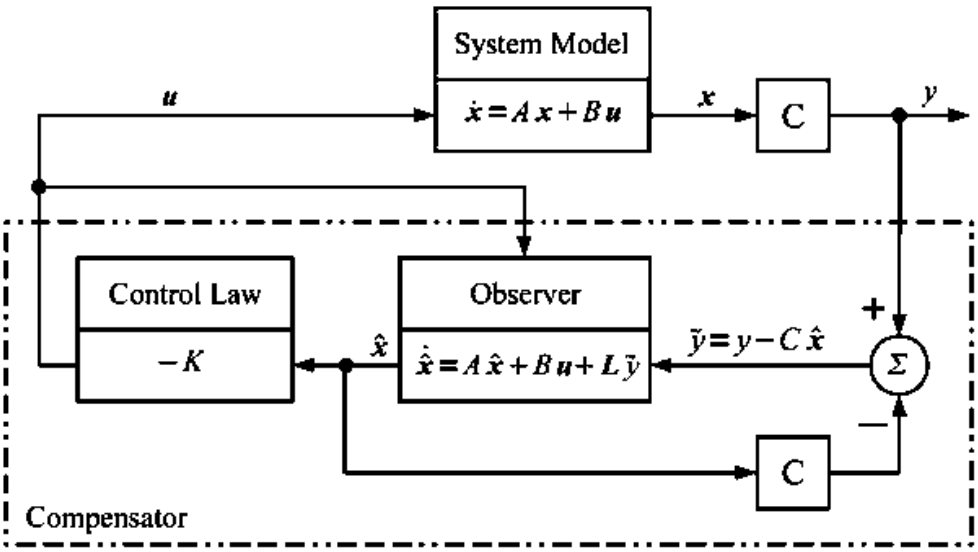
It follows from Assumption 1.2 that the pair $(A_4,C_4)$ is detectable, which provides the necessary condition for the existence of an observer for system (1.10).

Assumption 1.3 states that the nonlinear systems considered is Lipschitz. Many practical systems satisfy the Lipschtiz condition, at least locally. For example, trigonometric nonlinearities occurring in robotic applications and the nonlinearities which are square or cubic in nature, can be assumed to be Lipschitz.

21

# Full-Order State Observer

$$\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + L\big(y(t) - C\hat{x}(t)\big),$$




22

# Full-Order State Observer

Let the observer state estimation error be defined as follows

$$e(t) = x(t) - \hat{x}(t),$$

*then*

$$\dot{e}(t) = \dot{x}(t) - \dot{\hat{x}}(t) =$$

$$= \left[ Ax(t) + Bu(t) + \left( A\hat{x}(t) + Bu(t) + L\left( y(t) - C\hat{x}(t) \right) \right) \right] =$$

$$= \left[ Ax(t) + Bu(t) + \left( A\hat{x}(t) + Bu(t) + L\left( Cx(t) - C\hat{x}(t) \right) \right) \right] =$$

$$= \left( A - LC \right) x(t) - \left( A - LC \right) \hat{x}(t) = \left( A - LC \right) e(t).$$

23

# What Is a Luenberger Observer?

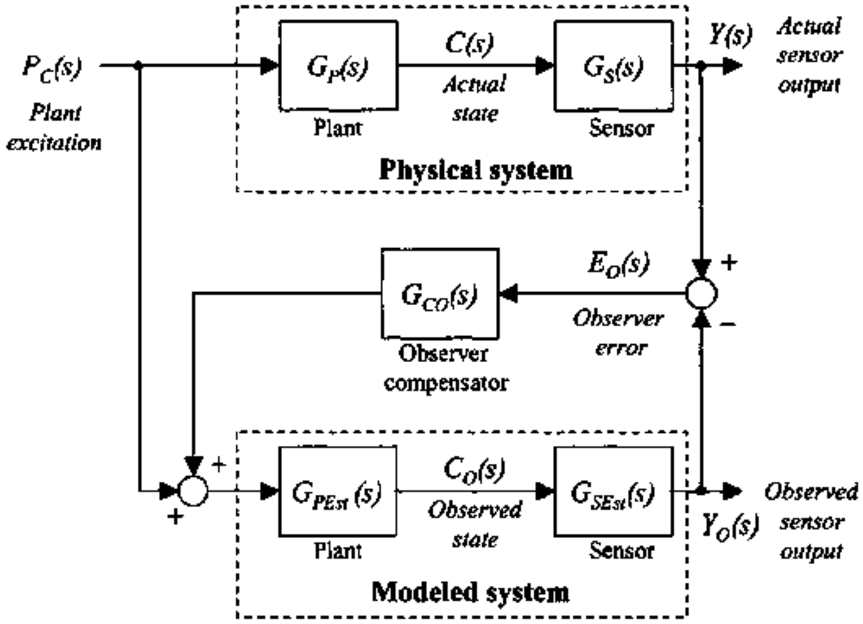
An observer is a mathematical structure that combines sensor output and plant excitation signals with models of the plant and sensor. An observer provides feedback signals that are superior to the sensor output alone.

We'll use the Luenberger observer, which combines five elements:

- a sensor output, $Y(s)$,
- a power converter output (plant excitation), $P_c(s)$,
- a model (estimation) of the plant, $Gp_{Est}(S)$,
- a model of the sensor, $Gs_{Ext}(s)$,
- a PI or PID observer compensator, $G_{co}(S)$.

24

## The general form of the Luenberger observer



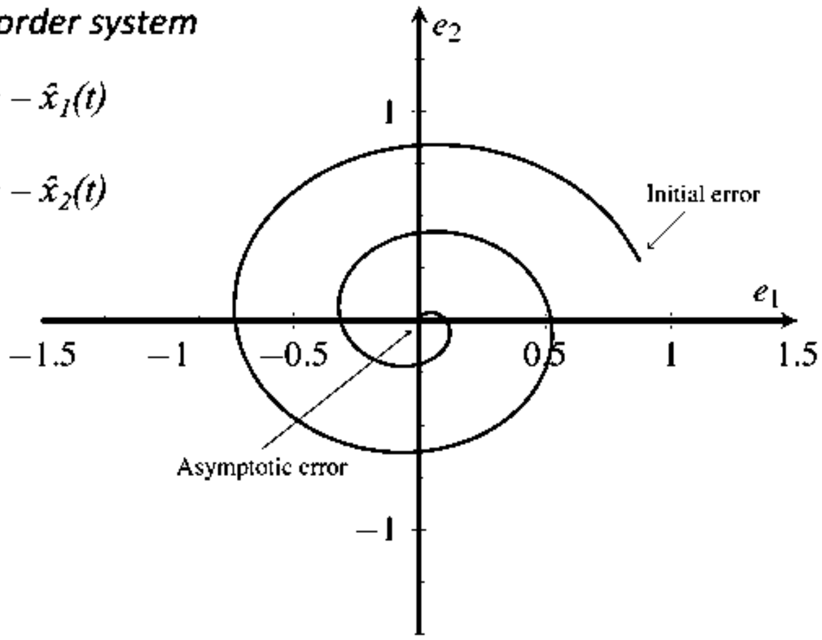

25

## Typical Luenberger Observer Error Trajectory

*for second order system*

$e_1(t) = x_1(t) - \hat{x}_1(t)$

$e_2(t) = x_2(t) - \hat{x}_2(t)$




26

# Luenberger Observer

For Subsystem (1.10), a Luenberger observerwith the following form is designed:

$$\begin{cases} \dot{\hat{z}}_2 = A_4\hat{z}_2 + A_3C_1^{-1}w_1 + f_2(T^{-1}\hat{z},t) + B_2u + L(w_2 - \hat{w}_2) \\ \hat{w}_2 = C_4\hat{z}_2 \end{cases} \quad (1.14)$$

where $L \in \mathcal{R}^{(n-r)\times(p-r)}$ is the gain of a traditional Luenberger observer.

27

# Luenberger Observer

If the state estimation errors are defined as

$$e_1 = z_1 - \hat{z}_1 \text{ and } e_2 = z_2 - \hat{z}_2,$$

then the state estimation error dynamics, before the occurrence of actuator faults, can be obtained as

$$\begin{aligned} \dot{e}_1 &= \dot{z}_1 - \dot{\hat{z}}_1 \\ &= A_1z_1 + A_2z_2 + f_1(T^{-1}z,t) + B_1u + E_1\Delta\psi \\ &\quad - A_1\hat{z}_1 + A_2\hat{z}_2 - f_1(T^{-1}\hat{z},t) - B_1u \\ &\quad - (A_1 - A_1^s)C_1^{-1}(w_1 - \hat{w}_1) - v_1 = \\ &= A_1^se_1 + A_2e_2 + [f_1(T^{-1}z,t) - f_1(T^{-1}\hat{z},t)] + E_1\Delta\psi - v_1 = \\ &= A_1^se_1 + A_2e_2 + \Delta f_1 + E_1\Delta\psi - v_1 \end{aligned} \quad (1.15)$$

28

# Luenberger Observer

$$\begin{aligned}\dot{e}_2 &= \dot{z}_2 - \dot{\hat{z}}_2 = \\ &= A_3 z_1 + A_4 z_2 + f_2(T^{-1}z,t) - A_4\hat{z}_2 - A_3 C_1^{-1} w_1 - f_2(T^{-1}\hat{z},t) \\ &\quad - L(w_2 - \hat{w}_2) = \\ &= (A - LC_4)e_2 + [f_2(T^{-1}z,t) - f_2(T^{-1}\hat{z},t)] = \\ &= (A - LC_4)e_2 + \Delta f_2\end{aligned} \tag{1.16}$$

*where*

$\Delta f_1 = f_1(T^{-1}z, t) - f_1(T^{-1}\hat{z}, t)$ and $\Delta f_2 = f_2(T^{-1}z, t) - f_2(T^{-1}\hat{z}, t)$.

29

***Theorem 1.1*** Given System (1.1) with Assumptions 2.1–2.4. When the system is free of actuator faults, the error dynamics (1.15) and (1.16) are asymptotically stable, if there exist matrices

$$A_1^s < 0, L, P_1 = P_1^T > 0 \text{ and } P_2 = P_2^T > 0$$

and positive scalars $\alpha_1$ and $\alpha_2$ such that

$$\Lambda := \begin{bmatrix} \Pi_1 + \frac{1}{\alpha_1} P_1 P_1 & P_1 A_2 \\ A_2^T P_1 & \Pi_2 + \frac{1}{\alpha_2} P_2 P_2 + a I_{n-r} \end{bmatrix} < 0 \tag{1.17}$$

where $\Pi_1 = A_1^{sT} P_1 + P_1 A_1^s$,

$\Pi_2 = (A_4 - LC_4)^T P_2 + P_2 (A_4 - LC_4)$,

$\alpha = \alpha_1 \mathcal{L}_{f1} ||T^{-1}||^2 + \alpha_2 \mathcal{L}_{f1}^2 ||T^{-1}||^2$.

30

- Let the actuator fault occurs at time instant $t_f$. Then the error dynamics (2.15) and (2.16) become

$$\dot{e}_1 = A_1^s e_1 + A_2 e_2 + \left(f_1(T^{-1}z,t) - f_1(T^{-1}\hat{z},t)\right) + E_1 \Delta\psi + B_1 f_a - \nu_1 \tag{1.37}$$

$$\dot{e}_2 = (A_4 - LC_4)e_2 + \left(f_2(T^{-1}z,t) - f_2(T^{-1}\hat{z},t)\right) + B_2 f_a \tag{1.38}$$

31

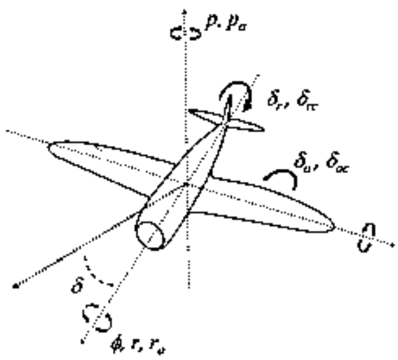
- ***Actuator FD scheme:*** *Actuator faults can be detected if the residual* $||e_{w2}||$ *exceeds a predefined threshold* $\varsigma$. *Otherwise the system is healthy within the considered time. The detection time* $t_d (t_d \geq t_f)$ *is defined as the first time instant such that* $||e_{w2}||$ *is observed greater than* $\varsigma$.

- *Remark 1.4* It follows from Lemma 1.3 that $e_2$ will approach to zero when System (2.1) is healthy. This implies that a small threshold $\varsigma$ can be selected. The value of $\varsigma$ does not significantly affect the performance of the proposed FD scheme.

32

# Simulation Results

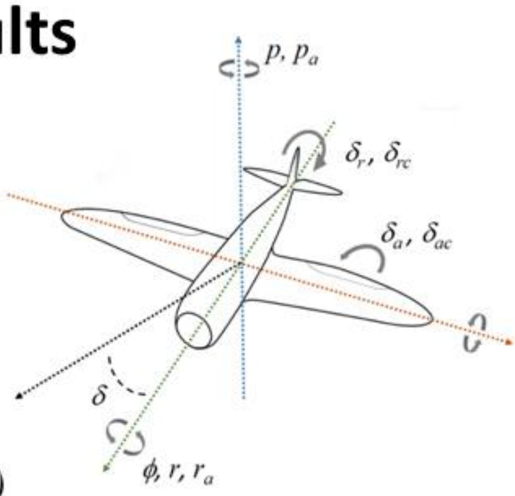
In this section, the effectiveness of the proposed schemes in detecting and isolating actuator faults has been demonstrated by an example of a modified seventh-order aircraft model.




33

# Simulation Results

*aircraft model*




The states are defined as

$x_1 = \varphi -$ *bank angle(rad)*

$x_2 = r -$ *yaw rate(rad/s)*

$x_3 = p -$ *roll rate(rad/s)*

$x_4 = \delta -$ *sideslip angle(rad)*

$x_5 = x_7 -$ *washout filter state*

$x_6 = \delta_r -$ *rudder deflection(rad)*

$x_7 = \delta_a -$ *aileon deflection(rad)*

34

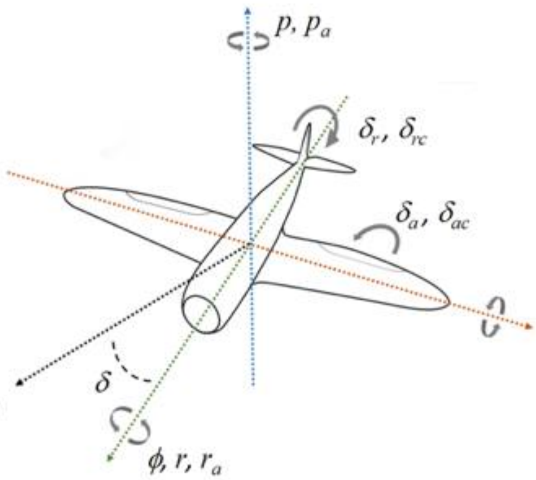
# Simulation Results

The inputs are

$u_1 = \delta_{rc}$ – *rudder command(rad)*
$u_2 = \delta_{ac}$ – *aileon command(rad)*

and outputs are

$y_1 = r_a$ – *roll acceleration(rad/s)*
$y_2 = p_a$ – *yaw acceleration(rad/s)*
$y_3 = \varphi$ – *bank angle(rad)*
$y_4 = x_7$ – *washout f ilter state*




35

$$\begin{cases} \dot{x}(t) = Ax(t) + f(x,t) + Bu(t) + Df_a(t) + E\Delta\psi(t), \\ y(t) = Cx(t), \end{cases} \qquad (1.1)$$

The system is in the form of (1.1) with

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -0.154 & -0.04 & 1.54 & 0 & -0.744 & -0.032 \\ 0 & 0.249 & -1 & -5.2 & 0 & 0.337 & -1.12 \\ 0.0386 & -0.996 & 0 & -2.117 & 0 & 0.02 & 0 \\ 0 & 0.5 & 0 & 0 & -4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -20.000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -25 \end{bmatrix} \qquad B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 20 & 0 \\ 0 & 25 \end{bmatrix}$$

$$C = \begin{bmatrix} 0 & -0.154 & -0.04 & 1.54 & 0 & -0.744 & -0.032 \\ 0 & 0.249 & -1 & -5.2 & 0 & 0.337 & -1.12 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \qquad E = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}^T$$

$$f(x,t) = \begin{bmatrix} \sin x_3 & \sin x_3 & 0 & 0 & \sin x_3 & 0 & 0 \end{bmatrix}^T \qquad \Delta\psi = 2\sin t$$

36

The actuator fault $f_a = col\,(f_{a1}, f_{a2})$ is applied to the system and defined as

$$f_{a_1} = \begin{cases} 0 & , \quad t \le 10s \\ 0.05\exp(0.01t) & , \quad t \ge 10s \end{cases}$$

$$f_{a_2} = \begin{cases} 0 & , \quad t \le 20s \\ 0.07\exp(0.03t) & , \quad t \ge 20s \end{cases}$$

The nonsingular transformation matrices $T$ and $S$ are selected as

$$T = \begin{bmatrix} 0.844 & 0.156 & 0.0405 & -1.5598 & 0 & 0.7535 & 0.0324 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad S = \begin{bmatrix} 1 & 0 & -0.8333 & 0 \\ -1.4359 & 1 & -0.4701 & 0 \\ 1.0128 & 0 & 0.1560 & 0 \\ 1.0128 & 0 & -0.8440 & 1 \end{bmatrix}$$

37

The system matrices under the new coordinate become

$$TAT^{-1} = \left[\begin{array}{c|cccccc} 1.4794 & 1.3088 & 0.7373 & 5.6393 & 0 & -16.3183 & -0.9083 \\ \hline -0.154 & -0.13 & -1.0338 & 1.2998 & 0 & -0.6280 & -0.027 \\ 0.249 & 0.2102 & -1.0101 & -4.8116 & 0 & 0.1494 & -1.1281 \\ -0.9574 & -0.8466 & 0.0388 & -3.6104 & 0 & 0.7414 & 0.0310 \\ -3.5 & 1.04 & -0.8583 & -5.4593 & -4 & 2.6372 & 0.1134 \\ 0 & 0 & 0 & 0 & 0 & -20 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -25 \end{array}\right]$$

$$SCT^{-1} = \left[\begin{array}{c|cccccc} -0.9873 & 0 & 0 & 0 & 0 & -0.0001 & 0 \\ \hline 0 & 0.4701 & -0.9426 & -7.4112 & 0 & 1.4053 & -1.0741 \\ 0 & -0.156 & -0.0405 & 1.5598 & 0 & -0.7535 & -0.0324 \\ 0 & -0.156 & -0.0405 & 1.5598 & 1 & -0.7535 & -0.0324 \end{array}\right]$$

$$TB = \left[\begin{array}{cc} 15.07 & 0.81 \\ \hline 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 20 & 0 \\ 0 & 25 \end{array}\right] \qquad TE = \left[\begin{array}{c} 1 \\ \hline 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{array}\right]$$

38

Imposing the stability constraint to the transformed system and formulating the problem in an LMI framework gives the values of the parameters of the proposed observers.

Parameters are obtained as

$P_1 = 0.0048,\ A_1^s = -25.8618,\ \alpha_1 = 0.0021,\ \gamma = 2.0190\times 10^{-11}$

$$P_0 = \begin{bmatrix} 0.3072 & 0.0109 & -0.0689 & -0.2578 & 0.0144 & 0.0058 \\ 0.0109 & 0.2432 & 0.0527 & 0.0969 & 0.0506 & 0.0342 \\ -0.0689 & 0.0527 & 0.4599 & 0.1648 & -0.0253 & 0.0254 \\ -0.2578 & 0.0969 & 0.1648 & 0.4662 & -0.0008 & 0.0002 \\ 0.0144 & 0.0506 & -0.0253 & -0.0008 & 0.1079 & 0.0545 \\ 0.0058 & 0.0342 & 0.0254 & 0.0002 & 0.0545 & 0.0039 \end{bmatrix}$$

39

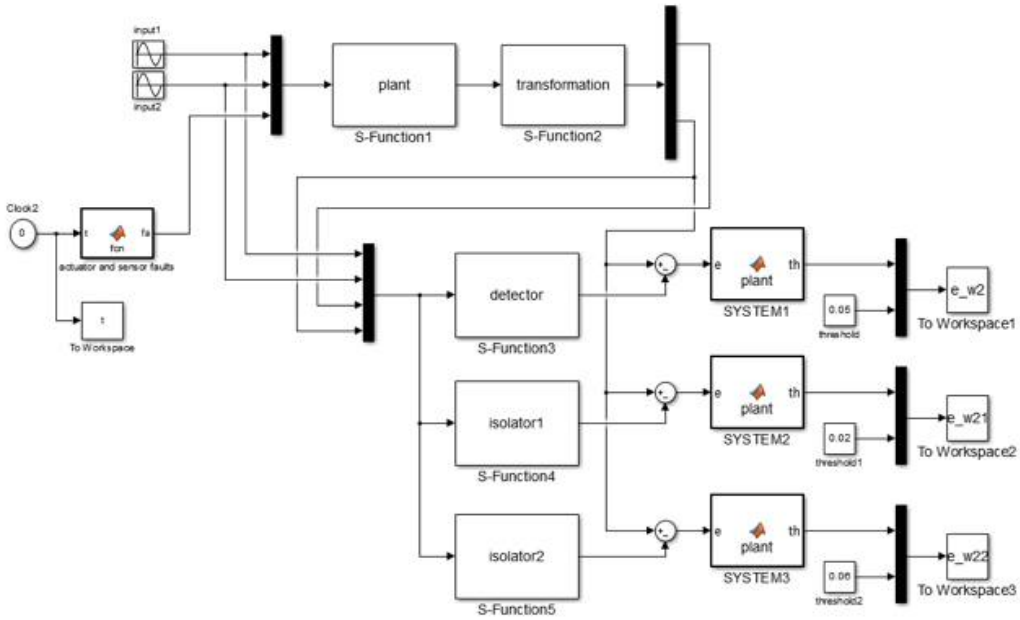
$$L = \begin{bmatrix} 2.3497 & 6.3662 & 1.6347 \\ -3.1985 & -13.8943 & -1.7068 \\ -3.8256 & -11.5108 & -1.5464 \\ 3.6515 & 10.8768 & -1.9943 \\ 58.2398 & 282.3094 & -3.8605 \\ -57.1666 & -394.4680 & 8.0411 \end{bmatrix}$$

$$F = \begin{bmatrix} -0.8797 & -4.4882 & -0.0156 \\ -0.7677 & -3.2451 & 0.0039 \end{bmatrix}$$

It is worth noting that the parameters obtained from LMI may differ from that shown here. This is expected because these are obtained by solving LMIs which does not give unique solutions.

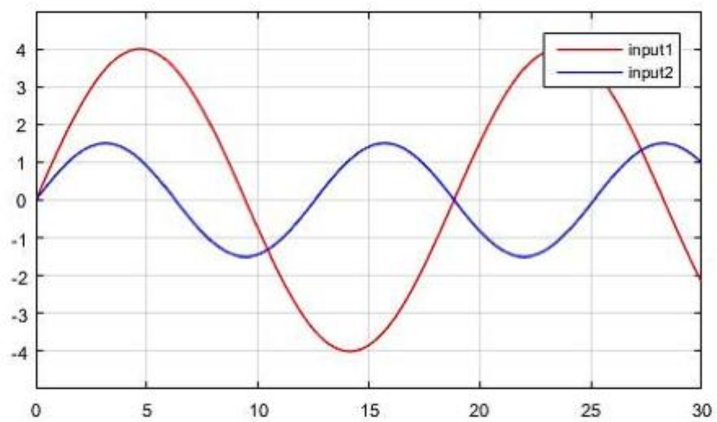
40

## FD MATLAB model



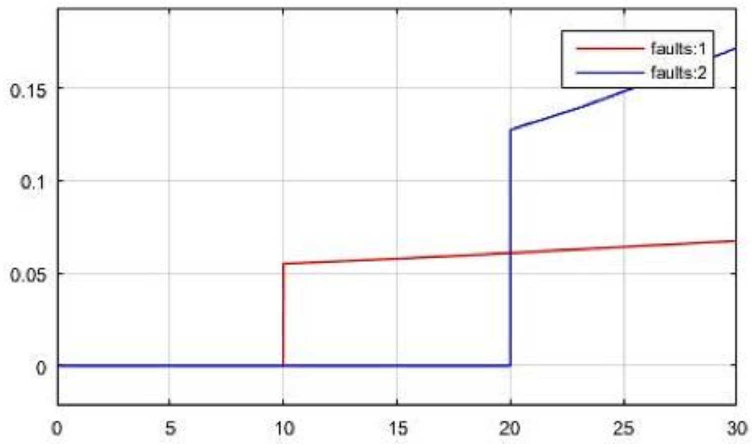

41

## Input signals *u*



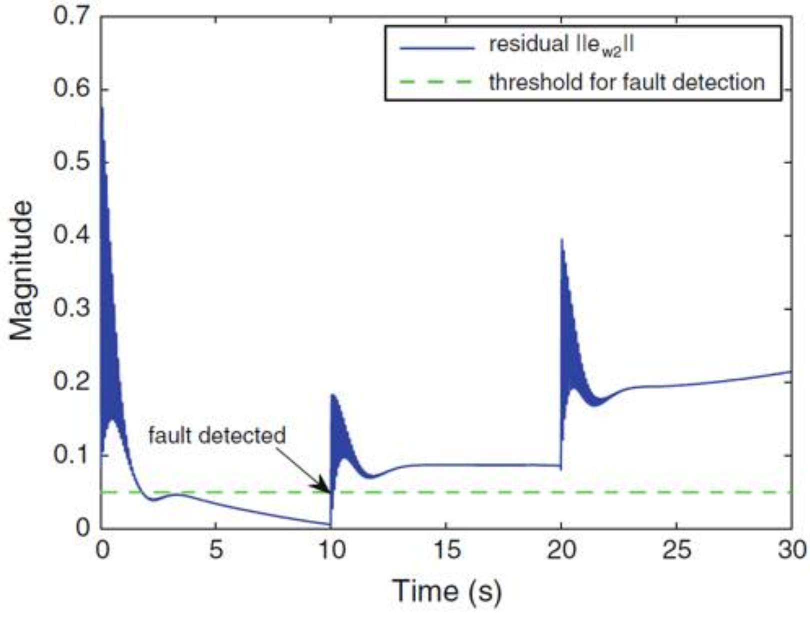

42

## Fault signals $f_a$



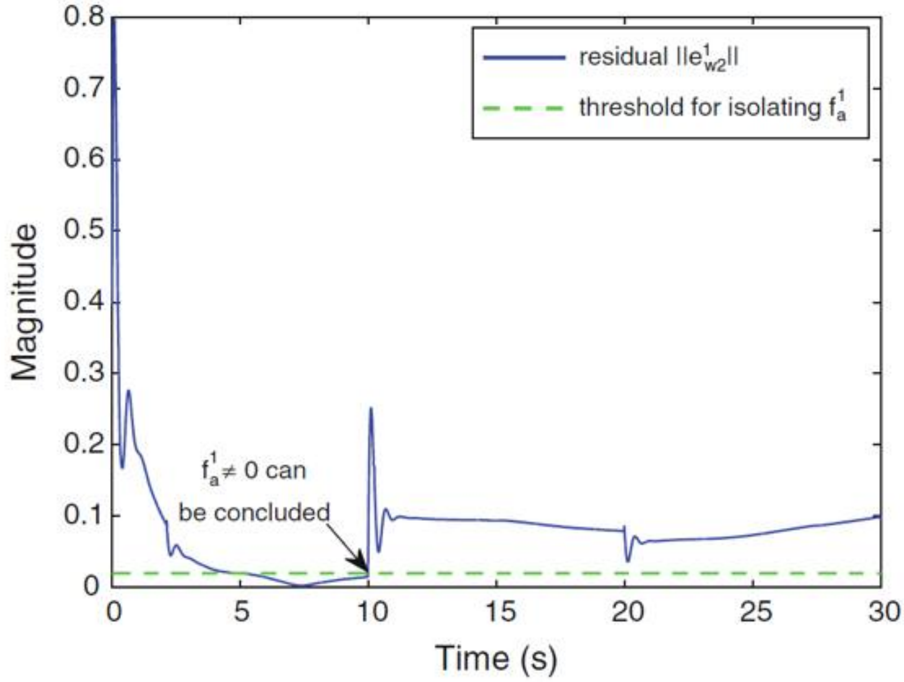

43

## Detection of the occurrence of actuator faults



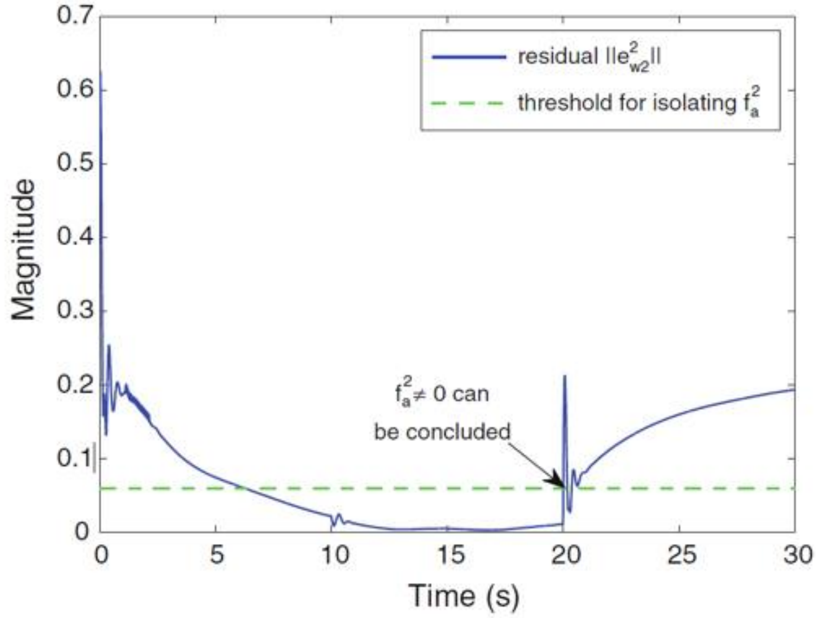

44

Isolation of $f^1_a$




5

Isolation of $f^2_a$




46

## Conclusions

We propose a scheme to robustly detect and isolate incipient actuator faults for uncertain Lipschitz nonlinear systems.

The proposed FDI scheme essentially transforms the original system into two subsystems where subsystem-1 includes both actuator faults and system uncertainties while subsystem-2 has actuator faults but without uncertainties.

Actuator faults can be detected by applying a Luenberger observer for subsystem-2, and isolated using a bank of SMOs for both subsystems based on the modified dedicated observer scheme.

47

## Conclusions

The most distinct feature of the proposed FDI scheme is that, by imposing a coordinate transformation to the original system, the effects of system uncertainties to the residual of subsystem-2 are completely decoupled, which makes the scheme sensitive to incipient faults while still robust to modelling uncertainty.

Thus, early detection can be achieved and a false alarm caused by modeling uncertainties can be totally avoided.

The sufficient conditions of stability of the proposed observers have been studied and represented in the form of LMI.

Its effectiveness has been demonstrated considering the example of a modified aircraft model.

48

Robust Detection of Sensor Faults in Nonlinear Systems

# Robust Detection of Sensor Faults in Nonlinear Systems

Anton Zhilenkov

## Outline

- Introduction
- Problem Formulation
- Sensor Fault Detection Scheme
- Modeling Results with an example of a single-link robotic arm with a revolute elastic joint

2

## Introduction

- With the development of modern technology, autonomous systems are more and more dependent on sensors which often carry the most important information in automated/feedback control systems.
- Faults occurring in sensors may lead to poor regulation or tracking performance, or even affect the stability of the control system.

3

## Introduction

- Therefore, the study of sensor FDI is becoming increasingly important. Compared with the study of actuator FD, the research on sensor FD is less studied in this realm.
- We will extend the method proposed for actuator FD to sensor FD.

4

- We will focus on the type of faults which can be modelled as additive changes appearing in sensors.
- A faulty system with actuator and sensor faults is depicted in fig. 2.1.

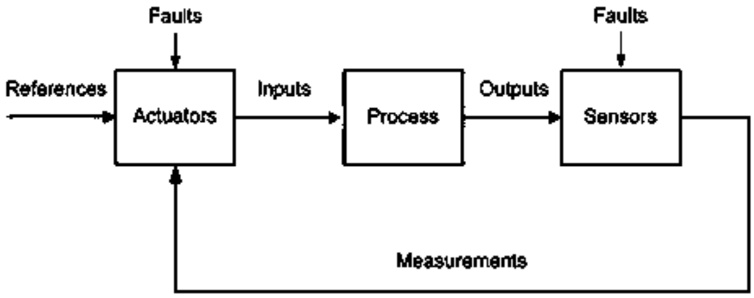
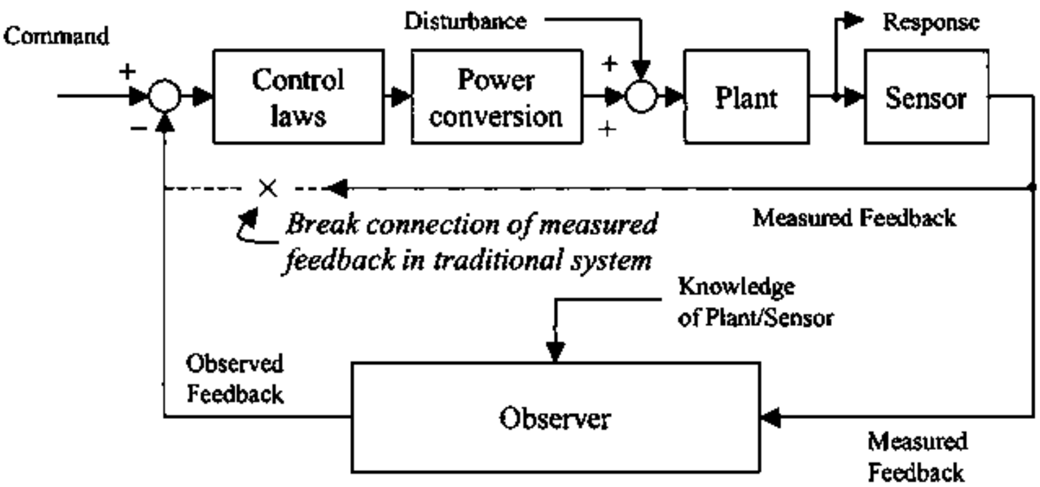



**Fig. 2.1** A faulty system which is subject to actuator faults and sensor faults

5

# Role of an observer in a control system




6

## Problem Formulation

- It is assumed only sensor faults occur in the system. In this case, the considered system has the following form:

$$\begin{cases} \dot{x}(t) = Ax(t) + f(x,t) + Bu(t) + E\Delta\psi(t) \\ y(t) = Cx(t) + Df_s(\tau), \end{cases} \qquad (2.1)$$

where $x \in \mathcal{R}^n$ - vector of state variables;

$u \in \mathcal{R}^m$ - vector of inputs; $y \in \mathcal{R}^p$ - vector of outputs;

$f_s \in \mathcal{R}^q$ - vector of unknown sensor faults;

$\Delta\psi \in \mathcal{R}^r$ - system uncertainties;

$f(x, t)$ - known nonlinear continuous term.

$A \in \mathcal{R}^{n\times n}$, $B \in \mathcal{R}^{n\times m}$, $C \in \mathcal{R}^{p\times n}$, $D \in \mathcal{R}^{n\times h}$ and $E \in \mathcal{R}^{n\times r}$ are known constant matrices with *C, D* and *E* being of full rank, (p≥q+r).

7

## Problem Formulation

- **Assumption 2.1**

$$rank(CE) = rank(E).$$

- **Assumption 2.2** For every complex number $s$ with nonnegative real part

$$rank\begin{bmatrix} sI - A & E \\ C & 0 \end{bmatrix} = n + rank(E)$$

8

## Problem Formulation

- **Assumption 2.3** The nonlinear continuous term *f(x,t)* is assumed to be known and Lipschitz about the state $x$ uniformly, i.e.,

$$\| f(x, t) - f(\hat{x}, t) \| \leq \mathcal{L}_f \| x - \hat{x} \|, \quad \forall x, \hat{x} \in \mathcal{R}^n$$

where $\mathcal{L}_f$ is the known Lipschitz constant.

9

## Problem Formulation

- **Assumption 2.4** The actuator fault vector $f_s$ and uncertainty vector $\Delta\psi$ satisfies the following constraint:

$$\| f_s \| \leq \rho_s \text{ and } \| \Delta\psi \| \leq \xi, \tag{2.4}$$

where $\rho_s$ and $\xi$ are two known positive constants.

10

## Problem Formulation

- **Lemma 2.1** *Under Assumption 2.1, there exist state and output transformations*:

$$z = Tx = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}, \quad w = Sy = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \qquad (2.5)$$

*such that in the new coordinate, the system matrices become,*

$$TAT^{-1} = \begin{bmatrix} A_1 & A_2 \\ A_3 & A_4 \end{bmatrix}, \; TB = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix},$$

$$TE = \begin{bmatrix} E_1 \\ 0 \end{bmatrix}, \; SCT^{-1} = \begin{bmatrix} C_1 & 0 \\ 0 & C_4 \end{bmatrix}, \quad SD = \begin{bmatrix} 0 \\ D_2 \end{bmatrix} \qquad (2.6)$$

where $C_1$ is *invertible*.

11

## Problem Formulation

- *After introducing the state and output transformations T and S, system (2.1) is converted into the following two systems*

$$\begin{cases} \dot{z}_1 = A_1 z_1 + A_2 z_2 + f_1(T^{-1}z, t) + B_1 u + E_1 \Delta\psi \\ w_1 = C_1 z_1 \end{cases} \qquad (2.6)$$

$$\begin{cases} \dot{z}_2 = A_3 z_1 + A_4 z_2 + f_2(T^{-1}z, t) + B_2 u \\ w_2 = C_4 z_2 + D_2 f_S \end{cases} \qquad (2.7)$$

where $f_1(T^{-1}z, t) = T_1 f(T^{-1}z, t)$ and $f_2(T^{-1}z, t) = T_2 f(T^{-1}z, t)$.

12

## Problem Formulation

**Lemma 2.2** *The pair ($A_4$,$C_4$) is detectable if and only if Assumption 2.2 holds.*

In order to apply the method developed in actuator FD lecture, we define a new state

$$z_3 = \int_0^{-t} w_2(\tau)d\tau$$

so that

$$\dot{z}_3(t) = C_4 z_2 + D_2 f_s \qquad (2.8)$$

13

## Problem Formulation

Equations (2.7) and (2.8) can be combined to form an augmented system of order n + p − 2r as

$$\begin{bmatrix} \dot{z}_2 \\ \dot{z}_3 \end{bmatrix} = \begin{bmatrix} A_4 & 0 \\ C_4 & 0 \end{bmatrix} \begin{bmatrix} z_2 \\ z_3 \end{bmatrix} + \begin{bmatrix} A_3 \\ 0 \end{bmatrix} z_1 + $$

$$+ \begin{bmatrix} f_2(T^{-1}z, t) \\ 0 \end{bmatrix} + \begin{bmatrix} B_2 \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ D_2 \end{bmatrix} f_s, \qquad (2.9)$$

$$w_3 = z_3.$$

14

## Problem Formulation

System (2.9) can then be rewritten in a more compact form as

$$\begin{cases} \dot{z}_0 = A_0 z_0 + \bar{A}_3 z_1 + \bar{f}_2(T^{-1}z,t) + B_0 u + D_0 f_s \\ w_3 = C_0 z_0, \end{cases} \qquad (2.10)$$

where $z_0 = \begin{bmatrix} z_2 \\ z_3 \end{bmatrix} \in \sim^{n+p-2r}$, $\bar{A}_3 = \begin{bmatrix} A_3 \\ 0 \end{bmatrix} \in \sim^{(n+p-2r)\times r}$,

$w_3 \in \sim^{p-r}$, $A_0 = \begin{bmatrix} A_4 & 0 \\ C_4 & 0 \end{bmatrix} \in \sim^{(n+p-2r)\times(n+p-2r)}$,

$B_0 = \begin{bmatrix} B_2 \\ 0 \end{bmatrix} \in \sim^{(n+p-2r)\times m}$, $D_0 = \begin{bmatrix} 0 \\ D_2 \end{bmatrix} \in \sim^{(n+p-2r)\times q}$

15

## Problem Formulation

$$\begin{cases} \dot{z}_0 = A_0 z_0 + \bar{A}_3 z_1 + \bar{f}_2(T^{-1}z,t) + B_0 u + D_0 f_s \\ w_3 = C_0 z_0, \end{cases} \qquad (2.10)$$

where

$C_0 = \begin{bmatrix} 0 \\ I_{p-r} \end{bmatrix}^T \in \sim^{(p-r)\times(n+p-2r)}$, $\bar{f}_2(T^{-1}z,t) = \begin{bmatrix} f_2(T^{-1}z,t) \\ 0 \end{bmatrix}$.

16

## Problem Formulation

Accordingly, System (2.6) can be rewritten as

$$\begin{cases} \dot{z}_1 = A_1 z_1 + \bar{A}_2 z_0 + f_1(T^{-1}z, t) + B_1 u + E_1 \Delta\psi \\ w_1 = C_1 z_1, \end{cases} \qquad (2.11)$$

where

$$\bar{A}_2 = \begin{bmatrix} A_2 \\ 0_{r\times(p-r)} \end{bmatrix}^T$$

17

## Sensor FD Scheme

For Subsystem (2.11), Sliding Mode Observer has the form as

$$\begin{cases} \dot{\hat{z}} = A_1 \hat{z}_1 + \bar{A}_2 \hat{z}_0 + f_1(T^{-1}\hat{z}, t) + B_1 u + (A_1 - A_1^s) C_1^{-1}(w_1 - \hat{w}_1) + v_1 \\ \hat{w}_1 = C_1 \hat{z}_1 \end{cases} \qquad (2.17)$$

where $A_1^s \in \sim^{r\times r}$ is a stable matrix which needs to be determined.

$\hat{z}$ is defined as $\hat{z} := col(C_1^{-1} w_1 \hat{z}_2)$

18

# Sensor FD Scheme

The discontinuous output error injection term $v_1$ is defined by

$$\bar{v}_1 = \begin{cases} k_1 \dfrac{P_1(C_1^{-1}w_1 - \hat{z}_1)}{\|P_1(C_1^{-1}w_1 - \hat{z}_1)\|} & \text{if} \quad C_1^{-1}w_1 \neq 0 \\ 0 & \text{otherwise,} \end{cases} \qquad (2.18)$$

where $k_1 = \|E_1\|\xi + \eta_1$ and $\eta_1$ is a positive scalar which needs to be determined.

19

# Luenberger Observer

For Subsystem (2.10), we design the following Luenberger observer:

$$\begin{cases} \dot{\hat{z}}_0 = A_0\hat{z}_0 + \bar{A}_3C_1^{-1}w_1 + \bar{f}_2(T^{-1}\hat{z},t) + B_0u + L_0(w_3 - \hat{w}_3) \\ \hat{w}_3 = C_0\hat{z}_0 \end{cases} \qquad (2.19)$$

where $L_0 \in \sim^{(n+p-2r)\times(p-r)}$ is the gain of the Luenberger observer

20

# Luenberger Observer

If the state estimation errors are defined as

$$e_1 = z_1 - \hat{z}_1 \text{ and } e_2 = z_2 - \hat{z}_2,$$

then the state estimation error dynamics, before the occurrence of sensor faults, can be obtained as

$$\dot{e}_1 = A_1^s e_1 + \overline{A}_2 e_0 + (f_1(T^{-1}z,t) - f_1(T^{-1}\hat{z},t)) + E_1 \Delta\psi - v_1 \tag{2.20}$$

$$\dot{e}_0 = (A_0 - L_0 C_0)e_0 + (\overline{f}_2(T^{-1}z,t) - \overline{f}_2(T^{-1}\hat{z},t). \tag{2.21}$$

21

# Luenberger Observer

- We now present Theorem 2.1 which establishes sufficient conditions for the existence of the proposed observers (2.17)–(2.19) and outlines a constructive design procedure.

22

***Theorem 2.1*** Given System (2.1) with Assumptions 2.1–2.4. When the system is free of sensor faults, the error dynamics (2.20) and (2.21) are asymptotically stable, if there exist matrices

$$A_1^s < 0,\ L_0,\ P_1 = P_1^T > 0 \text{ and } P_0 = P_0^T > 0$$

and positive scalars $\alpha_1$ and $\alpha_2$ such that

$$\Lambda := \begin{bmatrix} \Pi_1 + \frac{1}{\alpha_1} P_1 P_1 & P_1 \bar{A}_2 \\ \bar{A}_2^T P_1 & \Pi_2 + \frac{1}{\alpha_0} P_0 P_0 + aI_{n+p-2r} \end{bmatrix} < 0 \qquad (2.22)$$

where $\Pi_1 = A_1^{sT} P_1 + P_1 A_1^s$,

$$\Pi_2 = (A_0 - L_0 C_0)^T P_0 + P_0 (A_0 - L_0 C_0),$$

$$\alpha = \alpha_1 \chi_{f1} \|T^{-1}\|^2 + \alpha_2 \chi_{f1}^2 \|T^{-1}\|^2.$$

23

*Remark 2.2 The problem of finding matrices to satisfy Inequality (3.22) can be transformed into the following LMI feasibility problem using Schur complement*

$$\begin{bmatrix} X + X^T & P_1 & P_1 \bar{A}_2 & 0 \\ P_1 & -\alpha_1 I & 0 & 0 \\ \bar{A}_2^T P_1 & 0 & A_0^T P_0 + P_0 A_0 - C_0^T Y_0^T - Y_0 C_0 + aI & P_0 \\ 0 & 0 & P_0 & -\alpha_0 I \end{bmatrix} < 0$$

24

- ***Sensor FD scheme:*** *Actuator faults can be detected if the residual* $||e_{w3}||$ *exceeds a predefined threshold ς. Otherwise the system is healthy within the considered time. The detection time* $t_d(t_d \geq t_f)$ *is defined as the first time instant such that* $||e_{w3}||$ *is observed greater than ς.*

25

# Simulation Results

- In this section, the effectiveness of the proposed schemes in detecting and isolating actuator faults has been demonstrated considering an example of a single-link robotic arm with a revolute elastic joint.

The dynamics is described by

$$\begin{aligned} J_l\ddot{q}_1 + F_1\dot{q}_1 + k(q_1 - q_2) + mgl\sin q_1 = 0 \\ J_m\ddot{q}_2 + F_m\dot{q}_2 - k(q_1 - q_2) = u \end{aligned} \tag{2.51}$$

26

- Space shuttle remote manipulation system

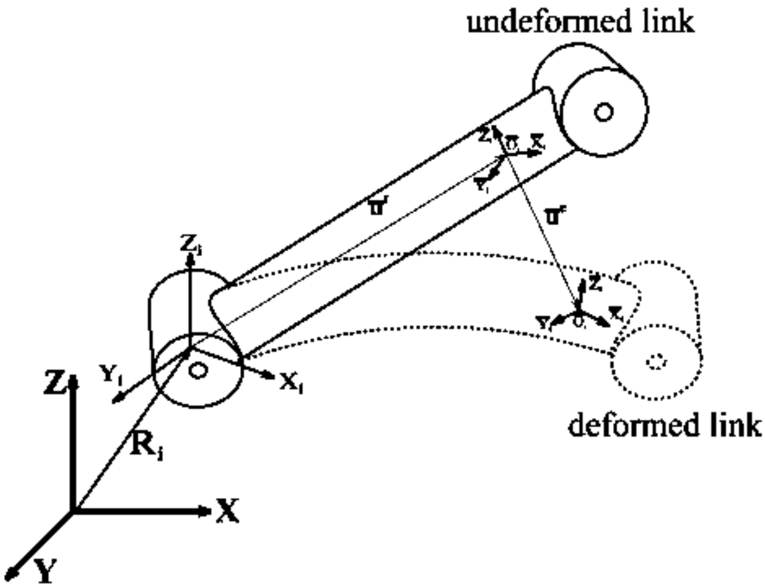
27

- Truck mounted concrete boom pump

28

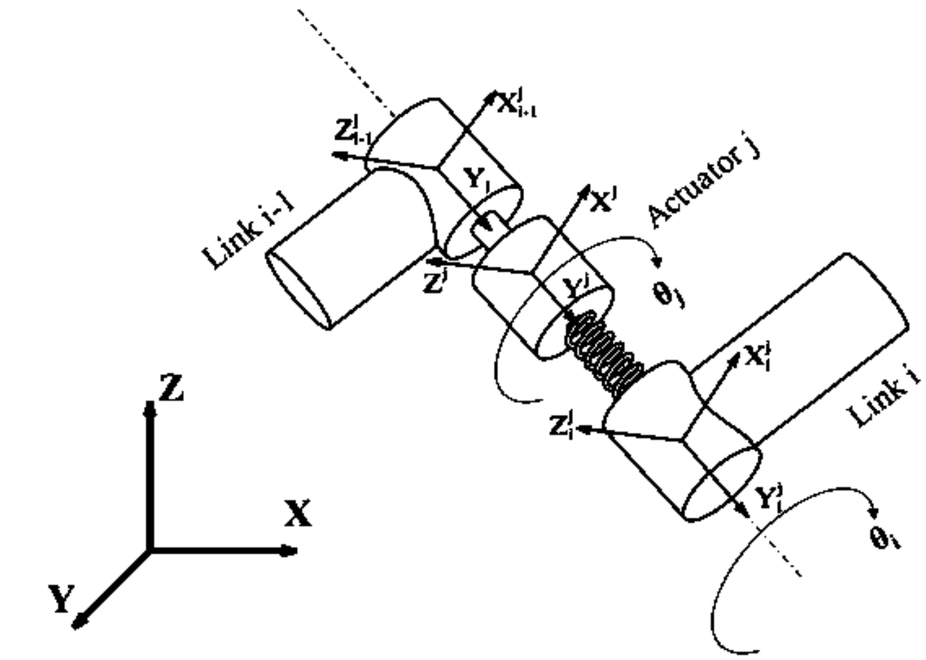
- Kuka DLR Light weight robot

29



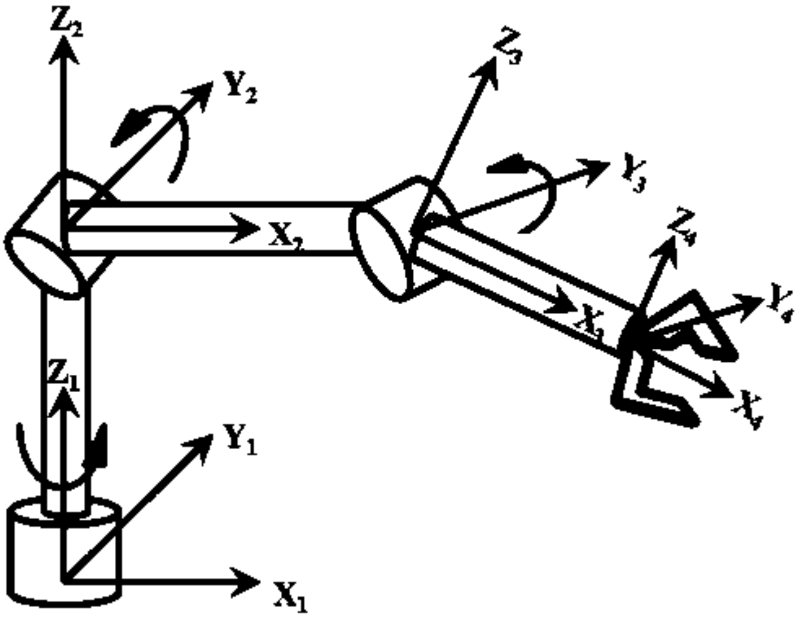

- Representation of an arbitrary point on a flexible link

30




- : Flexible joint j assembly

31




- Spatial RRR flexible manipulator

32

# Simulation Results

The dynamics is described by

$$J_l\ddot{q}_1 + F_1\dot{q}_1 + k(q_1 - q_2) + mgl\sin q_1 = 0$$
$$J_m\ddot{q}_2 + F_m\dot{q}_2 - k(q_1 - q_2) = u \quad (2.51)$$

where $q_1$ and $q_2$ denote the link position and the rotor position, respectively;

$u$ is the torque delivered by the motor;

$m$ is the linkmass, $l$ is the center of mass,

$J_m$ is the link inertia, $J_l$ is the motor rotor inertia,

33

# Simulation Results

The dynamics is described by

$$J_l\ddot{q}_1 + F_1\dot{q}_1 + k(q_1 - q_2) + mgl\sin q_1 = 0$$
$$J_m\ddot{q}_2 + F_m\dot{q}_2 - k(q_1 - q_2) = u \quad (2.51)$$

$F_m$ is the viscous friction coefficient,

$F_l$ is the viscous friction coefficient,

$k$ is the elastic constant, and $g$ is the gravity constant.

34

# Simulation Results

In the simulation, the values of these parameters are chosen as

$m = 4$, $l = 0.5$, $J_m = 1$, $J_l = 2$,

$F_m = 1$, $F_l = 0.5$, $k = 2$

and $g = 9.8$

(all in SI units).

35

Choosing $x_1 = q_1$, $x_2 = \dot{q}_1$, $x_3 = q_2$, $x_4 = \dot{q}_2$ and assuming that the link position, the link velocity and the rotor position can be measured, the dynamics (2.51) can be represented in the following state-space form as

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{-k}{J_l} & \frac{-F_l}{J_l} & \frac{k}{J_l} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{k}{J_m} & 0 & \frac{-k}{J_m} & \frac{-F_m}{J_m} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{-mgl}{J_l} \sin x_1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{J_m} \end{bmatrix} u + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \Delta\psi,$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 2 & 0 \\ 0 & 1 \end{bmatrix} f_s. \qquad (2.52)$$

36

# Simulation Results

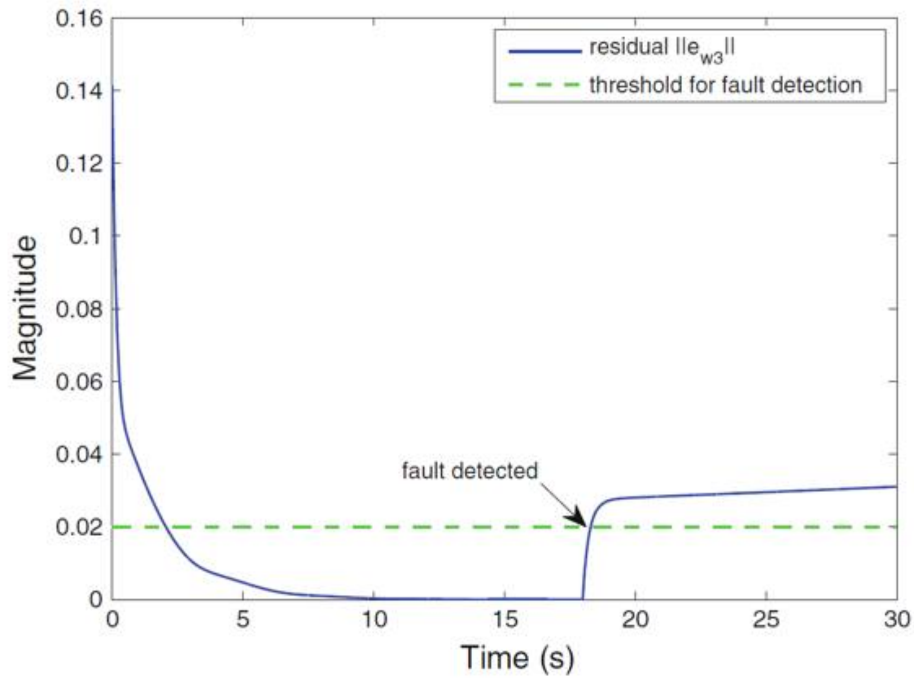
**Case-1** In this case the sensor faults are given as

$$f_{s1} = \begin{cases} 0 & ,t \le 18s \\ 0.05\exp(0.01t), & t \ge 18s \end{cases}$$

$$f_{s2} = 0, \forall t$$

37

Detection of the occurrence of sensor faults




38

# Simulation Results

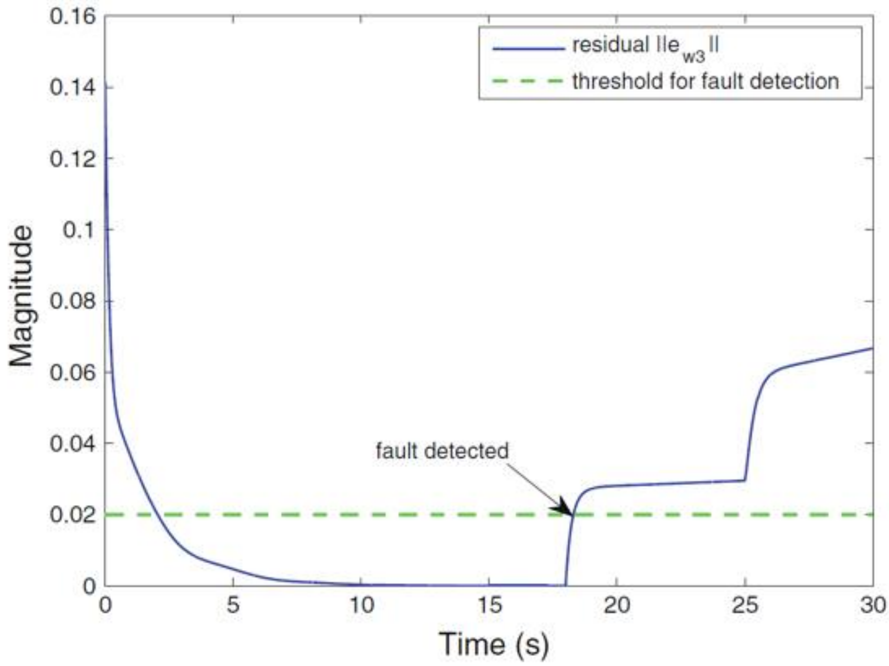
**Case-2** In this case the sensor faults are given as

$$f_{s1} = \begin{cases} 0 & ,t \le 18s \\ 0.05\exp(0.01t), & t \ge 18s \end{cases}$$

$$f_{s2} = \begin{cases} 0 & ,t \le 25s \\ 0.07\exp(0.03t), & t \ge 25s \end{cases}$$

39

Detection of the occurrence of sensor faults




40

# Conclusions

In this lecture, a new sensor FD scheme is presented. The proposed FD scheme essentially transforms the original system into two subsystems where subsystem-1 includes system uncertainties, but is free from sensor faults and subsystem-2 has sensor faults but without uncertainties. Using the integral observer-based approach, sensor faults in subsystem-2 are transformed into actuator faults and detected by designing a Luenberger observer for this subsystem.

41

# Conclusions

Its effectiveness has been demonstrated considering the example of a single-link robotic arm with a revolute elastic joint.

Simulation results confirm that the proposed method can effectively detect and isolate incipient sensor faults in the presence of system

42

**Modeling of systems and complexes**

Modeling and control of robotic systems

Kinematics of industrial robots

# Kinematics of Industrial Robots

Dr. Oleg Borisov

## Basic Concepts and Definitions: Joints and Generalized Coordinates

> **Kinematic Chain**
> The *kinematic chain* is used to describe the geometry of the robot manipulator. Ot represents a graphic representation of the sequence of manipulator links connected by joints.

There are two elementary types of 1-DOF joints

- revolte (joint coordinate is angular)
- prismatic (joint coordinat is linear)

Both joint coordinates are so-called *generalized coordinates*

$$q_i = \begin{cases} \theta_i, & \text{if the link } i \text{ is revolute,} \\ d_i, & \text{if the link } i \text{ is prisnatic.} \end{cases} \tag{1}$$

> **Configuration**
> A set of all the generalized coordinates of the manipulator, which uniquely determines it in the space, is called *configuration*.

## Basic Concepts and Definitions: FK and IK

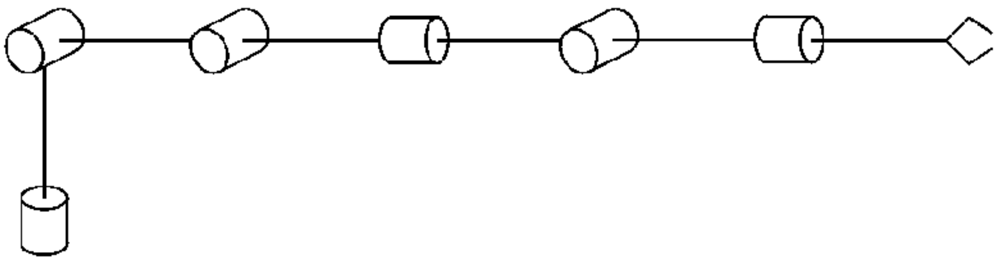
There are two fundamental tasks of the kinematics analysis

- forward kinematics
- inverse kinematics

> **Forward kinematics**
> The *forward kinematics (FK)* is to calculate the coordinates of the tool frame (its position and orientation) given the configuration of the robot.

> **Inverse kinematics**
> The *inverse kinematics (IK)* is to calculate the configuration of the robot given the coordinates of the tool frame (its position and orientation).

## Forward Kinematics: Algorithm



Kinematic chain of 6-DOF robot

1. Assigning frames to the links.
2. Determining Denavit-Hartenberg parameters
3. Forming homogeneous transformation matrices
4. Parametrization of rotation matrix

## Forward Kinematics: Assigning Frames

**Choice of $z_i$-axes**

Choose the axis $z_i$ so that it coincides with the axis of rotation or translational motion of the subsequent joint $i+1$ depending on its type. This means that the relative location of adjacent links (coordinate systems) will be determined precisely by the variable around (or along) this axis.

**Choice of $x_i$-axes**

Choose the axis $x_i$, $i = \{1, 2 \ldots, n-1\}$ so that the following two conditions are satisfied.

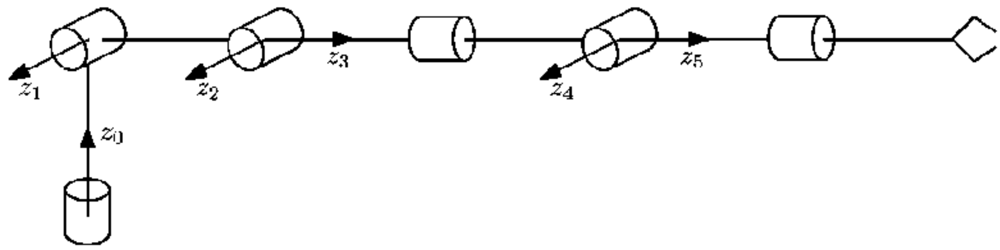
- The axis $x_i$ is perpendicular to the axis $z_{i-1}$.
- The axis $x_i$ intersects the axis $z_{i-1}$.

**Choice of $y_i$-axes**

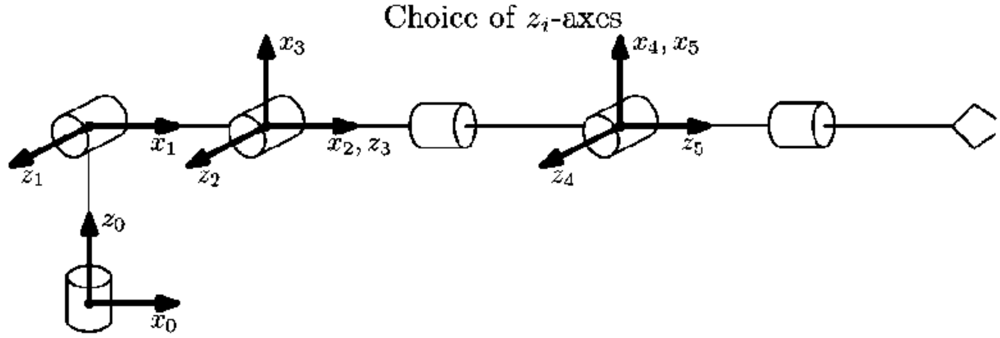
Choose the axis $y_i$ so that the frame given by the unit vectors $\vec{x}_i$, $\vec{y}_i$, $\vec{z}_i$ is right-handed, i.e. in the direction given by the vector product:

$$\vec{y}_i = \vec{z}_i \times \vec{x}_i. \quad (2)$$

## Forward Kinematics: Assigning Frames

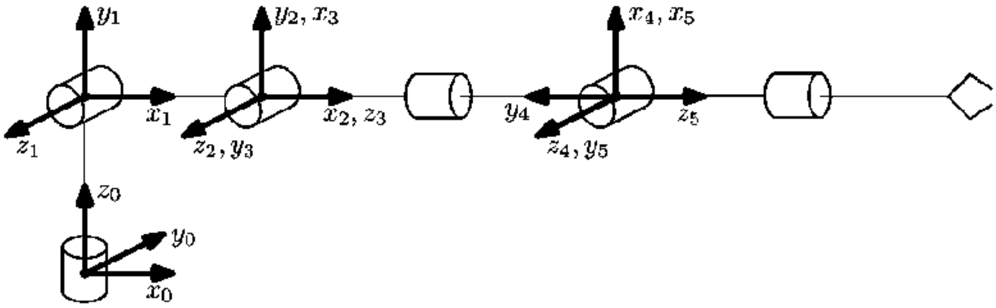



Choice of $z_i$-axes

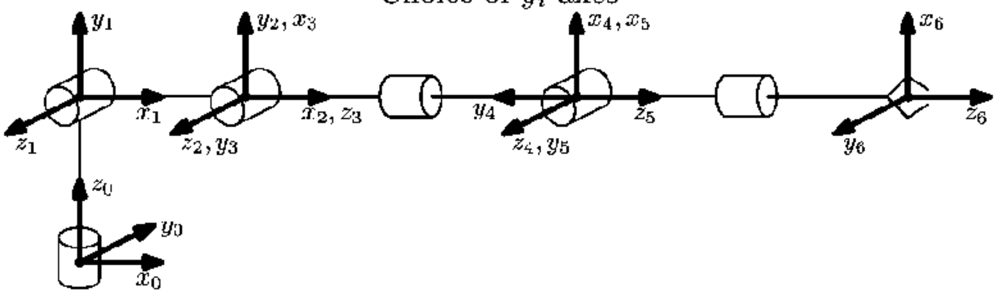



Choice of $x_i$-axes

## Forward Kinematics: Assigning Frames




Choice of $y_i$-axes



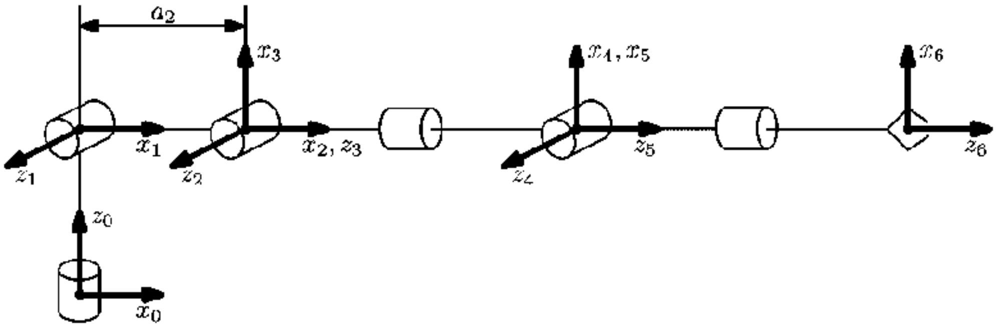

Choice of $n$-axes

## Forward Kinematics: DH parameters

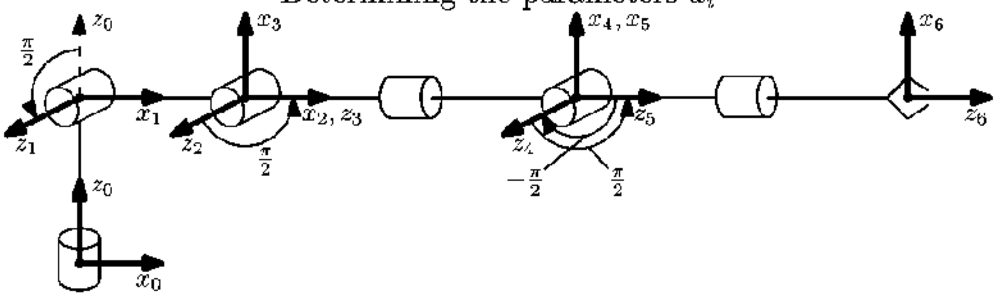
The Denavit-Hartenberg convention allows to reduce the number of coordinates that uniquely determine the body (its frame) in the space, from six to four, known as the *Denavite-Harteberg parameters* listed below.

- $a_i$ is the distance along the axis $x_i$ from $z_{i-1}$ to $z_i$
- $\alpha_i$ is the angle around the axis $x_i$ from $z_{i-1}$ to $z_i$
- $d_i$ is the distance along the axis $z_{i-1}$ from $x_{i-1}$ to $x_i$
- $\theta_i$ is the angle around the axis $z_{i-1}$ from $x_{i-1}$ to $x_i$

## Forward Kinematics: DH parameters

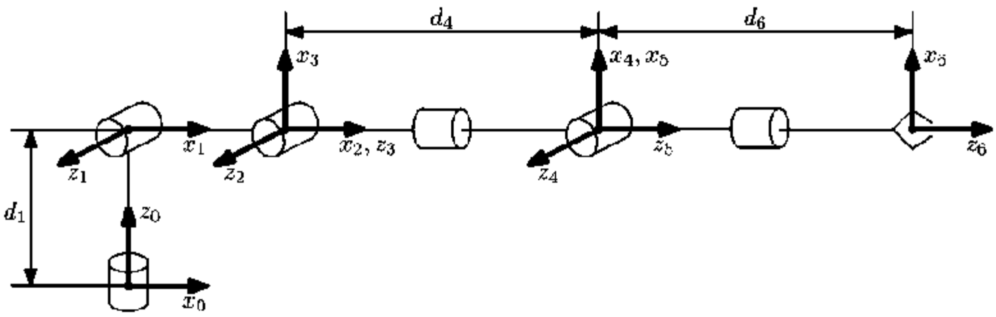



Determining the parameters $a_i$

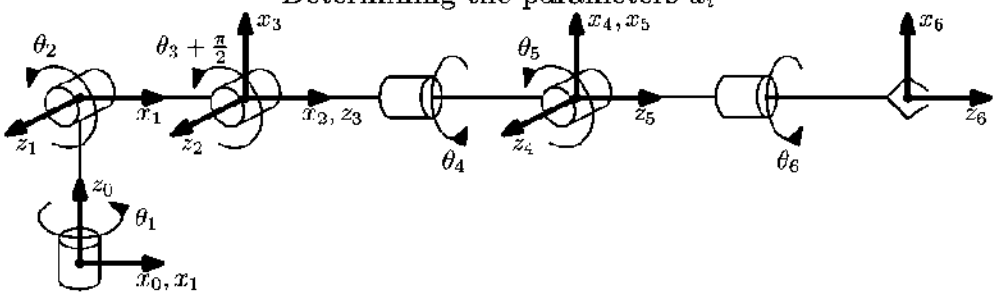



Determining the parameters $\alpha_i$

## Forward Kinematics: DH parameters




Determining the parameters $d_i$




Determining the parameters $\theta_i$

## Forward Kinematics: DH parameters

| Link, $i$ | $a_i$ | $\alpha_i$ | $d_i$ | $\theta_i$ |
|---|---|---|---|---|
| 1 | 0 | $\frac{\pi}{2}$ | $d_1$ | $\theta_1$ |
| 2 | $a_2$ | 0 | 0 | $\theta_2$ |
| 3 | 0 | $\frac{\pi}{2}$ | 0 | $\theta_3 + \frac{\pi}{2}$ |
| 4 | 0 | $-\frac{\pi}{2}$ | $d_4$ | $\theta_4$ |
| 5 | 0 | $\frac{\pi}{2}$ | 0 | $\theta_5$ |
| 6 | 0 | 0 | $d_6$ | $\theta_6$ |

DH parameters of the 6-DOF robot

## Forward Kinematics: HT Matrix

Consider to sets of coordinates $k^0$ and $k^n$ of the same point in the space expressed with respect to two frames $o_0x_0y_0z_0$ and $o_nx_ny_nz_n$, respectively:

$$k^0 = T_n^0 k^n, \tag{3}$$

where $T_n^0$ is the transformation carrying information about relative position and orientation of one frame with respect to another one.

### Homogeneous Transformation Matrixe

The matrix $T_n^0$ defining the relation between frames $o_0x_0y_0z_0$ and $o_nx_ny_nz_n$ is called *a homogeneous transformation (HT) matrix* and has the form

$$T_n^0 = \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} n_n^0 & s_n^0 & a_n^0 & p_n^0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} R_n^0 & p_n^0 \\ 0 & 1 \end{bmatrix}, \tag{4}$$

where the vectors $n_n^0$, $s_n^0$ and $a_n^0$ express directions of $x_n$, $y_n$ and $z_n$ with respect to $o_0x_0y_0z_0$, $R_n^0 \in \mathcal{SO}(3)$ is the rotation matrix of the frame $o_nx_ny_nz_n$ with respect to $o_0x_0y_0z_0$, $p_n^0 \in \mathbb{R}^3$ is the vector of linear displacement of the origin of $o_nx_ny_nz_n$ with respect to $o_0x_0y_0z_0$.

## Forward Kinematics: Properties of HT Matrix

1. The rotation by zero angle is determined by the identity matrix

$$R_{\beta=0} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = I. \tag{5}$$

2. Rotation in the negative direction is determined by

$$R_{-\beta} = R_\beta^{-1} = R_\beta^T. \tag{6}$$

3. There are three basic rotation matrices around $x$, $y$ and $z$ axes given as

$$R_{x,\beta} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\beta & -\sin\beta \\ 0 & \sin\beta & \cos\beta \end{bmatrix},\ R_{y,\beta} = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix},\ R_{z,\beta} = \begin{bmatrix} \cos\beta & -\sin\beta & 0 \\ \sin\beta & \cos\beta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

where $\beta$ is some angle.

## Forward Kinematics: Properties of HT Matrix

4. Serial rotations around several *current* axes are determined by multiplying on the right. For example, the transformation parametrized by Euler angles $\phi$, $\theta$ and $\psi$ is given as

$$\begin{aligned} R_{zyz} &= R_{z,\phi} R_{y,\theta} R_{z,\psi} = \\ &= \begin{bmatrix} c_\phi & -s_\phi & 0 \\ s_\phi & c_\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{bmatrix} \begin{bmatrix} c_\psi & -s_\psi & 0 \\ s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} = \\ &= \begin{bmatrix} c_\phi c_\theta c_\psi - s_\phi s_\psi & -c_\phi c_\theta s_\psi - s_\phi c_\psi & c_\phi s_\theta \\ s_\phi c_\theta c_\psi + c_\phi s_\psi & -s_\phi c_\theta s_\psi + c_\phi c_\psi & s_\phi s_\theta \\ -s_\theta c_\psi & s_\theta s_\psi & c_\theta \end{bmatrix}, \end{aligned} \tag{7}$$

where $c_\beta \equiv \cos\beta$, $s_\beta \equiv \sin\beta$, $\beta = \{\phi, \theta, \psi\}$.

## Forward Kinematics: Properties of HT Matrix

Using the DH convention form the homogeneous transformation matrices for each link as follows

$$T_i = T_{z,\theta_i} T_{z,d_i} T_{x,a_i} T_{x,\alpha_i} = \begin{bmatrix} R_{z,\theta_i} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I & p_{d_i} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I & p_{a_i} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_{x,\alpha_i} & 0 \\ 0 & 1 \end{bmatrix} =$$

$$= \begin{bmatrix} c_{\theta_i} & -s_{\theta_i} c_{\alpha_i} & s_{\theta_i} s_{\alpha_i} & a_i c_{\theta_i} \\ s_{\theta_i} & c_{\theta_i} c_{\alpha_i} & -c_{\theta_i} s_{\alpha_i} & a_i s_{\theta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad (8)$$

where $i$ is the link number, $R_{z,\theta_i}$ and $R_{x,\alpha_i}$ are the basic rotation matrices, $p_{d_i}$ and $p_{a_i}$ are vectors with nonzero components $p_z = d_i$ and $p_x = a_i$

$$R_{z,\theta_i} = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 \\ \sin\theta_i & \cos\theta_i & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad R_{x,\alpha_i} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha_i & -\sin\alpha_i \\ 0 & \sin\alpha_i & \cos\alpha_i \end{bmatrix},$$

$$p_{d_i} = \begin{bmatrix} 0 \\ 0 \\ d_i \end{bmatrix}, \qquad p_{a_i} = \begin{bmatrix} a_i \\ 0 \\ 0 \end{bmatrix}. \qquad (9)$$

## Forward Kinematics: Parametrization of Rotation Matrices

There different ways to parametrize rotation matrices

- Euler angles
- Roll-Pitch-Yaw angles
- Axis-Angle Representation

All of them are intended to reduce amount of parameters from 9 to 3.

## Forward Kinematics: Euler Angles

The matrix of $ZYZ$-transformation is given as

$$\begin{aligned} R_n^0(q) &= \begin{bmatrix} r_{11}(q) & r_{12}(q) & r_{13}(q) \\ r_{21}(q) & r_{22}(q) & r_{23}(q) \\ r_{31}(q) & r_{32}(q) & r_{33}(q) \end{bmatrix} = \\ &= \begin{bmatrix} c_\phi c_\theta c_\psi - s_\phi s_\psi & -c_\phi c_\theta s_\psi - s_\phi c_\psi & c_\phi s_\theta \\ s_\phi c_\theta c_\psi + c_\phi s_\psi & -s_\phi c_\theta s_\psi + c_\phi c_\psi & s_\phi s_\theta \\ -s_\theta c_\psi & s_\theta s_\psi & c_\theta \end{bmatrix}. \end{aligned} \tag{10}$$

Consider three cased depending on the entry $r_{33}(q)$.

## First Case

If $r_{33}(q) \neq \pm 1$ then $\sin\theta(q) \neq 0$. Use the Pythagorean trigonometric identity

$$\sin^2\theta(q) + \cos^2\theta(q) = 1, \tag{11}$$

$$\sin(\theta(q)) = \pm\sqrt{1 - \cos^2\theta(q)} = \pm\sqrt{1 - r_{33}(q)}, \tag{12}$$

from which it follows that $\theta(q)$ can be calculated as

$$\theta(q) = \operatorname{atan2}\left(\pm\sqrt{1 - r_{33}^2(q)}, r_{33}(q)\right). \tag{13}$$

Note that the remaining expressions to calculate $\phi(q)$ and $\psi(q)$ depend on the choice of the sign in front of the root in (13)

$$\phi(q) = \operatorname{atan2}\left(\pm r_{23}(q), \pm r_{13}(q)\right), \tag{14}$$

$$\psi(q) = \operatorname{atan2}\left(\pm r_{32}(q), \mp r_{31}(q)\right). \tag{15}$$

## Second Case

If $r_{33}(q) = 1$ then $\cos\theta(q) = 1$, $\sin\theta(q) = 0$, from which $\theta(q) = 0$ and as a result

$$
\begin{aligned}
R_n^0(q) &= \begin{bmatrix} c_\phi c_\psi - s_\phi s_\psi & -c_\phi s_\psi - s_\phi c_\psi & 0 \\ s_\phi c_\psi + c_\phi s_\psi & s_\phi s_\psi + c_\phi c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} = \\
&= \begin{bmatrix} c_{\phi+\psi} & -s_{\phi+\psi} & 0 \\ s_{\phi+\psi} & c_{\phi+\psi} & 0 \\ 0 & 0 & 1 \end{bmatrix} = \\
&= \begin{bmatrix} r_{11}(q) & r_{12}(q) & 0 \\ r_{21}(q) & r_{22}(q) & 0 \\ 0 & 0 & 1 \end{bmatrix}.
\end{aligned} \tag{16}
$$

This case leads to uncertainty, since only the sum $\phi(q) + \psi(q)$ can be computed

$$\phi(q) + \psi(q) = \text{atan2}\left(r_{21}(q), r_{11}(q)\right). \tag{17}$$

## Third Case

If $r_{33}(q) = -1$ then $\cos\theta(q) = -1$, $\sin\theta(q) = 0$, from which $\theta(q) = \pi$, as a result

$$
\begin{aligned}
R_n^0(q) &= \begin{bmatrix} -c_\phi c_\psi - s_\phi s_\psi & c_\phi s_\psi - s_\phi c_\psi & 0 \\ -s_\phi c_\psi + c_\phi s_\psi & s_\phi s_\psi + c_\phi c_\psi & 0 \\ 0 & 0 & -1 \end{bmatrix} = \\
&= \begin{bmatrix} -c_{\phi-\psi} & -s_{\phi-\psi} & 0 \\ s_{\phi-\psi} & c_{\phi-\psi} & 0 \\ 0 & 0 & -1 \end{bmatrix} = \\
&= \begin{bmatrix} r_{11}(q) & r_{12}(q) & 0 \\ r_{21}(q) & r_{22}(q) & 0 \\ 0 & 0 & -1 \end{bmatrix}.
\end{aligned} \tag{18}
$$

This case leads to uncertainty, since only the difference $\phi(q) - \psi(q)$ can be computed

$$\phi(q) - \psi(q) = \text{atan2}\left(-r_{12}(q), -r_{11}(q)\right). \tag{19}$$

## Inverse Kinematics

Initial data for IK are

- three linear coordinates (components of the vector $p_n^0$)
- three angular coordinates (e.g. Euler angles $\phi$, $\phi$ and $\psi$)
- DH parameters

The geometric (analytical) method of solving IK is to find explicit expressions using the apparatus of trigonometric functions, taking into account the kinematic scheme of the manipulator.

Consider kinematic decoupling approach applied to standard 6-DOF robot with spherical wrist. It is comprised of two subtasks

- position IK (to compute $q_1$, $q_2$ and $q_3$)
- orientation IK (to compute $q_4$, $q_5$ and $q_6$)

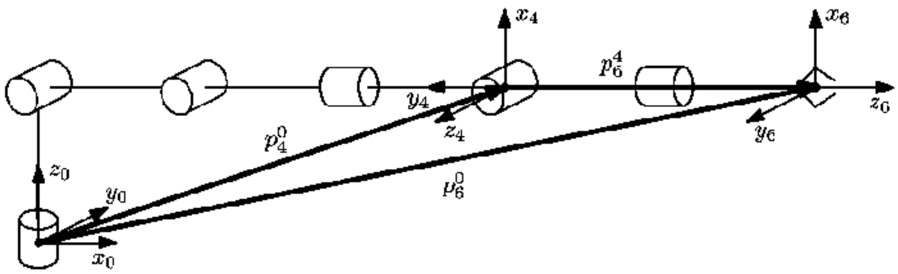
## Inverse Kinematics: Position IK

> **Spherical wrist**
> A spherical wrist is a kinematic scheme of the last three rotational joints such that their axes of rotation intersect at the same point.

The subtask is

- to determine relations between the given point of the end-effector and the point of three axes intersection
- to derive expressions for $q_1$, $q_2$ and $q_3$ given the point of three axes intersection

## Inverse Kinematics: Position IK




Kinematic decoupling

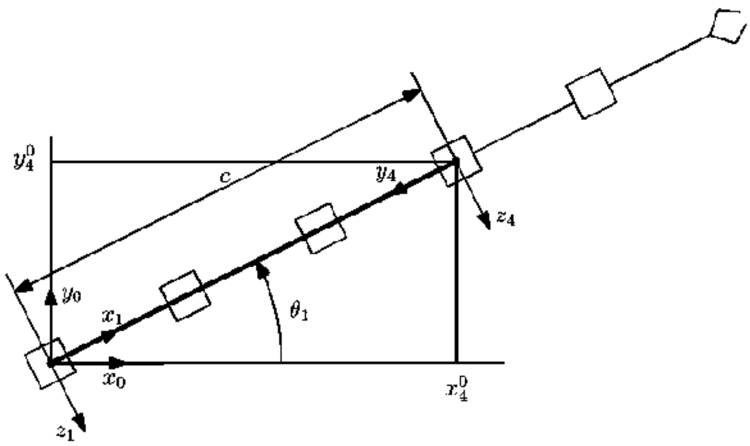
Using the sum of the vectors

$$p_6^0 = p_4^0 + d_6 R_6^0 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \qquad (20)$$

express coordinates of the point as

$$p_4^0 = p_6^0 - d_6 R_6^0 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} x_4^0 \\ y_4^0 \\ z_4^0 \end{bmatrix}. \qquad (21)$$

## Inverse Kinematics: Position IK




Vector $c$

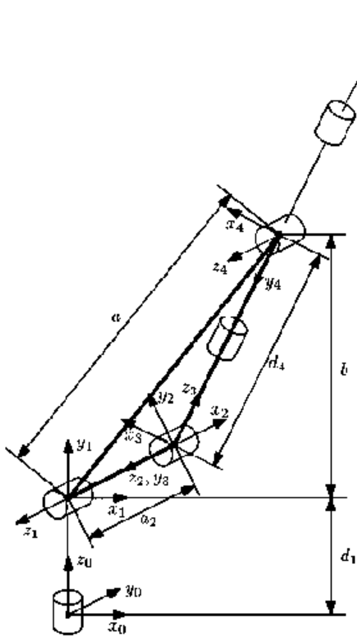
The first generalized coordinate can be computed as

$$\theta_1 = \text{atan2}(y_4^0, x_4^0) \qquad (22)$$

or

$$\theta_1 = \text{atan2}(y_4^0, x_4^0) + \pi. \qquad (23)$$

## Inverse Kinematics: Position IK

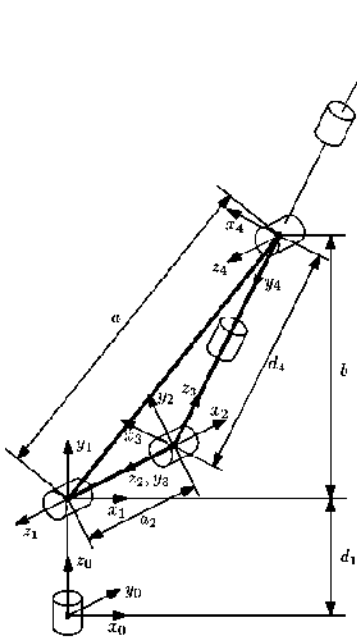



Vectors $a$ and $b$

Use the following notations

$$
\begin{aligned}
a &= \sqrt{\left(x_4^1\right)^2 + \left(y_4^1\right)^2 + \left(z_4^1\right)^2}, && (24)\\
b &= \left(z_4^0 - d_1\right), && (25)\\
c &= \sqrt{\left(x_4^0\right)^2 + \left(y_4^0\right)^2}. && (26)
\end{aligned}
$$

## Inverse Kinematics: Position IK




Vectors $a$ and $b$

Using the Pythagorean theorem write

$$a^2 = b^2 + c^2. \qquad (27)$$

Using the law of cosines write

$$
\begin{aligned}
a^2 &= a_2^2 + d_4^2 - 2a_2d_4\cos(\pi - \theta_3) = \\
&= a_2^2 + d_4^2 + 2a_2d_4\cos\theta_3. && (28)
\end{aligned}
$$

Combining the both expressions write

$$b^2 + c^2 = a_2^2 + d_4^2 + 2a_2d_4\cos\theta_3, \qquad (29)$$

from which express $\cos\theta_3$

$$\cos\theta_3 = \frac{b^2 + c^2 - a_2^2 - d_4^2}{2a_2d_4}. \qquad (30)$$

As a result the generalized coordinate $\theta_3$ can be computed as

$$\theta_3 = \operatorname{atan2}\left(\pm\sqrt{1 - \cos^2\theta_3}, \cos\theta_3\right). \qquad (31)$$

## Inverse Kinematics: Position IK

Consider difference between to angles

- angle $\alpha$ formed by $a$ and $c$
- angle $\beta$ formed by $a$ and $a_2$

Express the generalized coordinate $\theta_2$ as

$$\theta_2 = \alpha - \beta. \tag{32}$$

Taking into account trigonometric expressions

$$\tan\alpha = \frac{b}{c}, \tag{33}$$

$$\tan\beta = \frac{d_4 \sin\theta_3}{a_2 + d_4\cos\theta_3}, \tag{34}$$

rewrite (32) as

$$\theta_2 = \text{atan2}(b, c) - \text{atan2}(d_4 \sin\theta_3, a_2 + d_4\cos\theta_3). \tag{35}$$

## Inverse Kinematics: Orientation IK

Express the rotation matrix $R_6^0$ as

$$R_6^0 = R_3^0 R_6^3, \tag{36}$$

where $R_6^0$ is given, $R_3^0$ can be calculated solving FK. Express $R_6^3$ as

$$R_6^3 = \left(R_3^0\right)^{-1} R_6^0 = \left(R_3^0\right)^T R_6^0. \tag{37}$$

Consider $ZYZ$-transformation given by the Euler angles as

$$R_6^3 = R_{zyz} = R_{z,\phi} R_{y,\theta} R_{z,\psi} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}. \tag{38}$$

The remaining three generalized coordinates can be computed as

$$\theta_4 = \phi = \text{atan2}\left(\pm r_{23}, \pm r_{13}\right), \tag{39}$$

$$\theta_5 = \theta = \text{atan2}\left(\pm\sqrt{1 - r_{33}^2}, r_{33}\right), \tag{40}$$

$$\theta_6 = \psi = \text{atan2}\left(\pm r_{32}, \mp r_{31}\right). \tag{41}$$

## Inverse Kinematics: Summary

1. Solve forward kinematics
2. Calculate the coordinates of the intersection between the rotation axes given the coordinated of the tool
3. Solve position IK and get $\theta_1$, $\theta_2$ and $\theta_3$
4. Calculate $R_3^0$ from forward kinematics
5. Calculate matrix $R_6^3$
6. Solve orientation IK and get $\theta_4$, $\theta_5$ and $\theta_6$ as the Euler angles forming the matrix $R_6^3$

Dynamics of industrial robots

# Dynamics of Industrial Robots

Dr. Oleg Borisov

## Dynamical Model of Revolute Joint: Two Components

The electrical component of the model describes a circuit with the inductance, resistance and motor as

$$L\dot{i}(t) + Ri(t) = u(t) - K_\varepsilon \omega(t) = u(t) - K_\varepsilon \dot{\theta}(t), \tag{1}$$

where $L$, $R$, $i(t)$, $u(t)$ are the inductance, resistance, current and voltage of the armature, respectively, $K_\varepsilon$ is the back emf constant, $\omega(t)$, $\theta(t)$ are the angular velocity and position of the rotor, respectively.

The mechanical component of the model describes a gear train with the gear ratio $j$ connected with the motor as

$$J\ddot{\theta}(t) + K_f \dot{\theta}(t) = K_\mu i(t) - \mu_l(t), \tag{2}$$

where $J$ is the sum of the actuator and gear moments of inertia, $K_f$ is the friction constant, $K_\mu$ is the torque constant, $\mu_l(t) = \frac{1}{j}\mu_l(t)$, $\mu_l(t)$ is the load torque, $j$ is the gear ratio.

## Dynamical Model of Revolute Joint: Transfer Functions

Apply the Laplace transform and rewrite the model (1) and (2) as

$$\begin{aligned} (Ls + R)I(s) &= U(s) - K_\varepsilon s\Theta(s), & (3)\\ (Js + K_f)s\Theta(s) &= K_\mu I(s) - M_l(s). & (4) \end{aligned}$$

Taking into account (3) and (4) let us write the transfer function from the input $U(s)$ to the output $\Theta(s)$ with $M_l(s) = 0$

$$\frac{\Theta(s)}{U(s)} = \frac{K_\mu}{s((Ls + R)(Js + K_f) + K_\varepsilon K_\mu)}. \tag{5}$$

The transfer function from $M_l(s)$ to $\Theta(s)$ with $U(s) = 0$ is

$$\frac{\Theta(s)}{M_l(s)} = -\frac{Ls + R}{s((Ls + R)(Js + K_f) + K_\varepsilon K_\mu)}. \tag{6}$$

## Dynamical Model of Revolute Joint: Simplification

Now divide numerator and denominator of the transfer functions (5) and (6) by $R$

$$\frac{\Theta(s)}{U(s)} = \frac{\frac{K_\mu}{R}}{s\left(\left(\frac{L}{R}s+1\right)(Js+K_f)+\frac{K_\varepsilon K_\mu}{R}\right)}, \quad (7)$$

$$\frac{\Theta(s)}{M_l(s)} = -\frac{\frac{L}{R}s+1}{s\left(\left(\frac{L}{R}s+1\right)(Js+K_f)+\frac{K_\epsilon K_\mu}{R}\right)}. \quad (8)$$

Since the time constant of the electrical component is reasonably much smaller than the time constant of the mechanical one

$$\frac{L}{R} << \frac{J}{K_f}, \quad (9)$$

rewrite transfer functions (7) and (8)

$$\frac{\Theta(s)}{U(s)} \approx \frac{\frac{K_\mu}{R}}{s\left(Js+K_f+\frac{K_\varepsilon K_\mu}{R}\right)}, \quad \frac{\Theta(s)}{M_l(s)} \approx \frac{-1}{s\left(Js+K_f+\frac{K_\epsilon K_\mu}{R}\right)}. \quad (10)$$

## Dynamical Model of Revolute Joint: The Resultant Model

Define new notations for these transfer functions

$$\frac{\Theta(s)}{M_u(s)} \approx \frac{1}{s(Js+K)}, \quad (11)$$

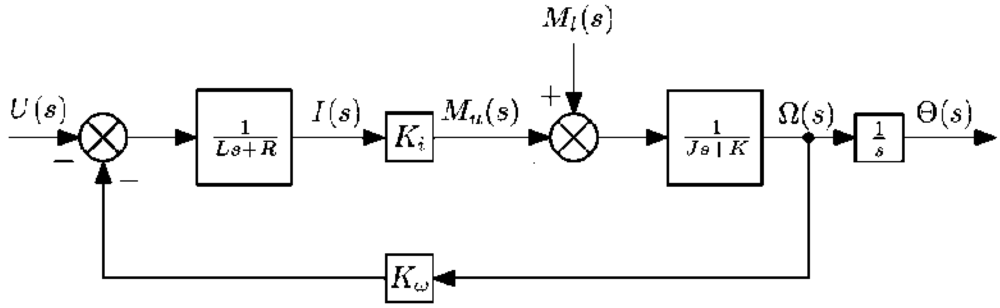
$$\frac{\Theta(s)}{M_l(s)} \approx \frac{1}{s(Js+K)}, \quad (12)$$

where $M_u(s) = \frac{K_\mu}{R}U(s)$, $K = K_f + \frac{K_\epsilon K_\mu}{R}$.

Combining transfer functions (11) and (12) we get

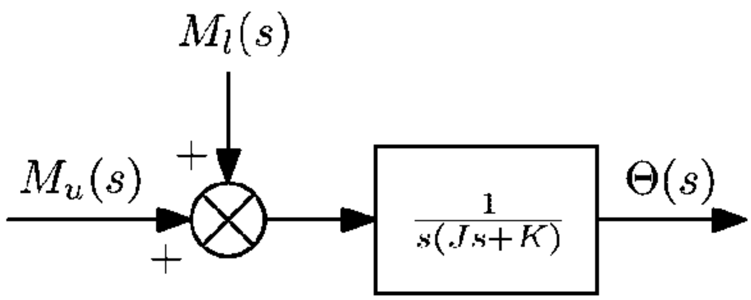
$$\Theta(s) = \frac{1}{s(Js+K)}\left(M_u(s) - M_l(s)\right) = P(s)\left(M_u(s) - M_l(s)\right). \quad (13)$$

## Dynamical Model of Revolute Joint: Initial Scheme




Initial scheme of the revolute joint model

## Dynamical Model of Revolute Joint: Simplified Scheme




Simplified scheme of the revolute joint model

## Dynamical Model of the Robot: Euler-Lagrange Equation

Dynamics of mechanical systems can be described by the *Euler-Lagrange equation* as

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = \mu_i, \tag{14}$$

where $L$ is the Lagrangian, $q_i$, $\dot{q}_i$ are the generalized coordinates and velocities, $\mu_i$ are the generalized torques applied to the joints.

The Laungragian $L$ can be computed as

$$L = K - P, \tag{15}$$

where $K$ and $L$ are the full kinetic and potential energies of the system, respectively.

## Dynamical Model of the Robot: Kinetic Energy

The kinetic energy of the link is comprised of the linear and angular components

$$K_i = \frac{1}{2} m_i |v_i|^2 + \frac{1}{2} \omega_i^T I_i^0 \omega_i, \tag{16}$$

where $m_i$ is the mass of the link, $v_i$ is the linear velocity of the center of mass, $\omega_i$ is the angular velocity of the frame assigned with the link, $I_i^0$ is the inertia tensor with respect to the base frame.

Express the linear and angular velocities using the Jacobian matrix

$$v_i = J_{v_i}(q)\dot{q}, \tag{17}$$

$$\omega_i = J_{\omega_i}(q)\dot{q}. \tag{18}$$

## Dynamical Model of the Robot: Kinetic Energy

Express the inertia tensor as

$$I_i^0 = R_i I R_i^T, \tag{19}$$

where $R_i$ is the rotation matrix between the base and link frames, $I$ is the inertia tensor with respect to the link frame given as $I = \begin{bmatrix} i_{11} & i_{12} & i_{13} \\ i_{21} & i_{22} & i_{23} \\ i_{31} & i_{32} & i_{33} \end{bmatrix}$,

where the elements are defined as

$$i_{11} = \iiint (y^2 + z^2)\rho(x, y, z)dxdydz, \quad i_{12} = i_{21} = -\iiint xy\rho(x, y, z)dxdydz,$$

$$i_{12} = \iiint (x^2 + z^2)\rho(x, y, z)dxdydz, \quad i_{13} = i_{31} = -\iiint xz\rho(x, y, z)dxdydz,$$

$$i_{13} = \iiint (x^2 + y^2)\rho(x, y, z)dxdydz, \quad i_{23} = i_{32} = -\iiint yz\rho(x, y, z)dxdydz,$$

where $\rho(x, y, z)$ is the funcition of mass density.

Rewrite the kinetic energy as

$$K_i = \frac{1}{2} m_i \dot{q}^T J_{v_i}^T J_{v_i} \dot{q} + \frac{1}{2} \dot{q}^T J_{\omega_i}^T R_i I R_i^T J_{\omega_i} \dot{q}. \tag{20}$$

## Dynamical Model of the Robot: Full Energy

The full kinetic energy of the robot can be computed as

$$K = \frac{1}{2} \dot{q}^T \sum_{i=1}^{n} \left( m_i J_{v_i}^T J_{v_i} + J_{\omega_i}^T R_i I R_i^T J_{\omega_i} \right) \dot{q} = \frac{1}{2} \dot{q}^T \Lambda(q) \dot{q}. \tag{21}$$

The potential energy of the each link is computed as

$$P_i = m_i g^T p_i, \tag{22}$$

whre $m_i$ is the mass of the link, $g$ is the vector defining the direction of the gravitation with respect to the base frame, $p_i$ is the radius-vector to the center of mass of the link expressed with respect to the base frame.

The full potential energy of the robot can be computed as

$$P = \sum_{i=1}^{n} m_i g^T p_i. \tag{23}$$

## Dynamical Model of the Robot: Model of Multilink System

Substitute the kinetic and potential energies to the Lengrangian

$$L = \frac{1}{2}\dot{q}^T \Lambda(q)\dot{q} - \sum_{i=1}^{n} m_i g^T p_i. \tag{24}$$

Substitute the Langrangian to the Euler-Langrange equation

$$\Lambda(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \mu, \tag{25}$$

where $\Lambda(q) \in \mathbb{R}^{n\times n}$ is the symmetrical matrix of inertia, $C(q,\dot{q}) \in \mathbb{R}^{n\times 1}$ is the matrix of Coriolis forces, $G(q) \in \mathbb{R}^{n\times 1}$ is the vector of gravitational forces.

## Dynamical Model of the Robot: Actuator Dynamics Revised

Write the dynamical model of the actuator dynamics as follows

$$J_i\ddot{\theta}_i(t) + F_i\dot{\theta}_i(t) = K_i\frac{u_i(t)}{r_i} - \mu_i(t), \tag{26}$$

where $F_i = K_f$, $K_i = K_\mu$, $r_i = R$, $\frac{u_i(t)}{r_i} = i(t)$, $i = \{1, 2, \dots n\}$ is the number of the link.

Take into account gear box

$$q_i = \frac{\theta_i}{j_i}. \tag{27}$$

Rewrite the actuator dynamics as

$$j_i^2 J_i\ddot{q}_i(t) + j_i^2 F_i\dot{q}_i(t) = j_i K_i\frac{u_i(t)}{r_i} - \bar{\mu}_i(t), \tag{28}$$

where $\mu_i = \mu_l$ for the link $i$.

## Dynamical Model of the Robot: Actuator Dynamics Augmentation

Add the actuator dynamical model to the model of the mechanical system and get

$$\Gamma(q)\ddot{q} + C(q,\dot{q})\dot{q} + F\dot{q} + G(q) = u, \tag{29}$$

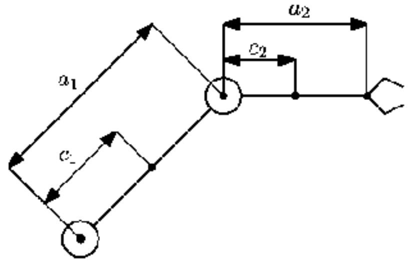
where the matrix $\Gamma(q)$ is of the form

$$\Gamma(q) = \Lambda(q) + J = \Lambda(q) + \begin{bmatrix} j_1^2 J_1 & 0 & \dots & 0 \\ 0 & j_2^2 J_2 & \dots & 0 \\ \vdots & \dots & \ddots & \vdots \\ 0 & 0 & 0 & j_n^2 J_n \end{bmatrix}, \tag{30}$$

where the friction vector and vector of control inputs are given respectively as

$$\begin{bmatrix} j_1^2 F_1 \\ j_2^2 F_2 \\ \vdots \\ j_n^2 F_n \end{bmatrix}, \quad \begin{bmatrix} j_1 K_1 \frac{u_1(t)}{R_1}(t) \\ j_2 K_2 \frac{u_2(t)}{R_2}(t) \\ \vdots \\ j_n K_n \frac{u_n(t)}{R_n}(t) \end{bmatrix}. \tag{31}$$

## Example of Two-Link Planar Manipulator: Jacobian matrices




Kinematic chain of two-link robot

Write relations between linear end-effector velocities and generalized ones using the notion of the Jacobian matrix as follows

$$v_1 = J_{v,1}\dot{q}, \quad v_2 = J_{v,2}\dot{q}, \tag{32}$$

where

$$J_{v,1} = \begin{bmatrix} -a_1 \sin q_1 & 0 \\ c_1 \cos q_1 & 0 \\ 0 & 0 \end{bmatrix}, \; J_{v,2} = \begin{bmatrix} -a_1 \sin q_1 - c_2 \sin(q_1+q_2) & -c_2 \sin(q_1+q_2) \\ a_1 \cos q_1 + c_2 \cos(q_1+q_2) & c_2 \cos(q_1+q_2) \\ 0 & 0 \end{bmatrix}.$$

# Example of Two-Link Planar Manipulator: Kinetic Energy

The kinetic energy is comprised of translational and rotational components. Let us address them separately. The translational component caused by the linear velocity can be computed as

$$K_{tr} = \frac{m_1 v_1^T v_1}{2} + \frac{m_2 v_2^T v_2}{2} = \underbrace{0.5\dot{q}^T m_1 J_{v,1}^T J_{v,1} \dot{q}}_{\text{1st link}} + \underbrace{0.5\dot{q}^T m_2 J_{v,2}^T J_{v,2} \dot{q}}_{\text{2nd link}} \tag{33}$$

The rotational component caused by the angular velocity can be computed as

$$K_{rt} = \underbrace{0.5\dot{q}^T I_1 \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \dot{q}}_{\text{1st link}} + \underbrace{0.5\dot{q}^T I_2 \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \dot{q}}_{\text{2nd link}} \tag{34}$$

# Example of Two-Link Planar Manipulator: Inertia Matrix

The inertia matrix $\Lambda(q)$ becomes of the form

$$\Lambda(q) = m_1 J_{v,1}^T J_{v,1} + m_2 J_{v,2}^T J_{v,2} + \begin{bmatrix} I_1 + I_2 & I_2 \\ I_2 & I_2 \end{bmatrix} = \begin{bmatrix} \lambda_{11} & \lambda_{12} \\ \lambda_{21} & \lambda_{22} \end{bmatrix}$$

$$= \begin{bmatrix} m_1 c_1^2 + m_2(a_1^2 + c_2^2 + 2a_1 c_2^2 + 2a_1 c_2 \cos q_2) + I_1 + I_2 & m_2(c_2^2 + a_1 c_2 \cos q_2) + I_2 \\ m_2(c_2^2 + a_1 c_2 \cos q_2) + I_2 & m_2 c_2^2 + I_2 \end{bmatrix}$$

## Example of Two-Link Planar Manipulator: Matrix of Coriolis Forces

Each element of the matrix of Coriolis forces $C(q)$ can be calculated using the equation

$$c_{kj} = \sum_{i=1}^{n} 0.5 \left( \frac{\partial \lambda_{kj}}{\partial q_i} + \frac{\partial \lambda_{ki}}{\partial q_j} - \frac{\partial \lambda_{ij}}{\partial q_k} \right) \dot{q}_i \tag{35}$$

The matrix of Coriolis forces $C(q)$ becomes of the form

$$C(q) = \begin{bmatrix} -m_2 a_1 c_2 \sin q_2 \dot{q}_2 & -m_2 a_1 c_2 \sin q_2 (\dot{q}_2 + \dot{q}_1) \\ m_2 a_1 c_2 \sin q_2 \dot{q}_1 & 0 \end{bmatrix} \tag{36}$$

## Example of Two-Link Planar Manipulator: Vector of Gravitational Forces

Each element of the vector of gravitational forces $G(q)$ can be calculated using the equation

$$g_i = \frac{\partial P}{\partial q_i} \tag{37}$$

The vector of gravitational forces $G(q)$ becomes of the form

$$G(q) = \begin{bmatrix} (m_1 c_1 + m_2 a_1) g \cos q_1 + m_2 c_2 g \cos(q_1 + q_2) \\ m_2 c_2 \cos(q_1 + q_2) \end{bmatrix} \tag{38}$$

## Example of Two-Link Planar Manipulator: Resultant Model

The resultant model of the two-link robot is

$$\lambda_{11}\ddot{q}_1 + \lambda_{12}\ddot{q}_2 + c_{11}\dot{q}_1 + c_{12}\dot{q}_2 + (m_1c_1 + m_2a_1)g\cos q_1 + m_2c_2g\cos(q_1+q_2) = \mu_1$$
$$\lambda_{21}\ddot{q}_1 - \lambda_{22}\ddot{q}_2 + c_{21}\dot{q}_1 + m_2c_2\cos(q_1+q_2) = \mu_2$$

## Summary

- Dynamical models of industrial robots allow to describe and take into account (designing a control law) physical processes specific to them
- The simplified model of the revolute joint can be represented by the transfer function of the relative degree 2
- The dynamical model of the industrial robot can be derived using the Euler-Lagrange approach

Motion planning for industrial robots

# Motion Planning for Industrial Robots

Dr. Oleg Borisov

## Basic Concepts and Definitions: Configuration Space

**Configuration**

*A configuration* $q$ is a set of all intermediate generalized coordinates (joint variables).

**Configuration space**

*Configirations space* $\mathcal{Q}$ is a set of all possible configurations $q$

$$\mathcal{Q} = \{q\}. \quad (1)$$

## Basic Concepts and Definitions: Workspace

**Workspace**

*Workspace* $\mathcal{W}$ is a set of points, which belong to the robot itself and the reachable environment including all the obstacles

$$\mathcal{R}(q) \subset \mathcal{W}, \quad \mathcal{O} \subset \mathcal{W}, \tag{2}$$

where $\mathcal{R}(q)$ is space occupied by the robot and $\mathcal{O}$ is space occupied by the obstacles.

In case of a planar manipulator which movements are constrained by the plane

$$\mathcal{W} \subset \mathbb{R}^2, \tag{3}$$

its workspace has two-dimensional.

In case of a spatial manipulator, which is able to move along three orthogonal axes

$$\mathcal{W} \subset \mathbb{R}^3, \tag{4}$$

its workspace is three-dimensional.

## Basic Concepts and Definitions: Collision-Free Space

**Collision-Free Space**

Space corresponding to collision of the robot with some obstacle is defined as follows

$$\mathcal{Q}_\times = \{q \in \mathcal{Q} | \mathcal{R}(q) \cap \mathcal{O} \neq 0\}, \tag{5}$$

from which collision-free space can be expressed as

$$\mathcal{Q}_0 = \mathcal{Q} \setminus \mathcal{Q}_\times. \tag{6}$$

## Basic Concepts and Definitions: Path and Trajectory

> **Path Planning**
> Path planning is a process of searching a cosecutive set of configurations within collision-free space connecting the initial configuration with the given final one.

> **Trajectory Planning**
> Trajectory planning is a process of time parametrization of the path, i.e. computation of reference functions of time for generalized coordinates, velocities and accelerations.

## Path Planning: Exact Cell Decomposition Approach

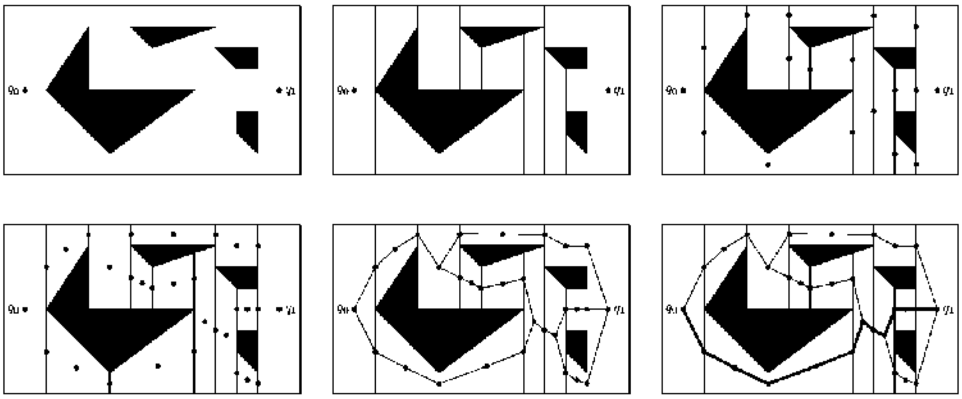
> **Exact Cell Decomposition**
> The idea of *exact cell decomposition* is to divide whole free configuration space on triangle or trapezoid cells and to construct a graph. Its nodes are represented by centers of the cells and its links are common sides between adjacent cells.

In case of exact cell decomposition there are two types of cells

- white cells correspond to the collision-free space
- black cells correspond to the collision space

Then given initial and final configurations, search of consecutive transition from one white cell to another one is carrying out to connect these two configurations and avoid all the black cells.

## Path Planning: Exact Cell Decomposition Approach



Steps of Exact Cell Decomposition Approach

## Path Planning: Approximate Cell Decomposition Approach

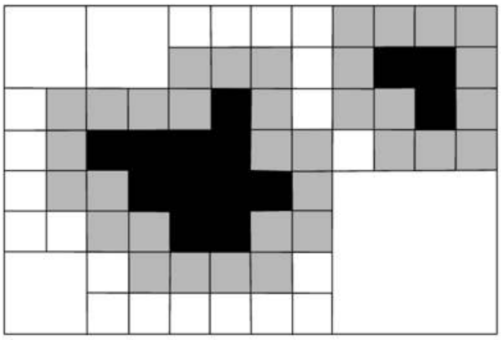
> **Approximate Cell Decomposition**
> Difference of the *approximate cell decomposition* with respect to its "exact" version is that instead of the whole configuration space its subset is divided on cells. So, the remaining space could include also slight parts of collision-free space, which is caused by complex shape of the collision space.

In case of approximate cell decomposition there are two types of cells

- white cells correspond to the collision-free space
- black cells correspond to the collision space
- gray cells correspond to the both spaces

While searching a path could pass both white and gray cells. If it touches grays cells, additional cell decomposition should be carried out until the path connecting initial and final configurations goes through white cells only.

## Path Planning: Approximate Cell Decomposition Approach



Approximate Cell Decomposition

## Path Planning: Potential Field Approach

**Potential Field Approach**

he robot is considered as a material point moving in a configuration space under influence of a potential field function $P(q)$. It has attraction component $P_a(q)$ assigned with the final configuration and repulsive component $P_r(q)$ assigned with the collision space

$$P(q) = P_a(q) + P_r(q). \tag{7}$$

## Path Planning: Potential Field Approach

Set the global minimum of the function $P(q)$ as the attraction component $P_a(q)$

$$P_a(q) = \frac{1}{2} k_a ||q - q_d||^2, \quad (8)$$

where $q$, $q_d$ are the current and desired configurations, respectively, $k_a$ is the scaling factor.

The repulsive component $P_r(q)$ ensures singularity of the function $P(q)$ when the material point is approaching the collision space

$$P_r(q) = \begin{cases} \frac{1}{2} k_r \left( \frac{1}{\delta(q)} - \frac{1}{\delta_0} \right)^2 & \text{if} \quad \delta(q) \leq \delta_0, \\ 0 & \text{if} \quad \delta(q) > \delta_0, \end{cases} \quad (9)$$

where $k_r$ is the scaling factor, $\delta(q)$ is the shortest distance from the current configuration to the collision space, $\delta_0$ is the minimum value.

## Path Planning: Potential Field Approach

The gradient descent algorithm can be used to plan a path

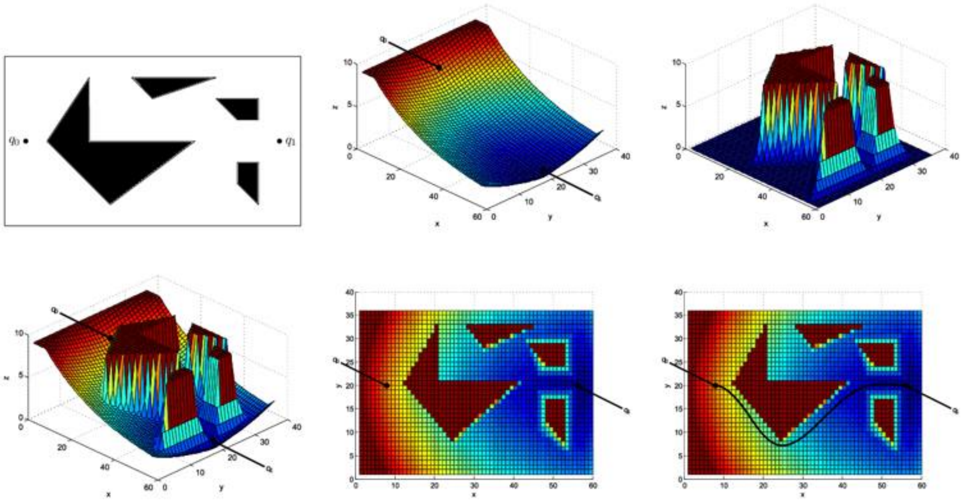
$$q_{j+1} = q_j - \gamma_j \nabla P(q_j), \quad (10)$$

where $\nabla P(q) = \begin{bmatrix} \frac{\partial P}{\partial q_1} & \frac{\partial P}{\partial q_2} \dots & \frac{\partial P}{q_n} \end{bmatrix}^T$, $\gamma_j$ is a iterative step, which can be either fixed, fractioned, or calculated in the direction of the fastest descent as

$$\gamma_j = \text{argmin}_j \, P(q_j - \gamma \nabla P(q_j)). \quad (11)$$

The main disadvantage of the potential field approach is possibility to stuack at the local minimum instead of the global one. So called random motion approach is used to avoid this issue.

## Path Planning: Potential Field Approach



Steps of Potential Field Approach

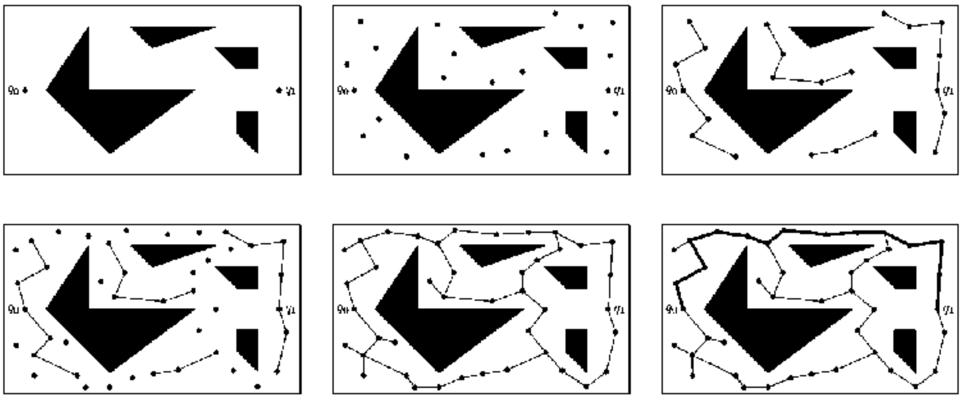
## Path Planning: Probabilistic Roadmap Approach

**Probabilistic Roadmap Approach**

Probabilistic roadmap approach is useful for fast path generation. It is based on the usage of random samples from the configuration space.

1. Several nodes (samples) are chosen randomly from the configuration space. Each node is assigned with a particular configuration.
2. Adjacent nodes are being connected between each other within the specified norm in the configuration space.
3. The fisrt two steps are repeated to cover sufficiently large area between the initial and final configurations.
4. A cosequtive set of samples are chosen to connect the initial and final configurations.

## Path Planning: Probabilistic Roadmap Approach



Steps of Probabilistic Roadmap Approach

## Trajectory Planning: Spline Functions Approach

### Spline Functions Approach

The idea of this approach is to interpolate generalized coordinates, velocities and accelerations between the reference points using the polynomials of the form

$$
\begin{aligned}
q_i(t) &= a_{l,i}t^l + a_{l-1,i}t^{l-1} + \cdots + a_{2,i}t^2 + a_{1,i}t + a_{0,i}, & (12)\\
\dot{q}_i(t) &= la_{l,i}t^{l-1} + (l-1)a_{l-1,i}t^{l-2} + \cdots + 2a_{2,i}t + a_{1,i}, & (13)\\
\ddot{q}_i(t) &= l(l-1)a_{l,i}t^{l-2} + (l-1)(l-2)a_{l-1,i}t^{l-3} + \cdots + 2a_{2,i}, & (14)
\end{aligned}
$$

where the degree $l$ and coefficients $a_{ji}$, $j = \{1, 2, \ldots, l\}$ are calculated depending on the constraints and continuity requirements on the trajectory.

1. Divide the whole trakectory on several elementary subtrajectories.
2. Compute relative time functions $\tau_i$ for each subtrjactory.
3. Apply constraints and continuity requirements on the trajectory.
4. Determine the highst polynomial degree for each subtrajectory.
5. Solve matrix equation to compute coefficients of all the polynomials.

## Trajectory Planning: Single Subtrajectory Case

Only initial and final configurations are given. No intermediate requirements.

Consider the following constraints for each link of the robot

$$\begin{aligned} q_i(t_0) &= \vartheta_0, \quad \dot{q}_i(t_0) = v_0, \quad \ddot{q}_i(t_0) = \alpha_0, & (15) \\ q_i(t_1) &= \vartheta_1, \quad \dot{q}_i(t_1) = v_1, \quad \ddot{q}_i(t_1) = \alpha_1. & (16) \end{aligned}$$

Choose the polynomial to interpolate intermediate values of the generalized coordinates

$$\vartheta(t) = a_5 t^5 + a_4 t^4 + a_3 t^3 + a_2 t^2 + a_1 t + a_0. \qquad (17)$$

Calculate the first and second derivatives of this polynomial to interpolate values of generalized velocities and accelerations

$$\begin{aligned} \dot{\vartheta}(t) &= v(t) = 5a_5 t^4 + 4a_4 t^3 + 3a_3 t^2 + 2a_2 t + a_1, & (18) \\ \ddot{\vartheta}(t) &= \alpha(t) = 20a_5 t^3 + 12a_4 t^2 + 6a_3 t + 2a_2. & (19) \end{aligned}$$

## Trajectory Planning: Single Subtrajectory Case

Write the system of equations taking into account the imposed constraits and continuity requirements as follows

$$\begin{cases} \vartheta_0 = a_5 t_0^5 + a_4 t_0^4 + a_3 t_0^3 + a_2 t_0^2 + a_1 t_0 + a_0, \\ v_0 = 5a_5 t_0^4 + 4a_4 t_0^3 + 3a_3 t_0^2 + 2a_2 t_0 + a_1, \\ \alpha_0 = 20a_5 t_0^3 + 12a_4 t_0^2 + 6a_3 t_0 + 2a_2, \\ \vartheta_1 = a_5 t_1^5 + a_4 t_1^4 + a_3 t_1^3 + a_2 t_1^2 + a_1 t_1 + a_0, \\ v_1 = 5a_5 t_1^4 + 4a_4 t_1^3 + 3a_3 t_1^2 + 2a_2 t_1 + a_1, \\ \alpha_1 = 20a_5 t_1^3 + 12a_4 t_1^2 + 6a_3 t_1 + 2a_2. \end{cases} \qquad (20)$$

Rewrite this system in matrix form as

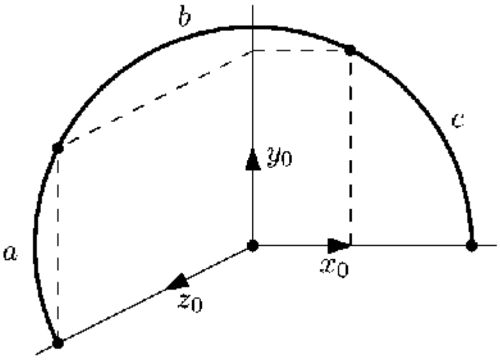
$$\underbrace{\begin{bmatrix} \vartheta_0 \\ v_0 \\ \alpha_0 \\ \vartheta_1 \\ v_1 \\ \alpha_1 \end{bmatrix}}_{\varrho} = \underbrace{\begin{bmatrix} t_0^5 & t_0^4 & t_0^3 & t_0^2 & t_0 & 1 \\ 5t_0^4 & 4t_0^3 & 3t_0^2 & 2t_0 & 1 & 0 \\ 20t_0^3 & 12t_0^2 & 6t_0 & 2 & 0 & 0 \\ t_1^5 & t_1^4 & t_1^3 & t_1^2 & t_1 & 1 \\ 5t_1^4 & 4t_1^3 & 3t_1^2 & 2t_1 & 1 & 0 \\ 20t_1^3 & 12t_1^2 & 6t_1 & 2 & 0 & 0 \end{bmatrix}}_{T} \underbrace{\begin{bmatrix} a_5 \\ a_4 \\ a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix}}_{\varsigma}, \qquad (21)$$

from which the vector of unknown coefficients can be easily expressed as

## Trajectory Planning: Multiple Subtrajectory Case

Consider a trajectory comprised of three subtrajectories as follows

- Leaving $(a)$
- Transition $(b)$
- Approach $(c)$




Trajectory comprised of three segments

## Trajectory Planning: Multiple Subtrajectory Case

Consider the following constraints for each link of the robot

$$\begin{aligned} q_i(t_0) &= \vartheta_0, \quad q_i(t_1) = \vartheta_1, \quad q_i(t_2) = \vartheta_2, \quad q_i(t_3) = \vartheta_3, && (23) \\ \dot{q}_i(t_0) &= v_0, \quad \ddot{q}_i(t_0) = \alpha_0, \quad \dot{q}_i(t_3) = v_3, \quad \ddot{q}_i(t_3) = \alpha_3. && (24) \end{aligned}$$

Use the relative time functions for each subtrajectory

$$\tau_a = \frac{t - t_0}{t_1 - t_0}, \qquad \tau_b = \frac{t - t_1}{t_2 - t_1}, \qquad \tau_c = \frac{t - t_2}{t_3 - t_2}, \tag{25}$$

where $t_0$, $t_1$, $t_2$, $t_3$ are the given time moments of passing all the reference configurations.

Impose continuity requirements to get a smooth trajectory

$$v_a(1) = v_b(0), \quad \alpha_a(1) = \alpha_b(0), \quad v_b(1) = v_c(0), \quad \alpha_b(1) = \alpha_c(0). \tag{26}$$

Taking into account the relative time functions rewrite constraints on the trajectory

$$\begin{aligned} \vartheta_a(0) &= \vartheta_0, \quad \vartheta_b(0) = \vartheta_1, \quad \vartheta_c(0) = \vartheta_2, && (27) \\ \vartheta_a(1) &= \vartheta_1, \quad \vartheta_b(1) = \vartheta_2, \quad \vartheta_c(1) = \vartheta_3, && (28) \end{aligned}$$

and continuity requirementsas follows

## Trajectory Planning: Multiple Subtrajectory Case

Choose the polynomial to interpolate intermediate values of the generalized coordinates within each subtrajectory

$$\vartheta_a(\tau_a) = a_4\tau_a^4 + a_3\tau_a^3 + a_2\tau_a^2 + a_1\tau_a + a_0, \tag{30}$$

$$\vartheta_b(\tau_b) = b_3\tau_b^3 + b_2\tau_b^2 + b_1\tau_b + b_0, \tag{31}$$

$$\vartheta_c(\tau_c) = c_4\tau_c^4 + c_3\tau_c^3 + c_2\tau_c^2 + c_1\tau_c + c_0, \tag{32}$$

Calculate the first derivative of these polynomials to interpolate values of generalized velocities

$$\dot\vartheta_a(\tau_a) = v_a(\tau_a) = 4a_4\tau_a^3 + 3a_3\tau_a^2 + 2a_2\tau_a + a_1, \tag{33}$$

$$\dot\vartheta_b(\tau_b) = v_b(\tau_b) = 3b_3\tau_b^2 + 2b_2\tau_b + b_1, \tag{34}$$

$$\dot\vartheta_c(\tau_c) = v_c(\tau_c) = 4c_4\tau_c^3 + 3c_3\tau_c^2 + 2c_2\tau_c + c_1, \tag{35}$$

Calculate the second derivative of these polynomials to interpolate values of generalized accelerations

$$\ddot\vartheta_a(\tau_a) = \alpha_a(\tau_a) = 12a_4\tau_a^2 + 6a_3\tau_a + 2a_2, \tag{36}$$

$$\ddot\vartheta_b(\tau_b) = \alpha_b(\tau_b) = 6b_3\tau_b + 2b_2, \tag{37}$$

$$\ddot\vartheta_c(\tau_c) = \alpha_c(\tau_c) = 12c_4\tau_c^2 + 6c_3\tau_c + 2c_2, \tag{38}$$

## Trajectory Planning: Multiple Subtrajectory Case

Write the system of equations taking into account the imposed constraits and continuity requirements as follows

$$\begin{cases} \vartheta_0 = a_0, \\ v_0 = a_1, \\ \alpha_0 = 2a_2, \\ \vartheta_1 = a_4 + a_3 + a_2 + a_1 + a_0, \\ \vartheta_1 = b_0, \\ 0 = 4a_4 + 3a_3 + 2a_2 + a_1 - b_1, \\ 0 = 12a_4 + 6a_3 + 2a_2 - 2b_2, \\ \vartheta_2 = b_3 + b_2 + b_1 + b_0, \\ \vartheta_2 = c_0, \\ 0 = 3b_3 + 2b_2 + b_1 - c_1, \\ 0 = 6b_3 + 2b_2 - 2c_2, \\ \vartheta_3 = c_4 + c_3 + c_2 + c_1 + c_0, \\ v_3 = 4c_4 + 3c_3 + 2c_2 + c_1, \\ \alpha_3 = 12c_4 + 6c_3 + 2c_2. \end{cases} \tag{39}$$

## Trajectory Planning: Multiple Subtrajectory Case

Rewrite this system in matrix form as

$$\underbrace{\begin{bmatrix}\vartheta_0\\ v_0\\ \alpha_0\\ \vartheta_1\\ \vartheta_1\\ 0\\ 0\\ \vartheta_2\\ \vartheta_2\\ 0\\ 0\\ \vartheta_3\\ v_3\\ \alpha_3\end{bmatrix}}_{\varrho} = \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0\\ 4 & 3 & 2 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0\\ 12 & 6 & 2 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1\\ 0 & 0 & 0 & 0 & 0 & 3 & 2 & 1 & 0 & 0 & 0 & 0 & -1 & 0\\ 0 & 0 & 0 & 0 & 0 & 6 & 2 & 0 & 0 & 0 & 0 & -2 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 3 & 2 & 1 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 12 & 6 & 2 & 0 & 0 \end{bmatrix}}_{T} \underbrace{\begin{bmatrix}a_4\\ a_3\\ a_2\\ a_1\\ a_0\\ b_3\\ b_2\\ b_1\\ b_0\\ c_4\\ c_3\\ c_2\\ c_1\\ c_0\end{bmatrix}}_{\varsigma}, \qquad (40)$$

from which the vector of unknown coefficients can be easily expressed as

$$\varsigma = T^{-1}\varrho. \qquad (41)$$

## Arc Approximation Algorithm of Spatial Movements

**Research Objective**

This study focuses on spatial motion planning algorithms, which allows to characterize sophisticated reference paths in 3D space and simplify the way how they can be given. The key point used in this study is approximation of a sequence of points by a sequence of arcs within a specified $\delta$-region.

In industry such algorithm can be applied for such tasks as surface finishing, engraving and welding. The last operation represents the main interest of this research.

## Problem


Mitsubishi RV-3SDB

**Objective**

The purpose is automated code generating to move the end-effector along some counters specified by the input bitmap image or 3D model.

After extracting coordinates of initial points sequence they already can be programmed using trivial point-to-point motion, but it might lead to some issues.

- significant input data (robot controller overload)
- decrease of the motion velocity (reconfiguration at each reference point)

## Arc Approximation Algorithm

**Basic Idea**

This approach is based on the feasibility of the standard software to move the end-effector along an arc, specified with only three points. This basic motion provided by the internal software is more natural then complex combinations of multiple linear point-to-point movements. As a result, the robot reconfigures only three times at the reference points forming this arc. Such solution allows to reduce the code size and increase the velocity.

## Planar Planning

Consider three reference points

$$p_1 = \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}, \quad p_2 = \begin{bmatrix} x_2 \\ y_2 \end{bmatrix}, \quad p_3 = \begin{bmatrix} x_3 \\ y_3 \end{bmatrix}. \tag{42}$$

All intermediate points between $p_1$, $p_2$ and $p_3$ should belong to a corresponding arc within some $\delta_{arc}$-region.

Consider two lines $p_1$–$p_2$ and $p_2$–$p_3$. In order to find coordinates of the arc center $c = \begin{bmatrix} x_c \\ y_c \end{bmatrix}$ consider three cases.

### Case 1

If $x_2 = x_3$ and $x_1 \neq x_2$ then

$$y_c = \frac{y_2 + y_3}{2}, \quad x_c = -k_1 \frac{y_c - (y_1 + y_2)}{2} + \frac{x_1 + x_2}{2},$$

where $k_1$ is the slope of the line $(x_1; y_1)$–$(x_2; y_2)$ given by $k_1 = \frac{y_2 - y_1}{x_2 - x_1}$.

### Case 2

If $x_1 = x_2$ and $x_2 \neq x_3$ then

$$y_c = \frac{y_1 + y_2}{2}, \quad x_c = -k_2 \frac{y_c - (y_2 + y_3)}{2} + \frac{x_2 + x_3}{2},$$

where $k_2$ is the slope of the line $(x_2; y_2)$–$(x_3; y_3)$ given by $k_2 = \frac{y_3 - y_2}{x_3 - x_2}$.

### Case 3

If all $x$-coordinates are distinct, then

$$x_c = \frac{k_1 k_2 (y_1 - y_3) + k_2 (x_1 + x_2) - k_1 (x_2 + x_3)}{2(k_2 - k_1)}, \tag{43}$$

$$y_c = -\frac{x_c - \frac{x_1 + x_2}{2}}{k_1} + \frac{y_1 + y_2}{2}, \tag{44}$$

where $k_1$ and $k_2$ are given above.

Calculate distance from a forth point $p_4 = \begin{bmatrix} x_4 \\ y_4 \end{bmatrix}$ to the arc formed by $p_1$, $p_2$ and $p_3$ as follows

$$d_{arc} = \left| \sqrt{(x_c - x_4)^2 + (y_c - y_4)^2} - r \right|, \tag{45}$$

where $r = \sqrt{(x_1 - x_c)^2 + (y_1 - y_c)^2}$ is the radius of the arc.

As a result we get a sequence of arcs each specified by three consecutive points. Such point list can be used together with the operator MVR P1 P2 P3, which allows to move along an arc specified by three reference points.

## Spatial Planning

Consider three points that do not lie on the same line. Coordinates of vectors specified in the Cartesian space are defined as

$$p_1^0 = \begin{bmatrix} x_1^0 \\ y_1^0 \\ z_1^0 \end{bmatrix}, \quad p_2^0 = \begin{bmatrix} x_2^0 \\ y_2^0 \\ z_2^0 \end{bmatrix}, \quad p_3^0 = \begin{bmatrix} x_3^0 \\ y_3^0 \\ z_3^0 \end{bmatrix}. \tag{46}$$

Consider two coordinate systems denoted as $x_0 y_0 z_0 o_0$ and $x_1 y_1 z_1 o_1$. Derive a normal to the plane $x_1 y_1 o_1$ through a cross product

$$n = \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix} = \left(p_2^0 - p_1^0\right) \times \left(p_3^0 - p_1^0\right). \tag{47}$$

Then calculate a unit vector

$$z = \begin{bmatrix} z_x \\ z_y \\ z_z \end{bmatrix} = \frac{n}{\sqrt{n_x^2 + n_y^2 + n_z^2}}. \tag{48}$$

Compute the rotational transformation as $R_1^0 = R_{z,\alpha} R_{y,\beta}$, where the angles $\alpha$ and $\beta$ can be calculated as follows

$$\alpha = \operatorname{atan2}\left(\frac{z_y}{\sqrt{z_x^2+z_y^2}}, \frac{z_x}{\sqrt{z_x^2+z_y^2}}\right), \quad \beta = \operatorname{atan2}\left(\sqrt{z_x^2+z_y^2}, z_z\right).$$

Substitute $\alpha$ and $\beta$ into the rotation matrices around $z-$ and $y$-axes

$$R_1^0 = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix}. \qquad (49)$$

Calculate coordinates of the reference points with respect to the local coordinate system using the rotation matrix

$$p_1^1 = R_0^1 p_1^0, \quad p_2^1 = R_0^1 p_2^0, \quad p_3^1 = R_0^1 p_3^0, \qquad (50)$$

where $R_0^1 = [R_1^0]^T$.

Denote coordinates as follows

$$p_1^1 = \begin{bmatrix} x_1^1 \\ y_1^1 \\ z_1^1 \end{bmatrix}, \quad p_2^1 = \begin{bmatrix} x_2^1 \\ y_2^1 \\ z_2^1 \end{bmatrix}, \quad p_3^1 = \begin{bmatrix} x_3^1 \\ y_3^1 \\ z_3^1 \end{bmatrix}. \qquad (51)$$

In order to find coordinates of the arc center $c^1 = \begin{bmatrix} x_c^1 \\ y_c^1 \\ z_c^1 \end{bmatrix}$ consider three cases.

### Case 1

If $x_2^1 = x_3^1$ and $x_1^1 \neq x_2^1$ then

$$y_c^1 = \frac{y_2^1 + y_3^1}{2}, \quad x_c^1 = -k_1 \frac{y_c^1 - (y_1^1 + y_2^1)}{2} + \frac{x_1^1 + x_2^1}{2},$$

where $k_1$ is the slope of the line $(x_1^1; y_1^1)$–$(x_2^1; y_2^1)$ given by $k_1 = \frac{y_2^1 - y_1^1}{x_2^1 - x_1^1}$.

### Case 2

If $x_1^1 = x_2^1$ and $x_2^1 \neq x_3^1$ then

$$y_c^1 = \frac{y_1^1 + y_2^1}{2}, \quad x_c^1 = -k_2 \frac{y_c^1 - (y_2^1 + y_3^1)}{2} + \frac{x_2^1 + x_3^1}{2},$$

where $k_2$ is the slope of the line $(x_2^1; y_2^1)$–$(x_3^1; y_3^1)$ given by $k_2 = \frac{y_3^1 - y_2^1}{x_3^1 - x_2^1}$.

### Case 3

If all $x$-coordinates are distinct, then

$$x_c^1 = \frac{k_1 k_2 (y_1^1 - y_3^1) + k_2 (x_1^1 + x_2^1) - k_1 (x_2^1 + x_3^1)}{2(k_2 - k_1)}, \qquad (52)$$

$$y_c^1 = -\frac{x_c^1 - \frac{x_1^1 + x_2^1}{2}}{k_1} + \frac{y_1^1 + y_2^1}{2}, \qquad (53)$$

where $k_1$ and $k_2$ are given above.

The third $z$-coordinate can be derived trivially as

$$z_c^1 = z_1^1 = z_2^1 = z_3^1. \tag{54}$$

Express coordinates of the center with respect to the base coordinate system

$$c^0 = \begin{bmatrix} x_c^0 \\ y_c^0 \\ z_c^0 \end{bmatrix} = R_1^0 c^1. \tag{55}$$

The equation of a plane is given as

$$n_x x + n_y y + n_z z + n_0 = 0, \tag{56}$$

where $n_0 = -(n_x x_3^0 + n_y y_3^0 + n_z z_3^0)$.

Distances from a forth point $p_4 = \begin{bmatrix} x_4^0 \\ y_4^0 \\ z_4^0 \end{bmatrix}$ respectively to the plane $d_{plane}$ and to the arc formed by $p_1$, $p_2$ and $p_3$ can be computed as

$$d_{plane} = \frac{|n_x x_4^0 + n_y y_4^0 + n_z z_4^0 + n_0|}{\sqrt{n_x^2 + n_y^2 + n_z^2}}, \tag{57}$$

$$d_{arc} = \left| \sqrt{(x_c^0 - x_4^0)^2 + (y_c^0 - y_4^0)^2 + (z_c^0 - z_4^0)^2} - r \right|, \tag{58}$$

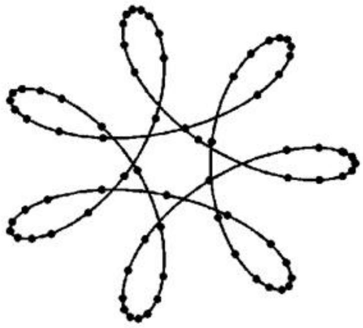
where $r = \sqrt{(x_1^0 - x_c^0)^2 + (y_1^0 - y_c^0)^2 + (z_1^0 - z_c^0)^2}$ is the radius of the arc.

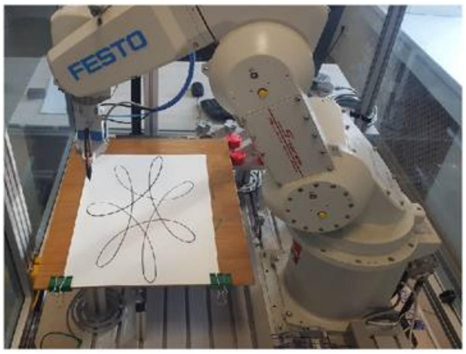
Then all points should be processed and checked on belonging them to a particular plane and arc within the specified $\delta_{plane}$- and $\delta_{arc}$-regions, respectively. As a result of this procedure, a sequence of three-points-sets each specifying a particular arc should be obtained.

## Experimental Approval

Experimental Approval

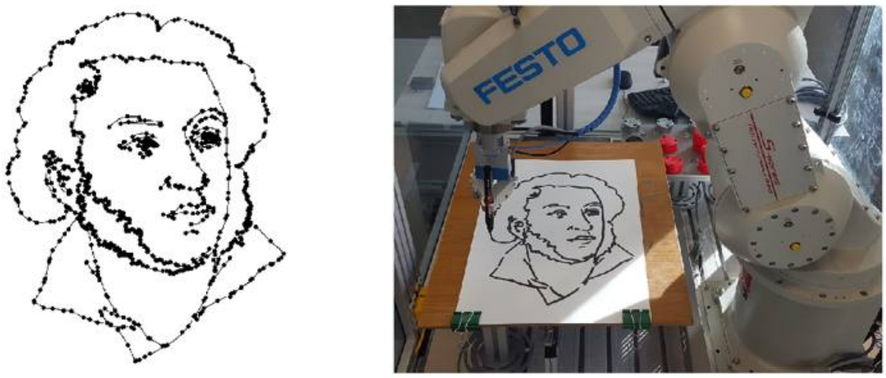
## Experimental Results: Planar Planning





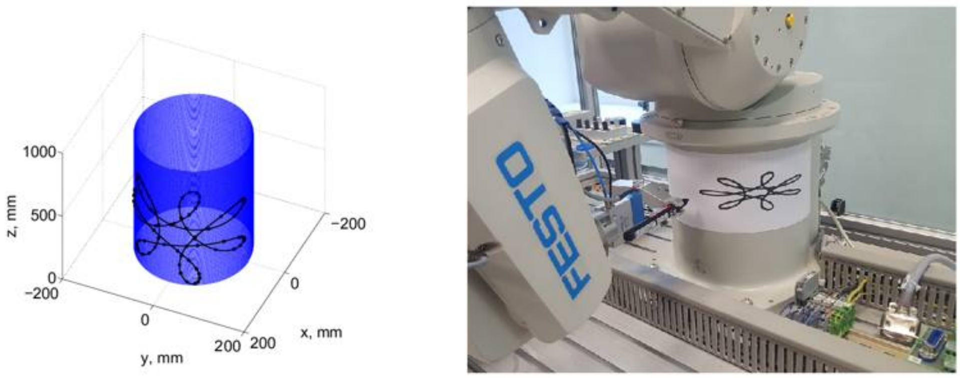

A hypotrochoid drawn by the robot on a flat surface

## Experimental Results: Planar Planning



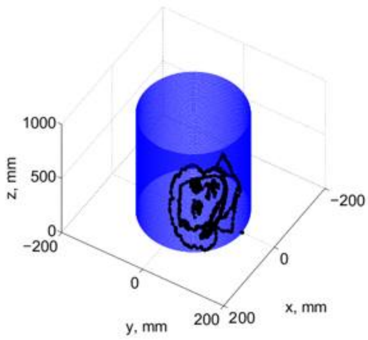

A portrait of Alexander Pushkin drawn by the robot on a flat surface

## Experimental Results: Spatial Planning




A hypotrochoid drawn by the robot on the a curved (cylindrical) surface

## Experimental Results: Spatial Planning




A portrait of Alexander Pushkin drawn by the robot on a curved (cylindrical) surface

## Summary

- Reference motion can be programmed manually using a teach pendant or automatically using some path planning algorithm
- Once a path is generated, its intermediate positions, velocities and accelerations should be interpolated
- Advanced algorithms for spatial movement planning can be designed for industrial applications
- The next step is control design to make the robot to track the reference trajectory

Control design for industrial robots

# Control Design for Industrial Robots

Dr. Oleg Borisov

## PD Controller

Consider the control plant specified by the transfer function. We introduce a proportional-differential (PD) controller with a transfer function

$$R(s) = k_p + k_d s. \tag{1}$$

We calculate the transfer function of a closed-loop system

$$\begin{aligned} W(s) &= \frac{R(s)P(s)}{1+R(s)P(s)} = \frac{\frac{k_p - k_d s}{Js^2+Ks}}{1+\frac{k_p+k_d s}{Js^2+Ks}} = \\ &= \frac{k_p + k_d s}{Js^2 + (K+k_d)s + k_p}. \end{aligned} \tag{2}$$

## PD Controller scheme

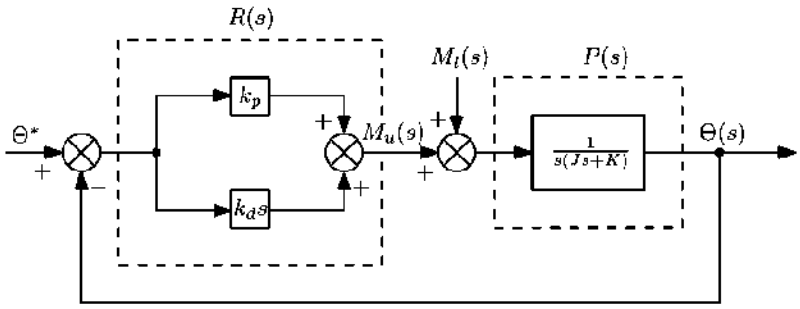



**Figure 1:** Simulation scheme of a closed-loop system with PD controller

Further, with known parameters of the object $J$ & $K$, based on the roots of the characteristic polynomial of the transfer function $Js^2 + (K + k_d)s + k_p$ , it is possible to calculate such coefficients of the PD controller $k_p$ & $k_d$ to ensure the required quality indicators of the closed system.

## PID Controller

Consider the control plant given by the transfer function. We introduce a proportional-integral-differential (PID) controller with a transfer function

$$R(s) = k_p + k_i \frac{1}{s} + k_d s. \tag{3}$$

With structural transformations we express the output variable

$$\Theta(s) - \frac{R(s)P(s)}{1 + R(s)P(s)} \Theta^*(s) + \frac{P(s)}{1 + R(s)P(s)} M_l(s). \tag{4}$$

We calculate the transfer function of a closed-loop system

$$\begin{aligned} W(s) &= \frac{R(s)P(s)}{1 + R(s)P(s)} = \frac{\frac{k_d s^2 + k_p s + k_i}{Js^3 + Ks^2}}{1 + \frac{k_d s^2 + k_p s + k_i}{Js^3 | Ks^2}} = \\ &= \frac{k_d s^2 + k_p s + k_i}{Js^3 + (K + k_d)s^2 + k_p s + k_i}. \end{aligned} \tag{5}$$

## PID Controller scheme

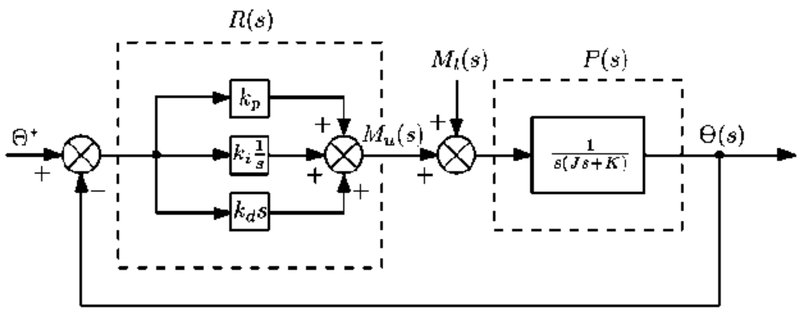



**Figure 2:** Simulation scheme of a closed-loop system with a PID controller

Further, with known parameters of the object $J$ & $K$, based on the roots of the characteristic polynomial of the transfer function $Js^3 + (K + k_d)s^2 + k_p s + k_i$, it is possible to calculate such coefficients of the PID regulator $k_p$, $k_i$ & $k_d$ in order to ensure the required quality indicators of the closed system.

## Robust control

Let us write down the consequtive compensator in the form of a transfer function

$$R(s) = k\gamma_0\sigma^{\rho-1}\frac{\alpha(s)}{\gamma(s)}, \tag{6}$$

where $\rho$ — relative degree of the plant, $k$ & $\sigma > k$ — tuning parameters of the controller, $\alpha(s)$ an arbitrary Hurwitz polynomial of degree $\rho - 1$, $\gamma(s)$ — Hurwitz polynomial of the form

$$\gamma(s) = s^{\rho-1} + \sigma\gamma_{\rho-2}s^{\rho-2} + \cdots + \sigma^{\rho-2}\gamma_1 s + \sigma^{\rho-1}\gamma_0. \tag{7}$$

## Robust control in closed-loop system

Consider the control object specified by the transfer function. Its relative degree is $\rho = 2$, so, chosen $\alpha(s) = s + 1$ & $\gamma_0 = 1$, rewrite the regulator (6) like

$$R(s) = \frac{k\sigma s + k\sigma}{s + \sigma}. \qquad (8)$$

Transfer function of a closed-loop system is

$$\begin{aligned} W(s) &= \frac{R(s)P(s)}{1 + R(s)P(s)} = \frac{\frac{k\sigma s + k\sigma}{(s+\sigma)(Js^2+Ks)}}{1 + \frac{k\sigma s + k\sigma}{(s+\sigma)(Js^2+Ks)}} = \\ &= \frac{k\sigma s + k\sigma}{(s+\sigma)(Js^2 + Ks) + k\sigma s + k\sigma}. \end{aligned} \qquad (9)$$

## Robust control scheme

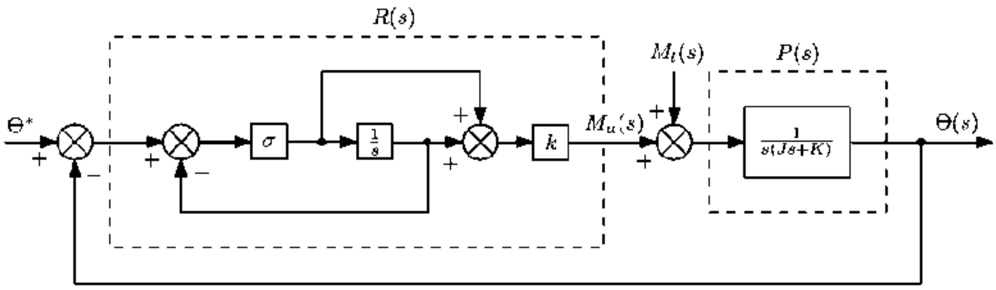



**Figure 3:** Simulation scheme of a closed-loop system with a consecutive compensator

The characteristic polynomial of the transfer function (9) contains unknown parameters of the plat, but due to the robustness of the regulator (8), for sufficiently large coefficients $k$ & $\sigma$ exponential stability is attained.

## Robust control extended

Adding the integral component we rewrite the regulator (6)

$$R(s) = k\gamma_0\sigma^{\rho-1}\frac{\beta(s)}{s\gamma(s)}, \tag{10}$$

where $\beta(s)$ — Hurwitz polynomial of degree $\rho$.

Having chosen $\beta(s) = s^2 + s + 1$ & $\gamma_0 = 1$ rewrite the regulator (10) like

$$R(s) = \frac{k\sigma s^2 + k\sigma s + k\sigma}{s^2 + \sigma s}. \tag{11}$$

Transfer function of a closed-loop system

$$\begin{aligned} W(s) &= \frac{R(s)P(s)}{1 + R(s)P(s)} = \frac{\frac{k\sigma s^2 + k\sigma s + k\sigma}{(s^2+\sigma s)(Js^2+Ks)}}{1 + \frac{k\sigma s^2 + k\sigma s + k\sigma}{(s^2+\sigma s)(Js^2+Ks)}} = \\ &= \frac{k\sigma s^2 + k\sigma s + k\sigma}{(s^2 + \sigma s)(Js^2 + Ks) + k\sigma s^2 + k\sigma s + k\sigma} \end{aligned} \tag{12}$$

## Robust control extended scheme

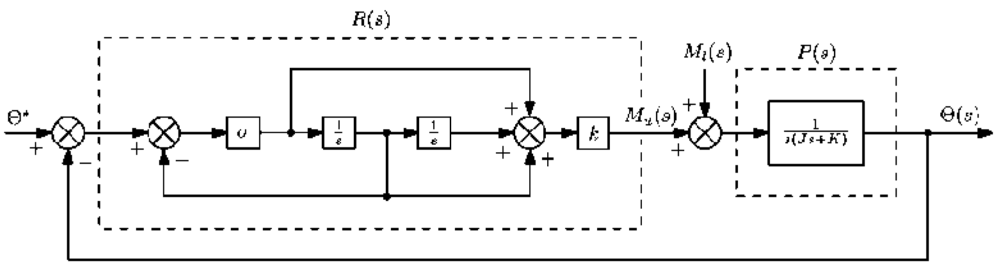



**Figure 4:** Simulation scheme for a closed-loop system with a consecutive compensator with integral loop

The increased order of astaticism of a system with a transfer function (12) makes it possible to compensate the effect of gravitational forces.

## Anti-Windup Control

Saturated input

$$\hat{u}(t) = \operatorname{sat} u(t) = \begin{cases} u_t, & \text{if} \quad u(t) \geq u_t, \\ u(t), & \text{if} \quad u_b < u(t) < u_t, \\ u_b, & \text{if} \quad u(t) \leq u_b, \end{cases} \tag{13}$$

where $u_t$ & $u_b$ — upper and lower limits of the input signal.

Let us write down the control law of the PID controller (3) like

$$u(t) = k_p\tilde{q}(t) + k_i\frac{\tilde{q}(t)}{p} + k_d p\tilde{q}(t), \tag{14}$$

where $p = \frac{d}{dt}$ — differentiation operator, $\tilde{q}(t) = q^* - q(t)$ — error.

Following the antivindap correction method we add to (14) an additional contour

$$u(t) = k_p\tilde{q}(t) + k_i\frac{\tilde{q}(t) + k_u\tilde{u}(t)}{p} + k_d p\tilde{q}(t), \tag{15}$$

where $k_u > 0$ — gain, $\tilde{u}(t) = \hat{u}(t) - u(t)$ — difference signal between saturated and source control.

## Anti-Windup Control scheme

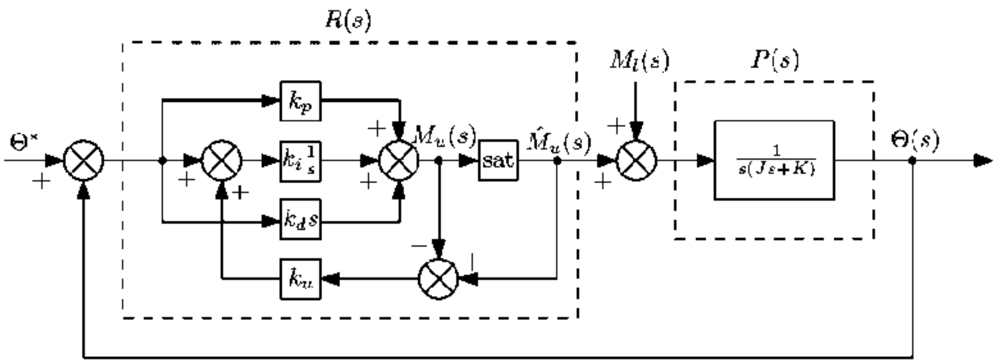



**Figure 5:** Simulation scheme of a closed-loop system with a PID controller and anti-windup correction

The control law (15) helps to avoid the effect of integral saturation in conditions of limited input.

## Anti-Windup Robust Control

Let us write down the control law of a consequtive compensator with an integrated circuit (10) like

$$u(t) = k\frac{\beta(p)}{p}\hat{\tilde{q}}(t), \quad (16)$$

$$\dot{\xi}(t) = \sigma(\Gamma\xi(t) + d\gamma_0\tilde{q}(t)), \quad (17)$$

$$\hat{\tilde{q}}(t) = h^T\xi(t), \quad (18)$$

where $\hat{\tilde{q}}(t)$ — error signal estimation $\tilde{q}(t)$, matrices and vectors $\Gamma$, $d$, $h$ in form

$$\Gamma = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -\gamma_0 & -\gamma_1 & -\gamma_2 & \dots & -\gamma_{p-1} \end{bmatrix},\ d = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix},\ h = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \quad (19)$$

## Anti-Windup Robust Control

Transform the control law (16), with integrator

$$u(t) = k\frac{\beta(p)}{p}\hat{\tilde{q}}(t) = k\left(\bar{\beta}(p) + \frac{\beta_0}{p}\right)\hat{\tilde{q}}(t) = k\bar{\beta}(p)\hat{\tilde{q}}(t) + k\frac{\beta_0}{p}\hat{\tilde{q}}(t), \quad (20)$$

where $\bar{\beta}(p) = \frac{\beta(p)-\beta_0}{p}$.

Following the anti-windup correction method we add to the (20) an additional contour

$$u(t) = k\bar{\beta}(p)\hat{\tilde{q}}(t) + k\frac{\beta_0}{p}\left(\hat{\tilde{q}}(t) + k_u\tilde{u}(t)\right), \quad (21)$$

where $k_u > 0$ — gain, $\tilde{u}(t) = \hat{u}(t) - u(t)$ — difference signal between saturated and source control.

Having chosen $\beta(p) = p^2 + p + 1$ & $\gamma_0 = 1$, rewrite the regulator (21) like

$$u(t) = kp\hat{\tilde{q}}(t) + k\hat{\tilde{q}}(t) + k\frac{1}{p}(\hat{\tilde{q}}(t) + k_u\tilde{u}(t)). \quad (22)$$

## Anti-Windup Robust Control

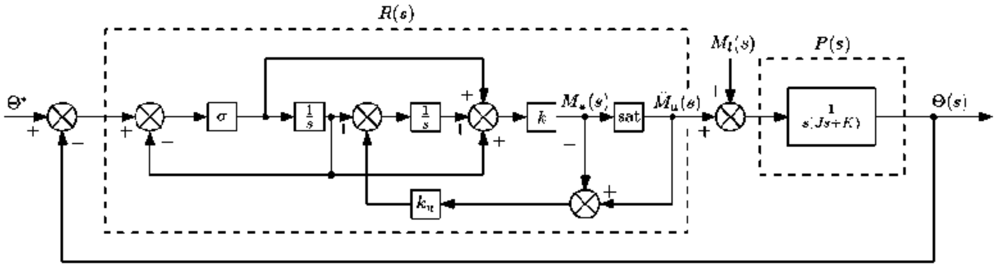


**Figure 6:** Simulation scheme of a closed-loop system with a consecutive compensator and anti-windup correction

The regulator (22) allows to solve the stabilization problem with the increased order of astaticism in comparison with the regulator (6) and with compensation of the integral saturation effect by means of anti-windup.

## Tracking control

Let's express the output signal $\Theta(s)$:

$$\Theta(s) = \frac{R(s)P(s) + F(s)P(s)}{1 + R(s)P(s)}\Theta^*(s) + \frac{P(s)}{1 + R(s)P(s)}M_l(s). \quad (23)$$

We choose the transfer function of direct coupling in the form:

$$F(s) = \frac{1}{P(s)}, \quad (24)$$

then the expression (23) takes the form:

$$\Theta(s) = \Theta^*(s) + + \frac{P(s)}{1 + R(s)P(s)}M_l(s). \quad (25)$$

## Tracking control scheme

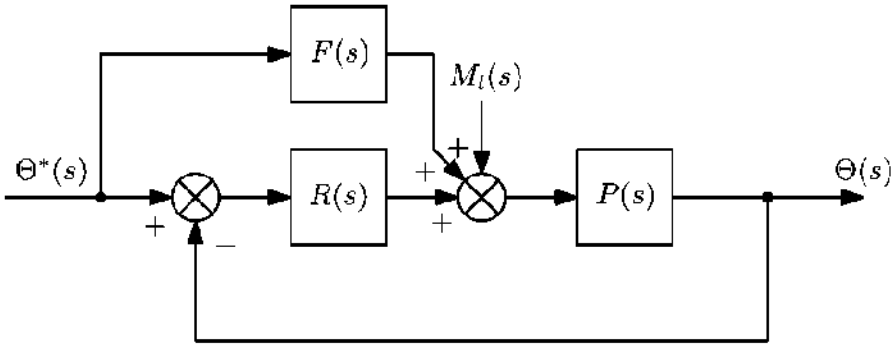



**Figure 7:** Simulation scheme for closed-loop tracking system

Direct link allows the system to monitor any given trajectory, provided that the system is completely stable. The steady-state error in this case will be due only to the influence of an external perturbation $M_l(s)$.

## Multivariable control

Consider the dynamic model of a robotic system

$$\Gamma(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = u. \tag{26}$$

Stabilization of desired $q^*$ will be performed with PD controller.

First, for simplicity, we neglect the effect of gravity, assuming that $G(q) = 0$. In view of this model (26) looks like

$$\Gamma(q)\ddot{q} + C(q,\dot{q})\dot{q} = u. \tag{27}$$

We choose the vector of control actions $u$ like

$$u = K_p(q^* - q(t)) - K_d\dot{q}(t) = K_p\tilde{q}(t) + K_d\dot{\tilde{q}}(t), \tag{28}$$

where $\tilde{q}(t) = q^* - q(t)$ — error between the specified configuration and the current one, $K_p$ & $K_d$ looks like

$$K_p = \begin{bmatrix} k_{p,1} & 0 & \dots & 0 \\ 0 & k_{p,2} & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & k_{p,n} \end{bmatrix}, \quad K_d = \begin{bmatrix} k_{d,1} & 0 & \dots & 0 \\ 0 & k_{d,2} & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & k_{d,n} \end{bmatrix}. \tag{29}$$

## Multivariable control

Substituting the control law (28) to the plant (27) we obtain a model of a closed system

$$\Gamma(q)\ddot{q} + C(q,\dot{q})\dot{q} = K_p\tilde{q}(t) + K_d\dot{\tilde{q}}(t). \tag{30}$$

To analyze the stability of a closed-loop system (30) we consider the candidate Lyapunov function in quadratic form

$$V(t) = \frac{1}{2}\tilde{q}^T K_p\tilde{q} + \frac{1}{2}\dot{\tilde{q}}^T \Gamma\dot{\tilde{q}}. \tag{31}$$

Taking the time derivative of (31) we get

$$\dot{V}(t) = -\dot{\tilde{q}}^T K_d\dot{\tilde{q}} \leq 0, \tag{32}$$

this together with the Lassalle theorem shows the asymptotic stability of a closed system (30).

## Multivariable control

When $\dot{V} = 0$ from (32) we can conclude that the generalized velocities and accelerations are zero $\dot{q}(t) = 0$ & $\ddot{q}(t) = 0$. Taking this into account we rewrite the equation of a closed system for $t \to \infty$

$$0 = K_p\tilde{q}(t), \tag{33}$$

from which it follows that $\tilde{q}(t) = q^* - q(t) = 0$ with $t \to \infty$.

The influence of gravity $G(q) \neq 0$ leads to the appearance of a steady error. The PD controller in this case does not provide asymptotic stability. The equation (33) looks like

$$G(q) = K_p\tilde{q}(t). \tag{34}$$

To eliminate the established error we supplement the law of control

$$u = K_p\tilde{q}(t) + K_d\dot{\tilde{q}}(t) + G(q), \tag{35}$$

which makes it possible to provide asymptotic stability with the influence of gravity.

Dynamic of robotic systems

**Dynamics of Robotic Systems**
**Euler-Lagrange Method and Special Cases**

**Sergey Kolyubin**

**Outline**

- Motivation
- Energy-based Approach - Euler-Lagrange Method
  - Energy calculation
  - Motion equation
- Special Cases
  - Drive dynamics
  - Flexible joints modeling
- Motion Equation in Operational Space

## Why Do We Need to Know Dynamics?

- simulation
- defining dynamic constraints
- mechanical design optimization
- trajectory planners and controllers synthesis

## Tasks

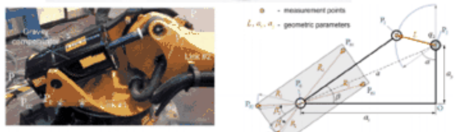
- Forward Dynamics: given desired trajectory (coordinates, velocities, acceleration) find generalized forces/torques
- Inverse Dynamics: given generalized forces/torques, find generated motion (trajectory)

## Tasks

- Forward Dynamics: given desired trajectory (coordinates, velocities, acceleration) find generalized forces/torques
- Inverse Dynamics: given generalized forces/torques, find generated motion (trajectory)

**Practical tasks**

- f/t calculation - find external (control design) and internal (find reaction forces in kinematic pairs)
- performance indicators - find possible cycle time given dynamic constraints
- (serial) manipulators balancing - unload drives in statics
- (parallel) manipulators dynamic balancing - minimize distortions during the motion by placing counter-weights



(a) gravity compensator of KUKA KR-270 TM robot

(b) Geometrical parameters of the gravity compensator and measurement point locations

## Tasks

- Forward Dynamics: given desired trajectory (coordinates, velocities, acceleration) find generalized forces/torques
- Inverse Dynamics: given generalized forces/torques, find generated motion (trajectory)

**Theoretical sub-tasks**

- trajectories calculation
- motion stability analysis
- calculating time response
- identifying critical motion modes

## Methods Comparison

- E-L - kinetic and potential energy
  - multibody dynamics as a whole
  - exclude reaction forces between links
  - symbolic form
  - better for analysis
- N-E - forces/torques balance
  - separate equation for each body
  - explicit relations for reaction forces
  - numeric recursion form
  - better for synthesis and real-time applications

By excluding reaction forces and substituting these relations we can derive E-L equations from N-E equations

## E-L General Framework

1. select generalized coordinates $q_1, q_2, \dots q_n$
2. derive relations for kinetic $\mathcal{K}$ and potential $\mathcal{P}$ energy as functions of generalized coordinates and its derivatives
3. calculate system Lagrangian $\mathcal{L}$
4. derive motion equation

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_k} - \frac{\partial L}{\partial q_k} = \tau_k, \; k = 1, 2, \dots n \tag{1}$$

$\tau_k$ is a generalized force/torque

## Full Kinetic Energy

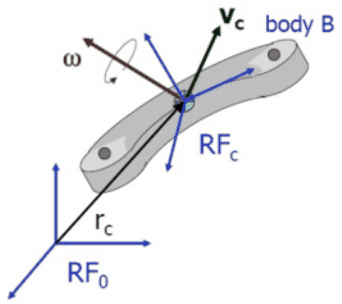



**Figure 1:** ©DeLuca

**Konig theorem**
Full energy consist of an energy assoc. with body CoM motion and relative body motion around it CoM

$$\mathcal{K} = \frac{1}{2}m|v|^2 + \frac{1}{2}\omega^T \mathcal{I} \omega$$

where $m$ is a body mass, $v$ and $\omega$ are linear and rotational velocities vectors, $\mathcal{I}$ is an inertia tensor

**All values in the same CF**

Formula for rotational velocity

$$\omega \leftarrow S(\omega) = \dot{R}(t)R^T(t),$$

where $R$ is a rotation matrix from body frame to inertial frame

## Kinetic Energy of $n$-links Robot

Sum of kinetic energy of linear and rotational motions

$$\mathcal{K} = \frac{1}{2}m\left|v_c\right|^2 + \frac{1}{2}\omega^T \mathcal{I} \omega$$

CoM velocities

- $v_c = \dot{r}_c$ and $\omega$ are functions of generalized coordinates $q$ and velocities $\dot{q}$

## Kinetic Energy of $n$-links Robot

Sum of kinetic energy of linear and rotational motions

$$\mathcal{K} = \frac{1}{2} m |v_c|^2 + \frac{1}{2} \omega^T \mathcal{I} \omega$$

CoM velocities

- $v_c = \dot{r}_c$ and $\omega$ are functions of generalized coordinates $q$ and velocities $\dot{q}$

---

Relations can be computed via Jacobian assoc. with links CoMs

$$v_{c,i} = J_{v_i}(q)\dot{q}, \quad \omega_i = J_{\omega_i}(q)\dot{q}$$

## Kinetic Energy of $n$-links Robot

Sum of kinetic energy of linear and rotational motions

$$\mathcal{K} = \frac{1}{2} m |v_c|^2 + \frac{1}{2} \omega^T \mathcal{I} \omega$$

CoM velocities

- $v_c = \dot{r}_c$ and $\omega$ are functions of generalized coordinates $q$ and velocities $\dot{q}$

---

Relations can be computed via Jacobian assoc. with links CoMs

$$v_{c,i} = J_{v_i}(q)\dot{q}, \quad \omega_i = J_{\omega_i}(q)\dot{q}$$

---

Robot kinetic energy

$$\mathcal{K} = \frac{1}{2} \dot{q}^T \left[ \sum_{i=1}^{n} m_i J_{v_i}(q)^T J_{v_i}(q) + J_{\omega_i}(q)^T R_i(q) I R_i(q)^T J_{\omega_i}(q) \right] \dot{q}$$

## Recurrent Velocities Formulas

- rotation (angular) velocities

$$\omega_i = \left(R_i^{i-1}(q_i)\right)^T [\omega_{i-1} + (1-\sigma_i)\dot{q}_i z_{i-1}] = \left(R_i^{i-1}(q_i)\right)^T \omega_i^{i-1}, \quad (2)$$

where $R_i^{i-1}(q_i)$ is a rotation matrix for neighbor CFs $O_{i-1}x_{i-1}y_{i-1}z_{i-1}$ and $O_i x_i y_i z_i$,

$$\sigma_i = \begin{cases} 0, \textit{for rotational joint}, \\ 1, \textit{for prismatic joint}, \end{cases}$$

$z_{i-1} = [0\,0\,1]^T$ is a vector of z axis coord. if D-H convention used, $\omega_i^{i-1}$ is a rotational velocity of $i$-th link with respect to CF $O_{i-1}x_{i-1}y_{i-1}z_{i-1}$

- linear velocities

$$v_{c,i} = v_i + \omega_i \times r_{c,i}, \quad (3)$$

where

$$v_i = \left(R_i^{i-1}(q_i)\right)^T \left[v_{i-1} + \sigma_i \dot{q}_i z_{i-1} + \omega_i^{i-1} \times r_{i-1,i}^{i-1}\right] \quad (4)$$

denotes linear velocity of a CF origin $O_i$, $r_{c,i}$ is a CoM vector for $i$-th link with respect to $O_i$, $r_{i-1,i}^{i-1}$ are coordinates of radius-vectors from $O_{i-1}$ to $O_i$ with respect to CF $O_{i-1}x_{i-1}y_{i-1}z_{i-1}$.

## Potential Energy of $n$-links Robot

Potential energy of $i$-th link

$$\mathcal{P}_i = m_i g^T r_{c,i}$$

where $r_{c,i}$ is a CoM coordinates vector

$$\begin{vmatrix} r_{c,i} \\ 1 \end{vmatrix} = {}^0H_1(q_1)\,{}^1H_2(q_2)\cdots{}^{i-1}H_i(q_i) \begin{vmatrix} {}^i r_{c,i} \\ 1 \end{vmatrix} \quad (5)$$

Robot potential energy

$$\mathcal{P} = \sum_{i=1}^{n} \mathcal{P}_i = \sum_{i=1}^{n} m_i g^T r_{c,i}$$

For a serial kinematic chain

$$\mathcal{P} = \sum_{i=1}^{n} \mathcal{P}_i$$

$$\mathcal{P}_i = \mathcal{P}_i(q_j, j \le i)$$

## Motion Equation

- Kinetic energy

$$\begin{aligned} \mathcal{K} &= \frac{1}{2}\dot{q}^T \left[\sum_{i=1}^{n} m_i J_{v_i}(q)^T J_{v_i}(q) + J_{\omega_i}(q)^T R_i(q) I R_i(q)^T J_{\omega_i}(q)\right] \dot{q} \\ &= \frac{1}{2}\dot{q}^T M(q)\dot{q} = \frac{1}{2}\sum_{k,j} m_{kj}(q)\dot{q}_k\dot{q}_j \end{aligned}$$

- for conservative generalized forces $\psi_k = -\frac{\partial \mathcal{P}}{\partial q_k} + \tau_k$
- system Lagrangian $\mathcal{L} = \mathcal{K} - \mathcal{P}$

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{q}_k} - \frac{\partial \mathcal{L}}{\partial q_k} = \tau_k$$

$$\frac{d}{dt}\frac{\partial(\mathcal{K} - \mathcal{P})}{\partial \dot{q}_k} - \frac{\partial(\mathcal{K} - \mathcal{P})}{\partial q_k} = \tau_k$$

$$\frac{d}{dt}\frac{\partial \mathcal{K}}{\partial \dot{q}_k} - \frac{\partial(\mathcal{K} - \mathcal{P})}{\partial q_k} = \tau_k$$

## Motion Equation (contd.)

Equation structure

$$\frac{d}{dt}\frac{\partial \mathcal{K}}{\partial \dot{q}_k} - \frac{\partial(\mathcal{K} - \mathcal{P})}{\partial q_k} = \tau_k, \qquad k = 1, \ldots, n$$

---

1st term

$$\frac{\partial \mathcal{K}}{\partial \dot{q}_k} = \frac{\partial}{\partial \dot{q}_k}\left[\frac{1}{2}\dot{q}^T M(q)\dot{q}\right] = \sum_{j=1}^{n} m_{kj}\dot{q}_j$$

and

$$\begin{aligned} \frac{d}{dt}\frac{\partial \mathcal{K}}{\partial \dot{q}_k} = \frac{d}{dt}\left[\sum_{j=1}^{n} m_{kj}\dot{q}_j\right] &= \sum_{j=1}^{n} m_{kj}\ddot{q}_j + \sum_{j=1}^{n} \tfrac{d}{dt}\left[m_{kj}(q)\right]\dot{q}_j \\ &= \sum_{j=1}^{n} m_{kj}\ddot{q}_j + \frac{1}{2}\sum_{j=1}^{n}\sum_{i=1}^{n}\left(\frac{\partial m_{kj}}{\partial q_i} + \frac{\partial m_{ki}}{\partial q_j}\right)\dot{q}_i\dot{q}_j \end{aligned}$$

## Motion Equation (contd.)

2nd term

$$\frac{\partial(\mathcal{K}-\mathcal{P})}{\partial q_k} = \frac{\partial}{\partial q_k}\left[\frac{1}{2}\dot{q}M(q)\dot{q}-\mathcal{P}\right] = \frac{1}{2}\dot{q}\left[\frac{\partial}{\partial q_k}M(q)\right]\dot{q}-\frac{\partial}{\partial q_k}\mathcal{P}$$
$$= \frac{1}{2}\sum_{j=1}^{n}\sum_{i=1}^{n}\frac{\partial m_{ij}}{\partial q_k}\dot{q}_i\dot{q}_j-\frac{\partial}{\partial q_k}\mathcal{P}$$

## Motion Equation (contd.)

2nd term

$$\frac{\partial(\mathcal{K}-\mathcal{P})}{\partial q_k} = \frac{\partial}{\partial q_k}\left[\frac{1}{2}\dot{q}M(q)\dot{q}-\mathcal{P}\right] = \frac{1}{2}\dot{q}\left[\frac{\partial}{\partial q_k}M(q)\right]\dot{q}-\frac{\partial}{\partial q_k}\mathcal{P}$$
$$= \frac{1}{2}\sum_{j=1}^{n}\sum_{i=1}^{n}\frac{\partial m_{ij}}{\partial q_k}\dot{q}_i\dot{q}_j-\frac{\partial}{\partial q_k}\mathcal{P}$$

---

Resulting relations

$$\sum_{j=1}^{n}m_{kj}\ddot{q}_j+\frac{1}{2}\sum_{j=1}^{n}\sum_{i=1}^{n}\left(\frac{\partial m_{kj}}{\partial q_i}+\frac{\partial m_{ki}}{\partial q_j}\right)\dot{q}_i\dot{q}_j$$
$$-\frac{1}{2}\sum_{j=1}^{n}\sum_{i=1}^{n}\frac{\partial m_{ij}}{\partial q_k}\dot{q}_i\dot{q}_j+\frac{\partial}{\partial q_k}\mathcal{P}=\tau_k$$

## Motion Equation (contd.)

2nd term

$$\frac{\partial(\mathcal{K}-\mathcal{P})}{\partial q_k} = \frac{\partial}{\partial q_k}\left[\frac{1}{2}\dot{q}M(q)\dot{q}-\mathcal{P}\right] = \frac{1}{2}\dot{q}\left[\frac{\partial}{\partial q_k}M(q)\right]\dot{q}-\frac{\partial}{\partial q_k}\mathcal{P}$$
$$= \frac{1}{2}\sum_{j=1}^{n}\sum_{i=1}^{n}\frac{\partial m_{ij}}{\partial q_k}\dot{q}_i\dot{q}_j-\frac{\partial}{\partial q_k}\mathcal{P}$$

Resulting relations

$$\sum_{j=1}^{n} m_{kj}(q)\ddot{q}_j+\sum_{j=1}^{n}\sum_{i=1}^{n} c_{ijk}(q)\dot{q}_i\dot{q}_j+g_k(q)=\tau_k\,, \quad k=1,\dots,n$$

where $c_{ijk}=c_{jik}$ is a Christoffel symbol and

$$c_{ijk}(q)=\frac{1}{2}\left(\frac{\partial m_{kj}}{\partial q_i}+\frac{\partial m_{ki}}{\partial q_j}-\frac{\partial m_{ij}}{\partial q_k}\right), \quad g_k(q)=\frac{\partial}{\partial q_k}\mathcal{P}$$

is a potential energy gradient

## Motion Equation (contd.)

2nd term

$$\frac{\partial(\mathcal{K}-\mathcal{P})}{\partial q_k} = \frac{\partial}{\partial q_k}\left[\frac{1}{2}\dot{q}M(q)\dot{q}-\mathcal{P}\right] = \frac{1}{2}\dot{q}\left[\frac{\partial}{\partial q_k}M(q)\right]\dot{q}-\frac{\partial}{\partial q_k}\mathcal{P}$$
$$= \frac{1}{2}\sum_{j=1}^{n}\sum_{i=1}^{n}\frac{\partial m_{ij}}{\partial q_k}\dot{q}_i\dot{q}_j-\frac{\partial}{\partial q_k}\mathcal{P}$$

Resulting relations

$$\sum_{j=1}^{n} m_{kj}(q)\ddot{q}_j+\sum_{j=1}^{n}\sum_{i=1}^{n} c_{ijk}(q)\dot{q}_i\dot{q}_j+g_k(q)=\tau_k\,, \quad k=1,\dots,n$$

in a vectorial form

$$M(q)\ddot{q}+C(q,\dot{q})\dot{q}+G(q)=\tau$$

with Coriolis and centrifugal forces $C(q)$, $c_{kj}=\sum_{i=1}^{n} c_{ijk}(q)\dot{q}_i$.

## Accounting for Gear and Motor Dynamics

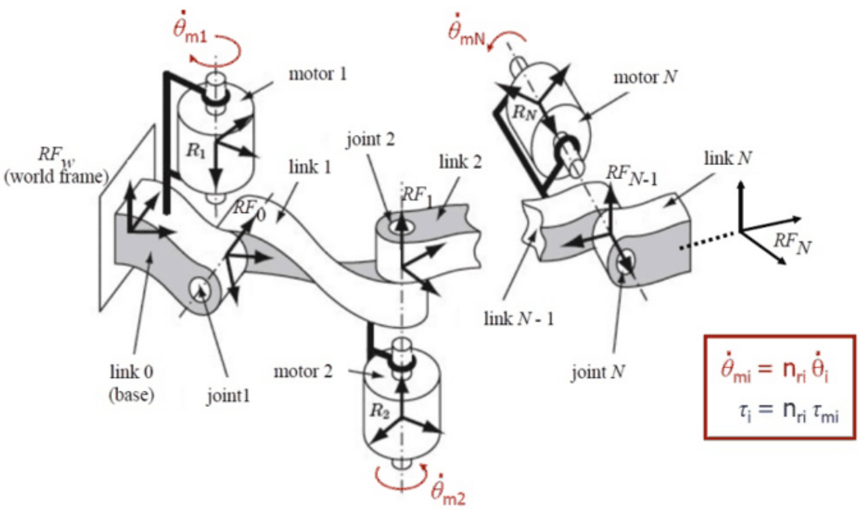
**Assumptions**

- drive is fixed to the link preceding the link it is moving
- motor and joint axis are coinciding

**General considerations**

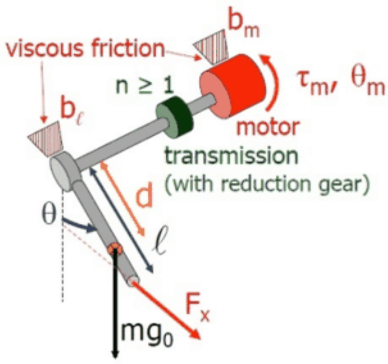
- drive mass should be added to link mass
- drive rotor inertia should be taking into account when computing total kinetic energy
- gear ration should be taken into account when computing velocities and forces

## Motor Placement




©DeLuca

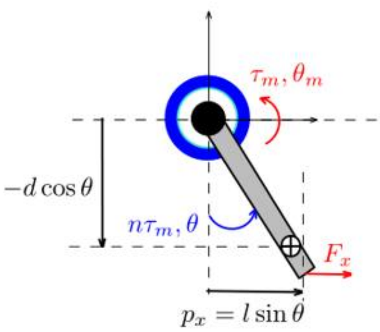
## Pendulum with Gear




©DeLuca

$I_l$ – link moment of inertia w.r.t. its CoM
$m$ – link mass
$d$ – distance from axis of rotation to link CoM
$\dot{\theta}$ – link rotation velocity (after gear)
$\dot{\theta}_m = n\dot{\theta}$ – motor rotation velocity (before gear)
$n$ – gear ratio
$I_m$ – drive moment of inertia w.r.t its axis of rotation

## Kinetic Energy




**Pendulum kinetic energy**

$$\mathcal{K}_l = \frac{1}{2}\left(I_l + md^2\right)\dot{\theta}^2,$$

**Drive kinetic energy**

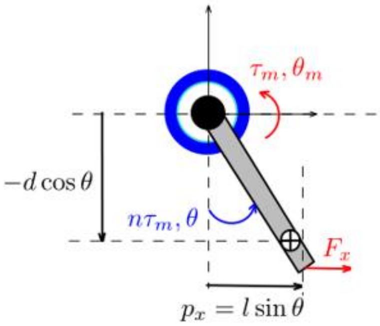
$$\mathcal{K}_m = \frac{1}{2}I_m\dot{\theta}_m^2,$$

**Total kinetic energy**

$$\mathcal{K} = \mathcal{K}_l + \mathcal{K}_m = \frac{1}{2}I\dot{\theta}^2,$$

where $I = I_l + md^2 + n^2 I_m$ is a total moment of inertia w.r.t. axis of rotation

## Potential Energy and Lagrangian




**Total potential energy**

$$\mathcal{P} = \mathcal{P}_0 - mg_0 d\cos\theta.$$

**System Lagrangian**

$$\mathcal{L} = \frac{1}{2}I\dot{\theta}^2 + mg_0 d\cos\theta - \mathcal{P}_0.$$

## Motion Equation

**From the link side**

$$I\ddot{\theta} + mg_0 d\sin\theta = \tau.$$

**From the motor side**

$$\frac{I}{n^2}\ddot{\theta}_m + \frac{m}{n}g_0 d\sin\frac{\theta_m}{n} = \tau_m - \left(\frac{k_{fl}}{n^2} + k_{fm}\right)\dot{\theta}_m + \frac{l}{n}\cos\frac{\theta_m}{n}F_x.$$

## Friction Forces

**General considerations**

- is a dissipative force
- localized in joints
- static model captures major influence for relatively fast motion

$$\tau = n\tau_m - k_{fl}\dot{\theta} - nk_{fm}\dot{\theta}_m + \dot{p}_x F_x = n\tau_m - (k_{fl} + n^2 k_{fm})\dot{\theta} + l\cos\theta F_x,$$

where $\tau_m$ is drive torque before gear, $k_{fm}$ and $k_{fl}$ are viscous friction coefficients

- dynamic models are more accurate, but usually hard to identify

## Flexible-Joints Robots

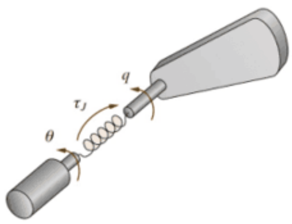



**Figure 2:** Flexible joint sketch

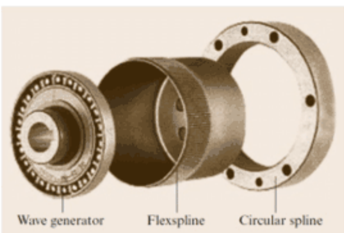



**Figure 3:** Harmonic drive

**Flexible joints**
Motor (input) and link (output) are connected by a flexible (deformable) element

- long shaft
- harmonic drive gearbox
- belts

Useful flexibility

1. physically (VSA, SEA)
2. on a software level

for

- safe pHRI
- explosive motions

## Modeling Flexible Joints

**Assumptions**

1. flexibility is localized in joint
2. small deformations for linear spring model
3. symmetric drive shafts with CoM on the axis of rotation
4. drive is located before the link it is actuating

## Modeling Flexible Joints

- introduce $2n$ generalized coordinates $q \in R^n$ for links and $\theta \in R^n$ for drives ($\theta_i = \theta_{mi}/r_i$, $r_i$ is a gear ratio)
- add drive kinetic energy

$$\mathcal{K}_{mi} = \frac{1}{2}\mathcal{I}_m\dot{\theta}_{mi}^2 = \frac{1}{2}\mathcal{I}_m r_i^2 \dot{\theta}_i^2$$

$$\mathcal{K}_m = \sum_{i=1}^{n} \mathcal{K}_{mi} = \frac{1}{2}\dot{\theta}^T M_m \dot{\theta}$$

$M_m$ is a diagonal drive inertia matrix

- add potential energy of a deformed spring

$$\mathcal{P}_{ei} = \frac{1}{2}K_i(q_i - \theta_i)^2$$

$$\mathcal{P}_e = \sum_{i=1}^{n} \mathcal{P}_{ei} = \frac{1}{2}(q-\theta)^T K(q-\theta)$$

$K$ is a matrix of joint stiffness coefficients

## Modeling Flexible Joints

**Motion equation**

$$M(q)\ddot{q} - c(q,\dot{q}) + G(q) + K(q-\theta) - 0,$$
$$M_m\ddot{\theta} + K(\theta - q) = \tau$$

## Operational Space Formulation

- Configuration space

$$M(q)\ddot{q} + c(q,\dot{q}) + g(q) = \tau$$

- Operational space

$$\Lambda\ddot{x}_e + \mu + \rho = F_e$$

## Operational Space Formulation

- Configuration space

$$M(q)\ddot{q} + c(q,\dot{q}) + g(q) = \tau$$

- Operational space

$$\Lambda \ddot{x}_e + \mu + \rho = F_e$$

- projecting joint forces/torques to end-effector forces

$$\tau = J_e^T F_e$$

- kinematic relations

$$\dot{x}_e = J_e \dot{q} \Rightarrow \ddot{x}_e = J_e \ddot{q}_e + \dot{J}_e \dot{q}_e$$

$$\ddot{x}_e = J_e M^{-1} \left( J_e^T F_e - (c(q,\dot{q}) + g(q)) \right) + \dot{J}_e \dot{q}_e \Rightarrow$$

$$\ddot{x}_e + J_e M^{-1}(c(q,\dot{q}) + g(q)) - \dot{J}_e \dot{q}_e = J_e M^{-1} J_e^T F_e$$

- operational-space model

$$\Lambda = \left( J_e M^{-1} J_e^T \right)^{-1} \qquad \mu = \Lambda J_e M^{-1} c(q,\dot{q}) - \Lambda \dot{J}_e \dot{q}_e \qquad \rho = \Lambda J_e M^{-1} g(q)$$

Trajectory control algorithms
Trajectory control algorithms based on stabilization of sets

# Trajectory control algorithms based on stabilisation of sets

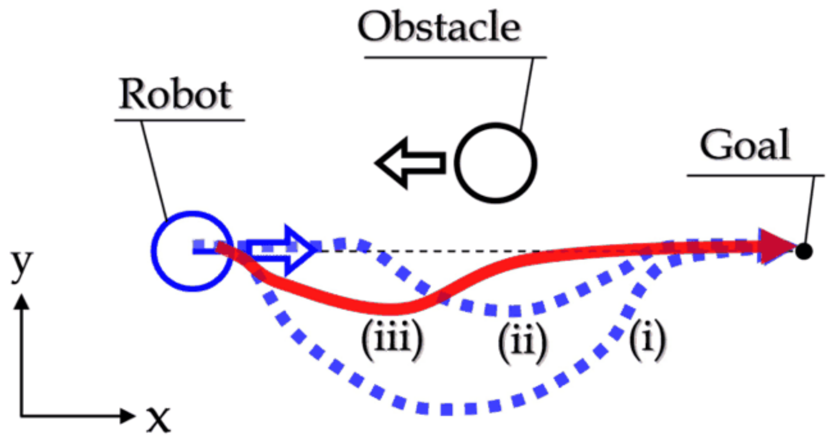
Aleksandr Y. Krasnov

## Examples of problems

Autonomous vehicles control

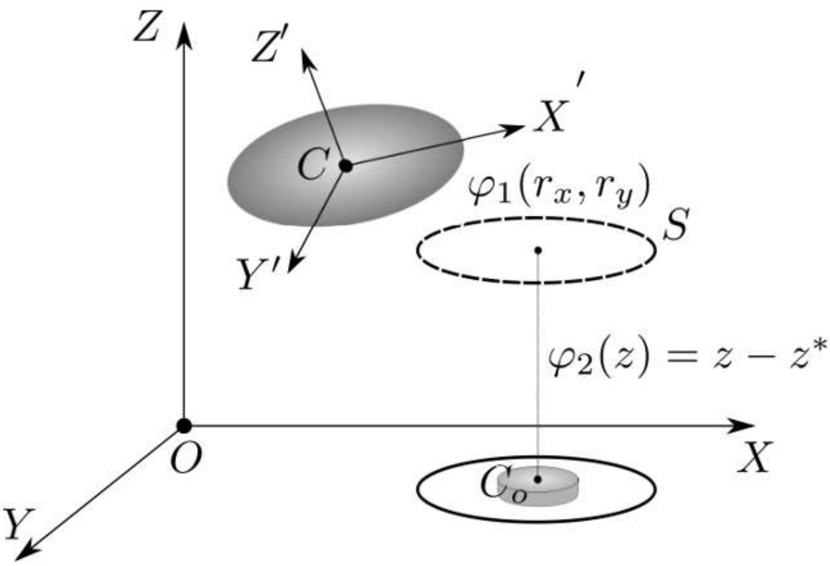
**Examples of problems**

Collision avoidance:



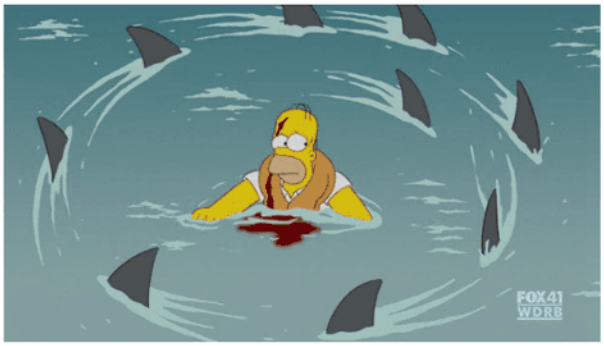

**Examples of problems**

Following the moving target:



## Examples of problems

# Control of underwater biomimetic robots (Robotic Fish)



Problem: robot can not stop after reaching a goal.

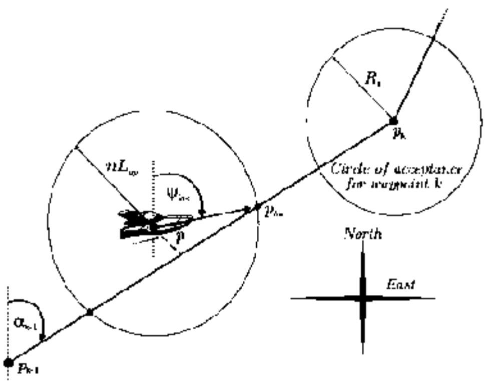
Possible solution: continue motion along closed curve around the the goal

## Possible solutions: Tracking approaches

- Virtual Target Tracking:
  - Backstepping based

    *Aguiar, A.P.; Hespanha, J.P., Kokotovic, P.V., "Path-following for nonminimum phase systems removes performance limitations," Automatic Control, IEEE Transactions on , vol.50, no.2, pp.234,259. Feb. 2005*
  - LOS(Line-of-Sight) methods

    *M. Breivik and T.I. Fossen Principles of Guidance Based Path Following in 2D and 3D Proceedings of the IEEE Conference on Decision and Control, Seville, Spain,2005,pp. 627-634*



## Possible solutions: Set stabilization approaches

- Sliding mode

  *Ashrafiuon, H., Muske, K. R., McNinch, L. C., and Soltan, R., "Sliding Model Tracking Control of Surface Vessels," IEEE Transactions on Industrial Electronics SS on Sliding Mode Control in Industrial Applications, 2008*

- Passification

  *M. El-Hawwary, M. Maggiore, Case Studies on Passivity-Based Stabilization of Closed Sets, International Journal of Control, 2011*

- Feedback linearization:
  - Methods of transversal linearization

    *Nielsen, C.; Fulford, C.; Maggiore, M., "Path following using transverse feedback linearization: Application to a maglev positioning system," American Control Conference, 2009*

  - Vector Field Path Following

    *Nelson, D.R.; Barber, D.B.; McLain, T.W.; Beard, R.W., "Vector field path following for small unmanned air vehicles," American Control Conference, 2005*

  - Coordination control by Iliya V. Miroshnik

## Description of the desired trajectory

Three ways to describe straight line:

- Explicit description

$$y = \frac{ax - c}{b}$$

- Implicit description

$$ax + by + c = 0$$

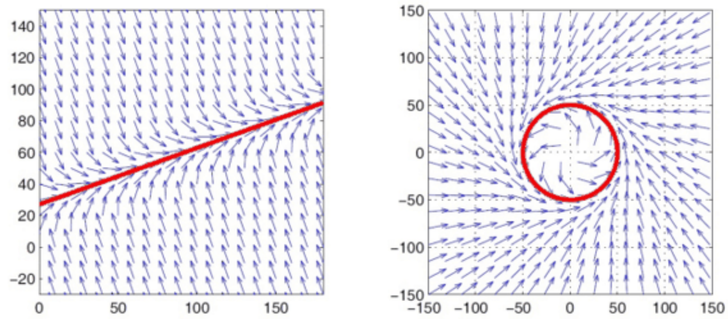
- Parametric description

$$x = x_0 + ft$$

$$y = y_0 + gt$$

## Main ideas of methods based on the stabilization of sets

- Implicit representation of curve
- Dependence on the current position in the space
- Invariance of desired path(an attractor in the output space)
- Potentially higher accuracy of motion




*Sujit, P.B.; Saripalli, S.; Sousa, J.B., "An evaluation of UAV path following algorithms," Control Conference (ECC), 2013*

*Nelson, D.R.; Barber, D.B.; McLain, T.W.; Beard, R.W., "Vector field path following for small unmanned air*

## Formal statement of control problem

- Geometric sub-task:

$$dist\left(p - f(p_d)\right) \to 0,$$

where $f(p_d)$ - a desired path of motion, $p_d$ - spatial coordinates of space, $p$ - current position of a plant.

- Kinematic sub-task - maintenance of desired velocity of motion along the path:

$$\lim_{t\to\infty} \Delta V = \lim_{t\to\infty} (V - V^*) \to 0,$$

where $V$ - current velocity of motion, $V^*$ - desired velocity.

## Motion on the plane

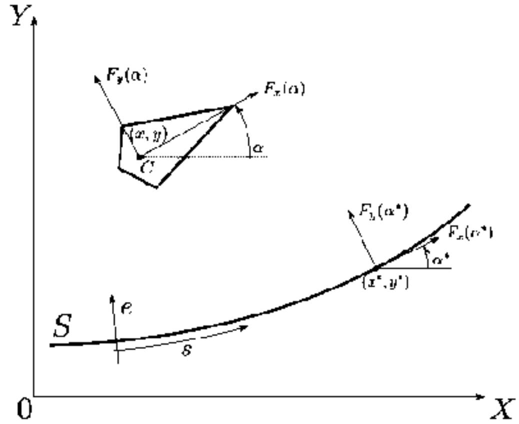
Dynamic model of robot motion:

$$\dot{v}_x = v_y\omega + \frac{1}{m}F_x, \quad (1)$$

$$\dot{v}_y = -v_x\omega + \frac{1}{m}F_y, \quad (2)$$

$$\dot{\omega} = \frac{1}{J}M_c, \quad (3)$$

where $v_x$ and $v_y$ are linear velocities,
$F_x$ and $F_y$ are control forces,
$\omega$ is the angular velocity,
$m$ is the mass of the plant,
$J$ is the moment of inertia,
$M_c$ is the control torque.




## Motion on the plane

Relation of linear velocities in the fixed and absolute frames:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = T^T(\alpha) \begin{bmatrix} v_x \\ v_y \end{bmatrix}, \quad (4)$$

where $T^T(\alpha) = \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix}$ is the rotational matrix of $C$-fixed frame.

Linear accelerations in absolute frame

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = \frac{1}{m}T^T(\alpha) \begin{bmatrix} F_x \\ F_y \end{bmatrix}. \quad (5)$$

# Motion on the plane

The desired path is an implicitly described smooth segment of curve $S$:

$$\varphi(x,y)=0, \tag{6}$$

and relevant local coordinate $s$ (path length) is defined as

$$s=\psi(x,y) \tag{7}$$

Selection of functions (6) and (7) is mostly limited by regularity condition implying that Jacobian matrix

$$\Upsilon(x,y)=\begin{bmatrix}\frac{\partial\psi}{\partial x} & \frac{\partial\psi}{\partial y}\\ \frac{\partial\varphi}{\partial x} & \frac{\partial\varphi}{\partial y}\end{bmatrix} \tag{8}$$

is not degenerate for any $(x,y)$ belonging to curve $S$, i.e. $det\Upsilon(x,y)\neq 0$

For regular geometrical objects there exists normalized description with orthogonal Jacobian matrix:

$$\Upsilon(x,y)=T(\alpha^*(s))=\begin{bmatrix}\cos\alpha^*(s) & \sin\alpha^*(s)\\ -\sin\alpha^*(s) & \cos\alpha^*(s)\end{bmatrix}\in SO(2)$$

where $T(\alpha^*(s))$ is the rotational matrix of moving Frenet frame, $\alpha^*(s)$ is $s$-dependent target angle determining the current orientation of Frenet frame.
Frenet matrix satisfies to the differential equation

$$\dot{T^*}(\alpha^*)=\dot{s}\xi(s)ET^*(\alpha^*), \tag{9}$$

where $E=\begin{bmatrix}0 & 1\\ -1 & 0\end{bmatrix}$ and $\xi(s)$ is the path curvature.

From (9) also follows

$$\dot{\alpha^*}=\dot{s}\xi(s). \tag{10}$$

## Angular orientation

Robot angular orientation with respect to curve $S$ is defined as

$$\alpha = \alpha^*(s) + \Delta\alpha, \tag{11}$$

where $\Delta\alpha = const$ is the desired robot orientation with respect to the path.

In matrix notation, (11) takes the form

$$T(\alpha) = T(\Delta\alpha)T(\alpha^*). \tag{12}$$

## Introducing errors and problem statement

Violation of condition (6) is characterised by orthogonal deviation

$$e = \varphi(x, y). \tag{13}$$

Violation of condition (10) is characterised by angular deviation

$$\delta = \alpha - \alpha^* + \Delta\alpha. \tag{14}$$

Therefore, the path following control problem consists in determination of inputs $F_x$, $F_y$ and $M$ in closed loop, which provides:

- stabilization of robot motion with respect to curve $S$;
- stabilization of robot angular orientation with respect to curve $S$;
- maintenance of the desired longitudinal motion by asymptotic zeroing of velocity error

$$\Delta V_s = V_s^* - \dot{s}. \tag{15}$$

## Coordinate transformation

Perform the transformation of the system model (1)-(3) to the task-based form with outputs $s$, $e$ and $\delta$. To do so, differentiate (7), (13) and (14) with respect to time:

$$\begin{bmatrix}\dot s\\ \dot e\end{bmatrix} = \Upsilon(x,y)\begin{bmatrix}\dot x\\ \dot y\end{bmatrix} = T(\alpha^*)\begin{bmatrix}\dot x\\ \dot y\end{bmatrix}, \tag{16}$$

$$\dot\delta = -\xi(s)\dot s + \omega. \tag{17}$$

Find the inverse transformation:

$$\begin{bmatrix}\dot x\\ \dot y\end{bmatrix} = T^T(\alpha^*)\begin{bmatrix}\dot s\\ \dot e\end{bmatrix},$$

$$\omega = \dot\delta + \xi(s)\dot s.$$

## Control design

Once more differentiate (16) and (17) with account for (5), (9) and (12):

$$\begin{bmatrix}\ddot s\\ \ddot e\end{bmatrix} + \xi(s)\dot s E^T\begin{bmatrix}\dot s\\ \dot e\end{bmatrix} = \frac{1}{m}T^T(\Delta\alpha)\begin{bmatrix}F_x\\ F_y\end{bmatrix}, \tag{18}$$

$$\ddot\delta + \xi(s)\ddot s + \dot\xi(s)\dot s = \frac{1}{J}M. \tag{19}$$

Now consider virtual task-based controls:

$$\begin{bmatrix}u_s\\ u_e\end{bmatrix} = \frac{1}{m}T^T(\Delta\alpha)\begin{bmatrix}F_x\\ F_y\end{bmatrix} \tag{20}$$

$$u_\delta = \frac{1}{J}M - \xi(s)u_s \tag{21}$$

## Control design

Substitute (20) and (21) to (18) and (19):

$$\begin{bmatrix} \ddot{s} \\ \ddot{e} \end{bmatrix} + \xi(s)\dot{s}E^T \begin{bmatrix} \dot{s} \\ \dot{e} \end{bmatrix} = \begin{bmatrix} u_s \\ u_e \end{bmatrix}, \qquad (22)$$

$$\ddot{\delta} + \dot{\xi}(s)\dot{s} + \xi^2(s)\dot{s}\dot{e} = u_\delta. \qquad (23)$$

Rewrite equations (22) and (23) with account for (15) for determining the velocity error dynamics:

$$\Delta\dot{V} + \xi(s)\dot{s}\dot{e} = -u_s,$$

$$\ddot{e} + \xi(s)\dot{s}^2 = u_e;$$

$$\ddot{\delta} + \dot{\xi}(s)\dot{s} + \xi^2(s)\dot{s}\dot{e} = u_\delta.$$

## Control design

Now select the controllers:

$$u_s = -\xi(s)\dot{s}\dot{e} + k_s\Delta V, \qquad (24)$$

$$u_e = \xi(s)\dot{s}^2 - k_{e1}\dot{e} - k_{e2}e, \qquad (25)$$

$$u_\delta = \dot{\xi}(s)\dot{s} + \xi^2(s)\dot{s}\dot{e} - k_{\delta 1}\dot{\delta} - k_{\delta 2}\delta, \qquad (26)$$

where $k_s$, $k_{e1}$, $k_{e2}$, $k_{\delta 1}$, $k_{\delta 2}$ are positive constants.

Finaly we determine actual control actions $F_x$, $F_y$ and $M$ and obtain

$$\begin{bmatrix} F_x \\ F_y \end{bmatrix} = mT(\Delta\alpha) \begin{bmatrix} -\xi(s)\dot{s}\dot{e} + k_s\Delta V \\ \xi(s)\dot{s}^2 - k_{e1}\dot{e} - k_{e2}e \end{bmatrix}, \qquad (27)$$

$$M = J(\xi(s)u_s + \dot{\xi}(s)\dot{s} + \xi^2(s)\dot{s}\dot{e} - k_{\delta 1}\dot{\delta} - k_{\delta 2}\delta). \qquad (28)$$

## Example. Straight line segment

The normalized equation of the straight line

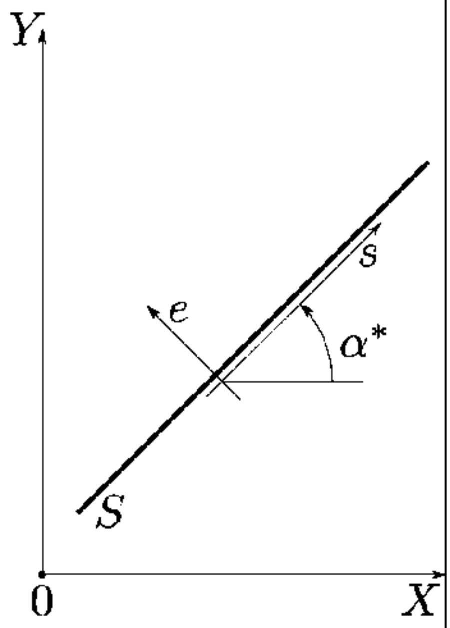
$$\varphi(q) = -\sin\alpha^* x + \cos\alpha^* y + \varphi_0 = 0,$$

$$\psi(q) = \cos\alpha^* x + \sin\alpha^* y + \psi_0,$$

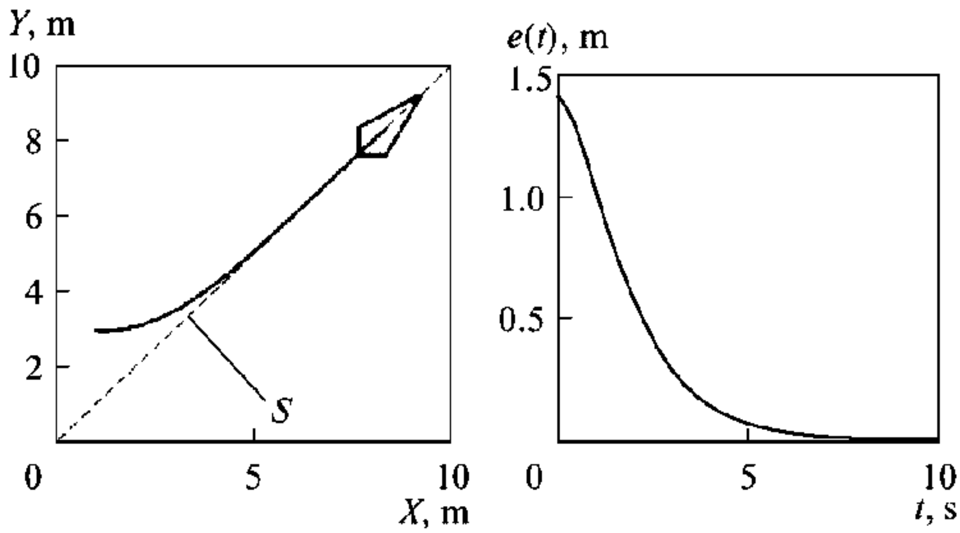
where $\alpha^*$ is the line inclination, $\varphi_0 = const$, $\psi_0 = const$.
Orthogonal Jacobian matrix takes the form

$$\Upsilon(q) = \begin{bmatrix} \cos\alpha^* & \sin\alpha^* \\ -\sin\alpha^* & \cos\alpha^* \end{bmatrix} \in SO(2).$$

Obviously, the path curvature is zero.




## Example. Simulation of the motion along the straight line.



## Example. An arc of a circle

The normalized equation of the arc of the circle

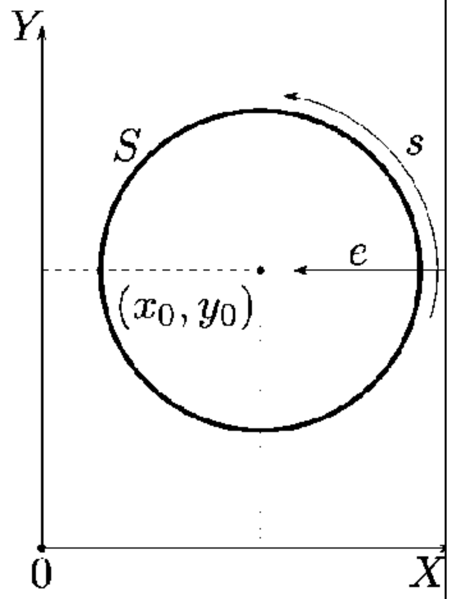
$$\varphi(q) = \frac{1}{2R}(R^2-(x-x_0)^2-(y-y_0)^2) = 0,$$

$$\psi(q) = R\arctan\frac{(y-y_0)}{(x-x_0)}.$$

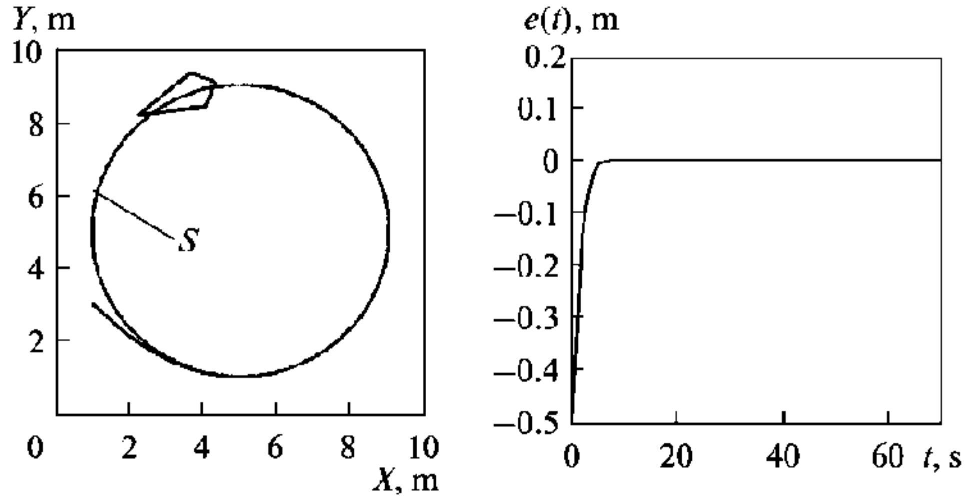
Orthogonal Jacobian matrix takes the form

$$\Upsilon(q) = \frac{1}{R}\begin{bmatrix} -(y-y_0) & (x-x_0) \\ -(x-x_0) & -(y-y_0) \end{bmatrix} \in SO(2).$$

The path curvature is $\xi(s) = \frac{1}{R}$




## Example. Simulation of the motion along the arc of a circle.



## Dynamic model

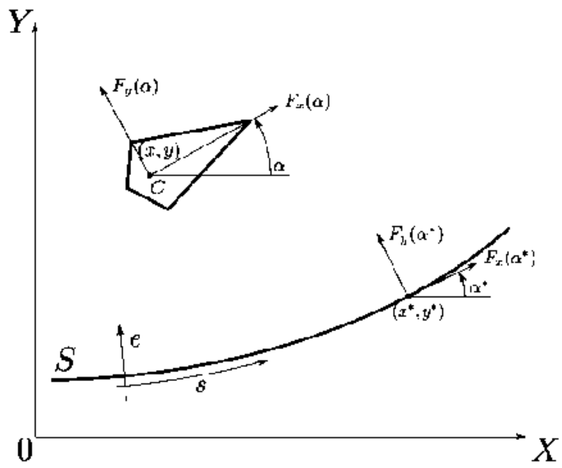
$$m\ddot{q} = F, \qquad (29)$$

$$\dot{q} = R_O^I(\alpha)v, \qquad (30)$$

$$R_O^I(\alpha) = \begin{bmatrix} cos\alpha & sin\alpha \\ -sin\alpha & cos\alpha \end{bmatrix}, \qquad (31)$$

$$\dot{\alpha} = \omega, \qquad (32)$$

where $q = \begin{bmatrix} x & y \end{bmatrix}^T \in R^2$ is vector of the Cartesian coordinates,
$F = \begin{bmatrix} F_x & F_y \end{bmatrix}^T \in R^2$ is vector of the control forces,
$\alpha$ is the orientation angle,
$\omega$ is the angular velocity.




## Motion on the plane

The desired path is an implicitly described smooth segment of curve $S$:

$$\varphi(q) = 0, \qquad (33)$$

and relevant local coordinate $s$ (path length) is defined as

$$s = \psi(q) \qquad (34)$$

Selection of functions (33) and (34) is mostly limited by regularity condition implying that Jacobian matrix

$$\Upsilon(q) = \begin{bmatrix} \frac{\partial\psi(q)}{\partial x} & \frac{\partial\psi(q)}{\partial y} \\ \frac{\partial\varphi(q)}{\partial x} & \frac{\partial\varphi(q)}{\partial y} \end{bmatrix} \qquad (35)$$

is not degenerate for any $(x, y)$ belonging to curve $S$, i.e. $det\Upsilon(x, y) \neq 0$

## Control design

Design of the velocity (inner) loop. Consider Lyapunov Function:

$$V_1 = \frac{1}{2}(\dot{q} - \bar{v})^T(\dot{q} - \bar{v}), \tag{36}$$

where $\bar{v}$ - a vector of desired velocities.

Find the derivation of the Lyapunov function $V_1$:

$$\dot{V}_1 = (\dot{q} - \bar{v})^T(\ddot{q} - \dot{\bar{v}}) = (\dot{q} - \bar{v})^T(\frac{F}{m} - \dot{\bar{v}}). \tag{37}$$

## Control design

Define the control signal as:

$$\frac{1}{m}F = \dot{\bar{v}} - k_q(\dot{q} - \bar{v}), \tag{38}$$

where $k_q$ is a positive constant. Then the derivation of the Lyapunov function $V_1$ is

$$\dot{V}_1 = -k_q(\dot{q} - \bar{v})^T(\dot{q} - \bar{v}) \leq 0, \tag{39}$$

which means asymptotic stability of the point $\dot{q} - v = 0$.

Now we can rewrite original system in reduced form:

$$\dot{q} = \bar{v}.$$

Let's construct the control $\bar{v}$ in th following form:

$$\bar{v} = u_e + u_s,$$

where $u_e$ is the term, which provides stabilization with respect to the desired path and $u_s$ provides desired velocity along the path.

## Reduced system

Perform the transformation of the system model (29)-(32) to the task-based form with outputs $s$ and $e_1$, using Jacobian matrix (35):

$$\begin{bmatrix} \dot{s} \\ \dot{e}_1 \end{bmatrix} = \Upsilon(q)\dot{q} = \Upsilon(q) R_I^O(\alpha) v. \quad (40)$$

We can choose the control signal $u_s$ in the form

$$u_s = R_O^I \Upsilon^{-1}(r) \begin{bmatrix} V^* \\ 0 \end{bmatrix} \quad (41)$$

Now design stabilization control $u_e$. Consider Lyapunov Function:

$$V_2 = \frac{k_e}{2}\varphi^2(q), \quad (42)$$

## Reduced system

Find the derivation of the Lyapunov function $V_2$.

$$\dot{V}_2 = k_e\varphi(q)\nabla\varphi(q) = (k_e\varphi(q)\nabla\varphi(q))^\top u_e +$$
$$+(k_e\varphi(q)\nabla\varphi(q))^\top \Upsilon^{-1}q)\begin{bmatrix} V^* \\ 0 \end{bmatrix} = (k_e\varphi(q)\nabla\varphi(q))^\top u_e.$$

As you can see, the second half of the expression is identically zero due to orthogonality. Now select $u_e$ as

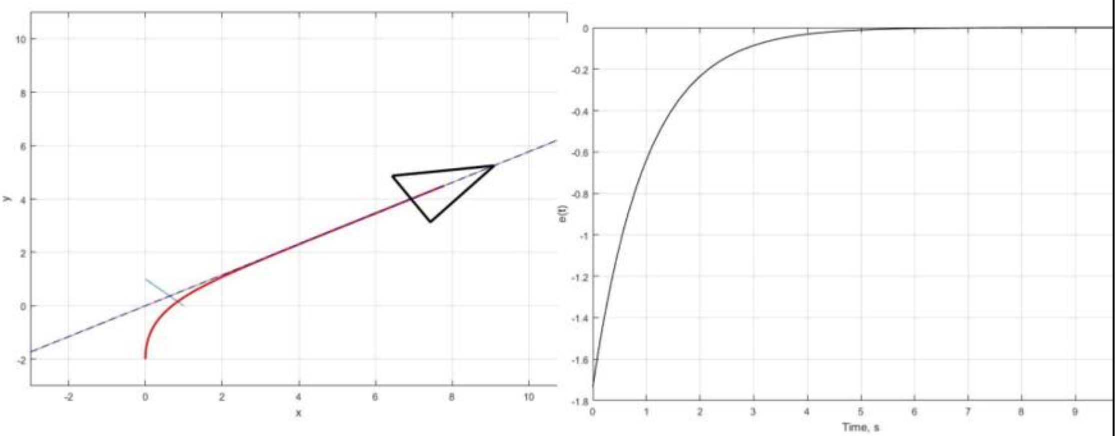
$$u_e = -k_e\varphi(q)\frac{\partial}{\partial q}\varphi(q), \qquad (43)$$

where $k_e$ is positive constant.
Then the derivation of the Lyapunov function $V_2$ is

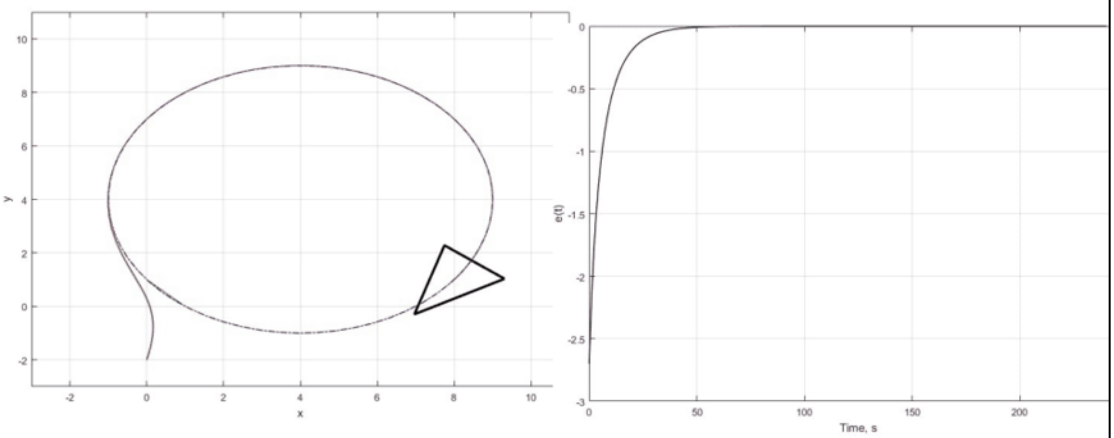
$$\dot{V}_2 = -u_e^2 \leq 0$$

It proves the asymptotic stability of the initial system at the point $e(q) = 0$.

## Example. Simulation of the motion along the straight line.



## Example. Simulation of the motion along the arc of a circle.




## 2D moving frame

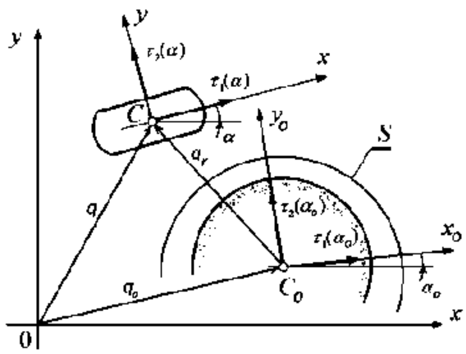
Dynamic model of the plant:

$$\begin{bmatrix} \dot{q} \\ \dot{\alpha} \end{bmatrix} = R^T(\alpha) \begin{bmatrix} v \\ \omega \end{bmatrix}, \qquad (44)$$

$$A \begin{bmatrix} \dot{v} \\ \dot{\omega} \end{bmatrix} = R^T(\alpha) \begin{bmatrix} F \\ M \end{bmatrix}, \qquad (45)$$

$$R(\alpha) = \begin{bmatrix} T(\alpha) & 0 \\ 0 & 1 \end{bmatrix},$$

$$A = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & J \end{bmatrix},$$



## External moving object

Dynamic model of the external moving object:

$$\dot{q}_o = v_o, \tag{46}$$

$$\dot{T}(\alpha_o) = \omega_o E T(\alpha_o), \tag{47}$$

Desired trajectory in relative coordinates

$$\varphi(q_r) = 0, \tag{48}$$

Local coordinate

$$s = \psi(q_r), \tag{49}$$

## Relative coordinates

Position, velocity and acceleration of the plant in moving frame:

$$q_r = T(\alpha_o)(q - q_o), \tag{50}$$

$$\alpha_r = \alpha - \alpha_o. \tag{51}$$

$$\dot{q}_r = \omega_o E q_r + T(\alpha_o)\left(\dot{q} - \dot{q}_o\right), \tag{52}$$

$$\dot{\alpha}_r = \omega - \omega_o, \tag{53}$$

$$\begin{aligned} \ddot{q}_r &= (\omega_o)^2 q_r + 2\omega_o E T(\alpha_o)\left(\dot{q} - \dot{q}_o\right) + \\ &+ \frac{1}{m} T(\alpha_o) T^T(\alpha) F_z, \end{aligned} \tag{54}$$

$$\ddot{\alpha}_r = \frac{1}{J} M. \tag{55}$$

## Task-oriented coordinates

Consider orthogonal deviation

$$e(q_r) = \varphi(q_r), \tag{56}$$

and local coordinate $s$

$$s = \psi(q_r) \tag{57}$$

Choosing of functions (56) and (57) based on regularity condition which implies that Jacoby matrix

$$\Upsilon(q_r) = \begin{bmatrix} \partial\psi/\partial q_r \\ \partial\varphi/\partial q_r \end{bmatrix} \tag{58}$$

is nondegenerate for all $q_r$, belongs to curve $S$, i.e. $det\Upsilon(q_r) \neq 0$.

## Trajectory control synthesis

Imply the transformation of model (44)-(45) to the task-oriented coordinates:

$$\begin{bmatrix} \dot{s} \\ \dot{e} \end{bmatrix} = T(\alpha_r^*)\left(T^T(\alpha_r)v_z + \omega_o E q_r - T(\alpha_o)v_o\right), \tag{59}$$

$$\dot{\delta} = -\dot{s}\xi(s) + \omega - \omega_o. \tag{60}$$

Choose local controllers as

$$\begin{aligned} u_s &= k_s\Delta V - \dot{s}\xi(s)\dot{e} - 2\omega_o\dot{e}, & (61)\\ u_e &= k_{e1}e + k_{e2}\dot{e} + \dot{s}^2\xi(s)\dot{e} + 2\omega_o\dot{s}, & (62)\\ u_\delta &= k_{\delta 1}\delta + k_{\delta 2}\dot{\delta} + \frac{\partial\xi}{\partial s}\dot{s} + \ddot{s}\xi(s). & (63) \end{aligned}$$

Final control laws

$$\begin{aligned} F &= mT(\alpha_r)T^T(\alpha_r^*)\left(\begin{bmatrix} u_s \\ u_e \end{bmatrix} - (\omega_o)^2 T(\alpha^*)q_r\right), & (64)\\ M &= Ju_\delta. & (65) \end{aligned}$$

# Collision avoidance strategies

Some collision avoidance strategies

- Bypass
- Detour

Equidistant border around the obstacle

$$\varphi^*(q) = x^2 + y^2 - R^2 = 0, \tag{66}$$

# Trajectory control in presence of moving external object.

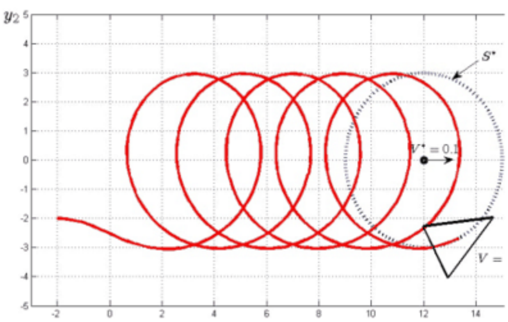



**Figure 1:** The results of modeling the motion relative to a moving external object.

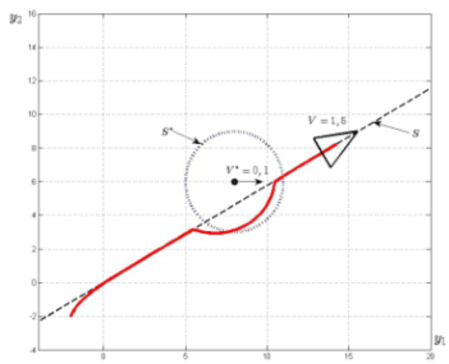



**Figure 2:** The results of modeling the detour of a moving obstacle.

## Spatial motion

Dynamic model:

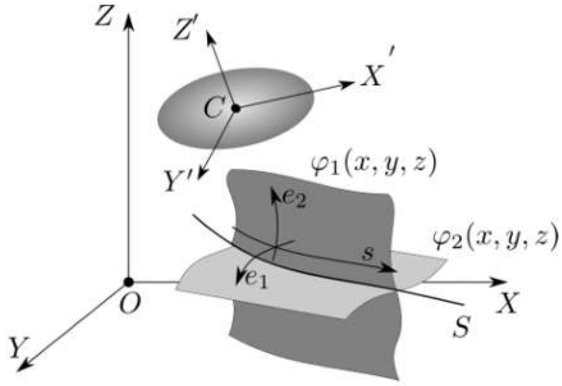
$$\dot{q} = v, \tag{67}$$

$$\dot{v} = \frac{1}{m} F_c, \tag{68}$$

$$\dot{R}(\alpha) = S(\omega) R(\alpha), \tag{69}$$

$$J\dot{\omega} + \omega \times J\omega = M_c, \tag{70}$$

$$S(\omega) = \begin{bmatrix} 0 & \omega_3 & -\omega_2 \\ -\omega_3 & 0 & \omega_1 \\ \omega_2 & -\omega_1 & 0 \end{bmatrix}.$$




## Rotational matrix

The rotation matrix $R(\alpha)$ can be represented through Euler angles as

$$R(\alpha) = R_3(\psi) R_2(\theta) R_1(\phi), \tag{71}$$

where

$$R_1(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix}$$

$$R_2(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$

$$R_3(\psi) = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

## Desired path

The desired path $S$ describes as an intersection of two implicit surfaces:

$$\varphi_1(x, y, z) = 0 \cap \varphi_2(x, y, z) = 0. \qquad (72)$$

Tangential velocity along the curve $S$ is defined as

$$\dot{s} = \frac{\nabla\varphi_1 \times \nabla\varphi_2}{\|\nabla\varphi_1 \times \nabla\varphi_2\|} v, \qquad (73)$$

where $\times$ is the vector product and $\|\cdot\|$ is the vector norm.
Jacobian matrix:

$$\Upsilon(x, y, z) = \begin{bmatrix} \frac{\nabla\varphi_1 \times \nabla\varphi_2}{\|\nabla\varphi_1 \times \nabla\varphi_2\|} \\ \frac{\nabla\varphi_1}{\|\nabla\varphi_1\|} \\ \frac{\nabla\varphi_2}{\|\nabla\varphi_2\|} \end{bmatrix} \qquad (74)$$

## Introducing errors and problem statement

Violation of condition (72) is characterised by orthogonal deviations

$$e_1 = \varphi_1(x, y, z). \qquad (75)$$

$$e_2 = \varphi_2(x, y, z). \qquad (76)$$

Therefore, the path following control problem consists in determination of inputs $F_c = \begin{bmatrix} F_x & F_y & F_z \end{bmatrix}$ and $M_c$ in closed loop, which provides:

- stabilization of robot motion with respect to curve $S$;
- maintenance of the desired longitudinal motion by asymptotic zeroing of velocity error

$$\Delta V_s = V_s^* - \dot{s}; \qquad (77)$$

- stabilization of robot angular orientation with respect to curve $S$.

## Translation motion control

Perform the transformation of the system model (67)-(70) to the task-based form with outputs $s$, $e_1$ and $e_2$. To do so, differentiate (73), (75) and (76) with respect to time:

$$\begin{bmatrix} \dot{s} \\ \dot{e_1} \\ \dot{e_2} \end{bmatrix} = \Upsilon(x, y, z)v. \tag{78}$$

Once more differentiate (78) with account for (68):

$$\begin{bmatrix} \ddot{s} \\ \ddot{e_1} \\ \ddot{e_2} \end{bmatrix} - \dot{\Upsilon}(x, y, z)v + \Upsilon(x, y, z)\frac{F_c}{m}. \tag{79}$$

## Translation motion control

Consider the virtual (task-based) controls:

$$\dot{\Upsilon}(x, y, z)v + \Upsilon(x, y, z)\frac{F_c}{m} = \begin{bmatrix} u_s \\ u_{e1} \\ u_{e2} \end{bmatrix} \tag{80}$$

Substitute (80) to (79) and obtain

$$\begin{bmatrix} \ddot{s} \\ \ddot{e_1} \\ \ddot{e_2} \end{bmatrix} = \begin{bmatrix} u_s \\ u_{e1} \\ u_{e2} \end{bmatrix}. \tag{81}$$

## Translation motion control

Now select the controllers:

$$u_s = K_s \Delta \dot{s}, \tag{82}$$

$$u_{e1} = -K_{1e1}\dot{e_1} - K_{2e1}e_1, \tag{83}$$

$$u_{e2} = -K_{1e2}\dot{e_2} - K_{2e2}e_2, \tag{84}$$

where $K_s$, $K_{1e1}$, $K_{2e1}$, $K_{1e2}$, $K_{2e2}$ are positive constants.

Finaly we determine actual control action $F_c$ and obtain

$$F_c = m\Upsilon(x,y,z)^{-1}(\begin{bmatrix} u_s \\ u_{e1} \\ u_{e2} \end{bmatrix} - \dot{\Upsilon}(x,y,z)v). \tag{85}$$

## Rotation motion control

Introduce vector of angular errors $\delta - \begin{bmatrix}\delta_\phi & \delta_\theta & \delta_\psi\end{bmatrix}^T \in R^3$ and the angular deviation matrix

$$R(\delta) = R(\alpha)R^T(\alpha^*)R^T(\Delta), \tag{86}$$

where $R(\alpha^*) \in SO(3)$ is the matrix of angular orientation of the body-fixed frame along the curve $S$, $R(\Delta) \in SO(3)$ is the matrix of the desired angular orientation. Define the angular error function as

$$e_r = \frac{1}{2}(R(\delta) - R(\delta)^T)^\vee, \tag{87}$$

where $\vee$ it the transformation $SO(3) \to R^3$.

## Rotation motion control

Define the angular speed error $e_\omega$. Differentiate (86) with account for (69) and obtain the equation

$$\frac{d}{dt}R(\delta) = S(\dot{\delta})R(\delta) = e_\omega R(\delta), \tag{88}$$

$$\frac{d}{dt}R(\delta) = S(\omega)R(\delta) - R(\alpha)R^T(\alpha^*)S(\omega^*)R^T(\Delta), \tag{89}$$

Use the property of skew symmetric matrix $RS(\omega)R^T = S(R\omega)$ and obtain final expression

$$\frac{d}{dt}R(\delta) = (S(\omega) - S(R(\alpha)R^T(\alpha^*)\omega^*))R(\delta), \tag{90}$$

and

$$e_\omega = \omega - R(\alpha)R^T(\alpha^*)\omega^*. \tag{91}$$

## Rotation motion control

Differentiating (91) with account for (69)

$$\dot{e_\omega} = \frac{1}{J}(M - \omega \times J\omega) + a_d, \tag{92}$$

where $a_d = -S(\omega)R(\alpha)R^T(\alpha^*)\omega^* + R(\alpha)R^T(\alpha^*)\dot{\omega}^*$. Resulting attitude controller has form

$$M_c = \omega \times J\omega - Ja_d - K_R e_r - K_\omega e_\omega. \tag{93}$$

where $K_R$, $K_\omega$, are positive constants.

## Numerical example

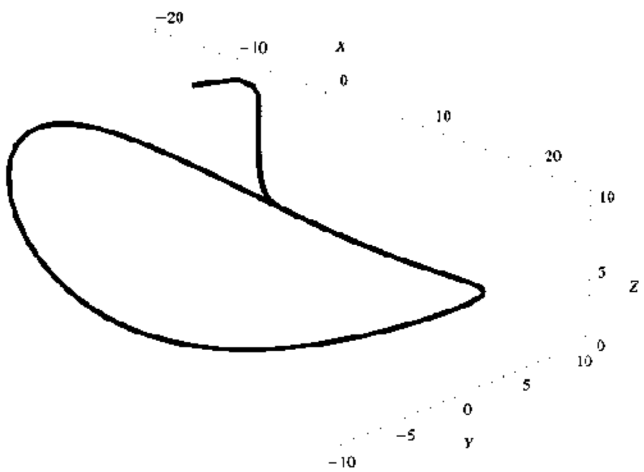
Consider the plant as a rigid body described by model (67)-(70) with $m = 1$, $J = 1$.

Initial position of the plant is $x_0 = \begin{bmatrix} -10 & 5 & 10 \end{bmatrix}^T$ and initial orientation is $\alpha_0 = \begin{bmatrix} 3 & 2 & 1 \end{bmatrix}^T$.

Parameters of the controller are $K_{1e1} = 1$, $K_{2e1} = 10$, $K_{1e2} = 1$, $K_{2e2} = 10$, $K_R = 20$, $K_\omega = 50$.

Desired speed along the path $\dot{s} = 1$.

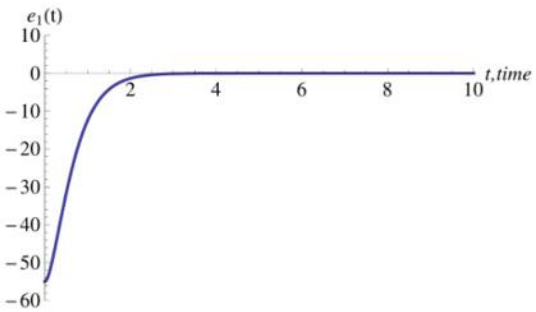
## Numerical example



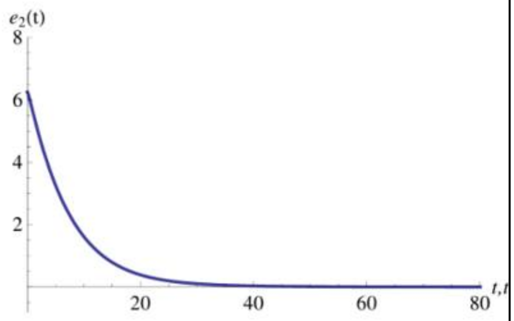

Motion along desired path:

$$\varphi_1(x, y, z) = 0.2x^2 + y^2 - R^2 = 0 \;\cap\; \varphi_2(x, y, z) = z + 0.05y^2 - 5 = 0$$

## Numerical example

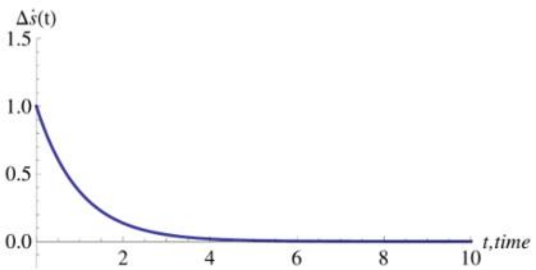



Position error $e_1 = \varphi_1(x, y, z)$

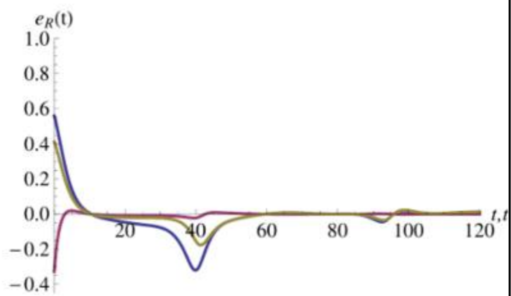



Position error $e_2 = \varphi_2(x, y, z)$

## Numerical example




Speed error $\Delta V = \dot{s}^* - \dot{s}$




Angular error $e_r$

## Moving frame description

Model of the plant motion:

$$\ddot{x}(t) = g - \frac{f(t)}{m}\vec{n}(t), \qquad (94)$$

$$\dot{R}(t) = R(t)S(\omega(t)), \qquad (95)$$

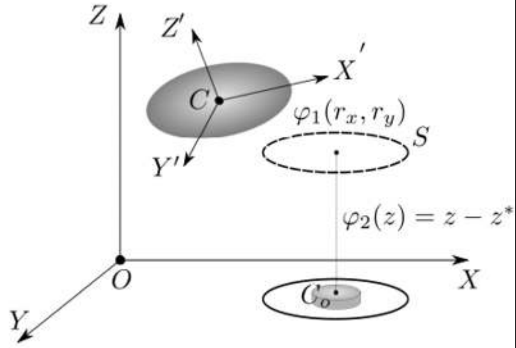
$$M_c(t) = J\dot{\omega}(t) + \omega(t) \times J\omega(t). \qquad (96)$$

Description of the moving frame:

$$\dot{x}_i = R_T(\alpha^*)v_i, \qquad (97)$$

$$\dot{\alpha}^* = \omega_i, \qquad (98)$$

$$\dot{R}_T(\alpha^*) = R_T S(\omega_i). \qquad (99)$$




## Moving frame description

Relative position:

$$r = R_T^\top(\alpha^*)(x - x_i), \qquad (100)$$

Relative velocity:

$$\dot{r} = R_T^\top(\alpha^*)\dot{x} - S(\omega_i)r, \qquad (101)$$

Relative acceleration:

$$\ddot{r} = R_T^\top(\alpha^*)\ddot{x} - 2S(\omega_i)\dot{r} - S^2(\omega_i)r. \qquad (102)$$

## Control design

Design of the velocity (inner) loop. Consider Lyapunov Function:

$$V_1 = \frac{1}{2}(\dot{r}_x - \bar{v})^T(\dot{r}_x - \bar{v}) + k_d \ln(2 - (R^\top \bar{n})^\top (R^\top R_T \bar{n}_d)), \qquad (103)$$

where $\bar{v}$ - a vector of desired velocities, $\bar{n}_d$ - a vector of desired orientation and $k_d$ - a positive constant.

Find the derivation of the Lyapunov function $V_1$:

$$\dot{V}_1 = (\dot{r} - \bar{u})^\top (R_T^\top - \frac{f}{m} R_T^\top \bar{n} - 2S(\omega_i)\dot{r} - S^2(\omega_i)r - \dot{\bar{u}})$$
$$+\gamma^\top \left( \omega - \omega_i - \frac{S(R^\top R_T \bar{n}_d)}{|\bar{n}_d|} R^\top R_T \dot{\bar{n}}_d \right), \qquad (104)$$

where $\gamma^\top = \frac{k_d (R^\top \bar{n})^\top S^\top (R^\top R_T \bar{n}_d)}{(2-(R^\top \bar{n})^\top (R^\top R_T \bar{n}_d))}$ and $|a|$ is the euclidean norm of vector a.

## Control design

Define the substitution of variables if following form:

$$\delta = R_T^\top g - 2S(\omega_i)\dot{r} - S^2(\omega_i)r - \dot{\bar{u}}, \delta = \frac{f_d}{m}\bar{n}_d,$$

where $f_d = |\delta|$ and $\bar{n}_d = \frac{\delta}{|\delta|}$.

Select control signals $f$ and $\omega = \omega_i$ in the form

$$f = f_d \cdot ((R_T^\top \bar{n})^\top \bar{n}_d) \quad k_v(\dot{r} \quad \bar{u})^\top R_T^\top \bar{n}, \qquad (105)$$

$$\omega_d = \omega_i + \frac{S(R^\top R_T \bar{n}_d)}{|\bar{n}_d|} R^\top R_T \dot{\bar{n}}_d) + \sigma - K_\gamma \gamma, \qquad (106)$$

where $k_v$, $k_\gamma$ are positive constants and $\sigma$ is

$$\sigma = \left( \frac{f_d(2 - (R^\top n)^\top (R^\top R_T n_d))}{m k_d} (\dot{r} - \bar{u})^\top S(R_T^\top \bar{n}) R_T^\top R \right). \qquad (107)$$

Then the derivation of the Lyapunov function $V_1$ is

$$\dot{V}_1 = -k_v((\dot{r} - \bar{u})^\top R_T^\top \bar{n})^2 - k_\gamma \gamma^\top \gamma \leq 0, \quad (108)$$

which means asymptotic stability of the point $\dot{r} - \bar{u} = 0$, $\bar{n} - \bar{n}_d$.
Now we can rewrite original system in reduced form:

$$\dot{r} = \bar{u}.$$

Let's construct the control $\bar{u}$ in th following form:

$$\bar{u} = u_e + u_s,$$

where $u_e$ is the term, which provides stabilization with respect to the desired path and $u_s$ provides desired velocity along the path.

## Reduced system

Perform the transformation of the system model (100)-(102) to the task-based form with outputs $s$, $e_1$ and $e_2$, using Jacobian matrix (74):

$$\begin{bmatrix} \dot{s} \\ \dot{e}_1 \\ \dot{e}_2 \end{bmatrix} = \Upsilon(r)\dot{r}$$

We can choose the control signal $u_s$ in the form

$$u_s = \Upsilon^{-1}(r) \begin{bmatrix} V_s^* \\ 0 \\ 0 \end{bmatrix} \quad (109)$$

Now design stabilization control $u_e$. Consider Lyapunov Function:

$$V_2 = \frac{k_1}{2}\varphi_1^2(r) + \frac{k_2}{2}\varphi_2^2(r), \quad (110)$$

## Reduced system

Find the derivation of the Lyapunov function $V_2$.

$$\dot{V}_2 = (k_1\varphi_1(r)\nabla\varphi_1(r) + k_2\varphi_2(r)\nabla\varphi_2(r))^\top \dot{r} =$$
$$(k_1\varphi_1(r)\nabla\varphi_1(r) + k_2\varphi_2(r)\nabla\varphi_2(r))^\top u_s +$$
$$+(k_1\varphi_1(r)\nabla\varphi_1(r) + k_2\varphi_2(r)\nabla\varphi_2(r))^\top \Upsilon^{-1}r) \begin{bmatrix} V^* \\ 0 \\ 0 \end{bmatrix} =$$
$$(k_1\varphi_1(r)\nabla\varphi_1(r) + k_2\varphi_2(r)\nabla\varphi_2(r))^\top u_s.$$

As you can see, the second half of the expression is identically zero due to orthogonality. Now select $u_e$ as

$$u_e = -(k_1\varphi_1(r)\nabla\varphi_1(r) + k_2\varphi_2(r)\nabla\varphi_2(r)), \qquad (111)$$

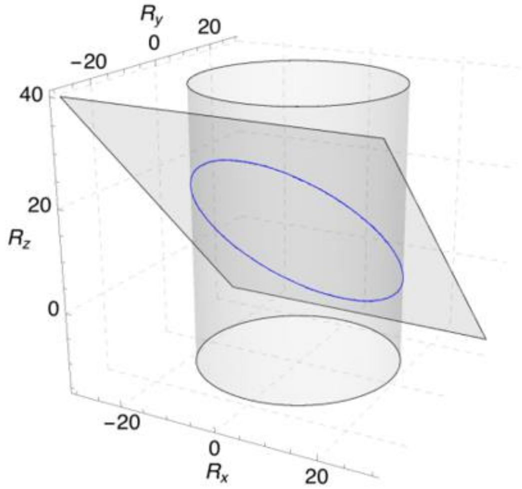
where $k_1$ and $k_2$ are positive constants.

## Resulting control

Resulting control:

$$\begin{aligned}
M_c &= \omega \times J\omega + J\dot{\omega}_d + k_\omega J(\omega - \omega_d), \\
\omega_d &= \omega_T + \frac{S\left(R^\top R_T \bar{n}_d\right)}{|\bar{n}_d|} R^\top R_T \dot{\bar{n}}_d + \sigma - k_\gamma \gamma, \\
\sigma^\top &= \frac{f_d \cdot \left(2 - (R^\top \bar{n})^\top (R^\top R_T \bar{n}_d)\right)}{mk_d} (\dot{r} - u)^\top S(R_T^\top n) R_T^\top R \\
\gamma^\top &= \frac{k_d (R^\top \bar{n})^\top S^\top \left(R^\top R_T \bar{n}_d\right)}{(2 - (R^\top \bar{n})^\top (R^\top R_T \bar{n}_d))}, \\
\delta &= R_T^\top g - 2S(\omega_T)\dot{r} - S^2(\omega_T) r - \dot{\bar{u}}, \\
f_d &= |\delta|, \quad \bar{n}_d = \frac{\delta}{|\delta|}, \\
f &= f_d \cdot ((R_T^\top \bar{n})^\top \bar{n}_d) - k_v (\dot{r} - \bar{u})^\top R_T^\top \bar{n}.
\end{aligned}$$

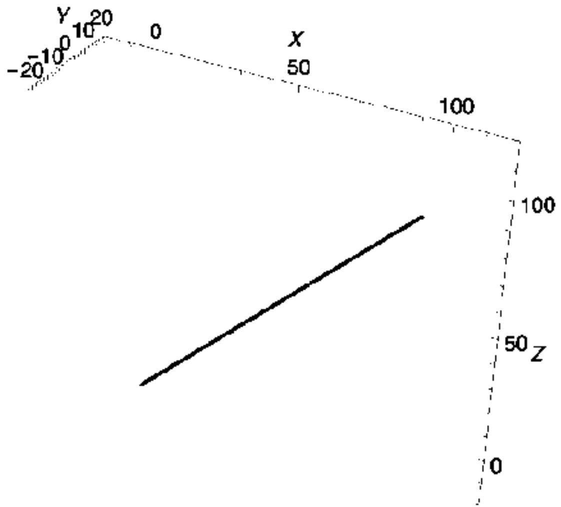
## Example



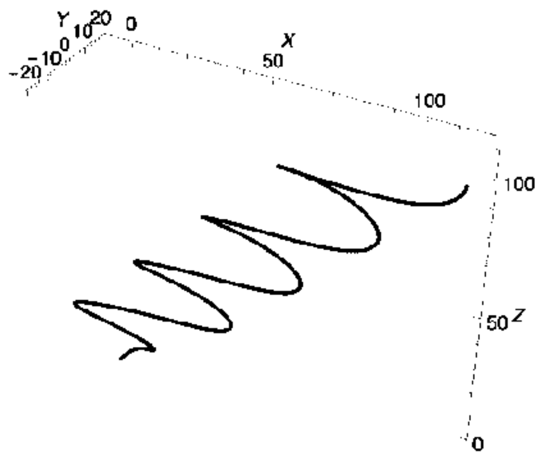

$$\varphi_1(r) = r_x^2 + r_y^2 - 400 = 0 \cap \varphi_2(r) = r_z + r_y - 10 = 0$$

The desired speed along the given path $\dot{s}^* = 30$.

## Example

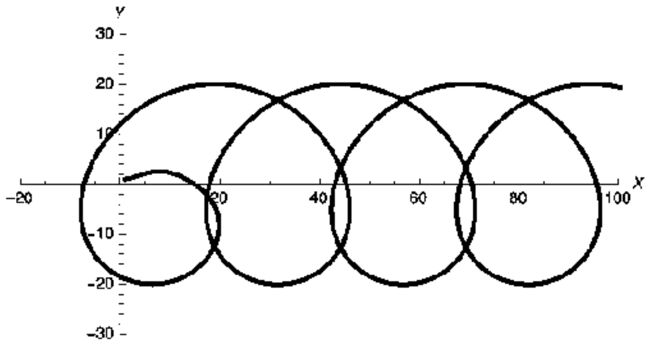



Moving frame spatial motion

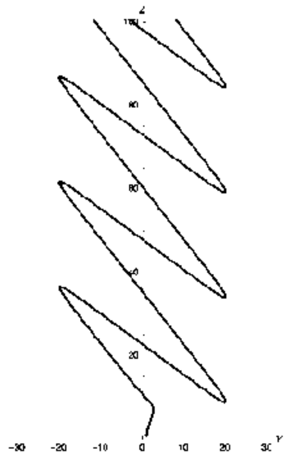



Plant spatial motion

## Example

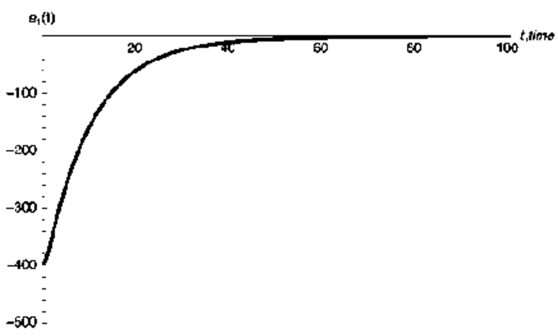


Projection of the plant motion on $XY$ plane

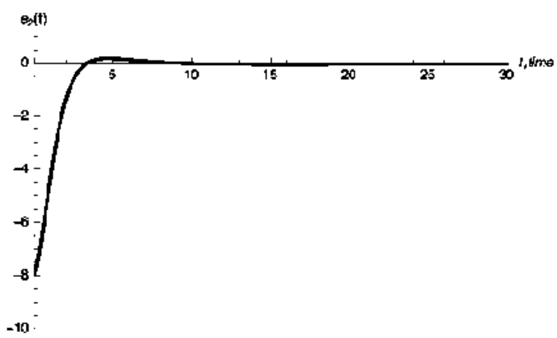


Projection of the plant motion on $YZ$ plane

## Example

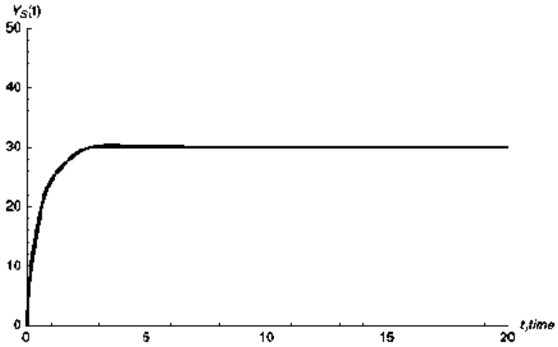


$e_1 = \varphi_1(r)$

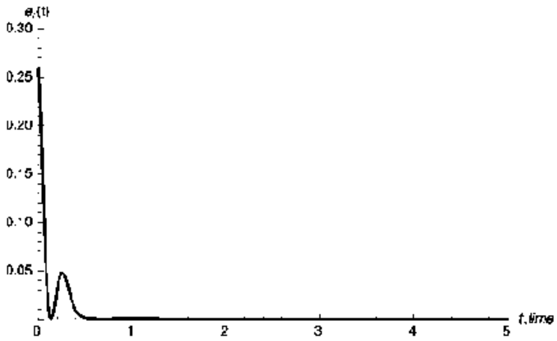


$e_2 = \varphi_2(r)$

## Example

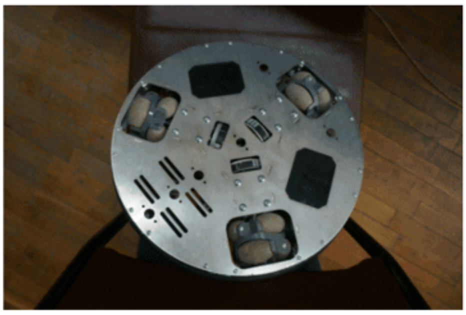



The velocity along the path
$V^*(t) = 30$




Angular error
$e_r = 1 - (R^\top \bar{n})^\top (R^\top R_T \bar{n}_d)$

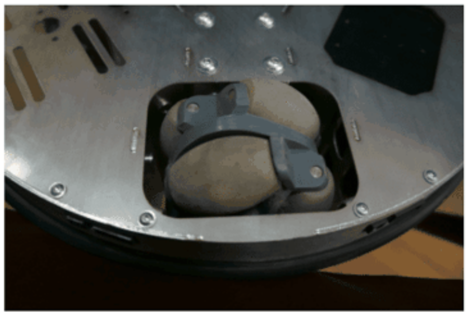
## Omnidirectional mobile robot "Robotino" by Festo Didactics



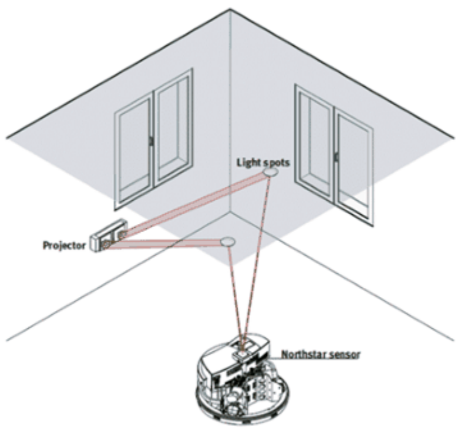
Geometric dimensions:

- Diameter: 370 mm
- Height: 210 mm
- Weight: 11 kg

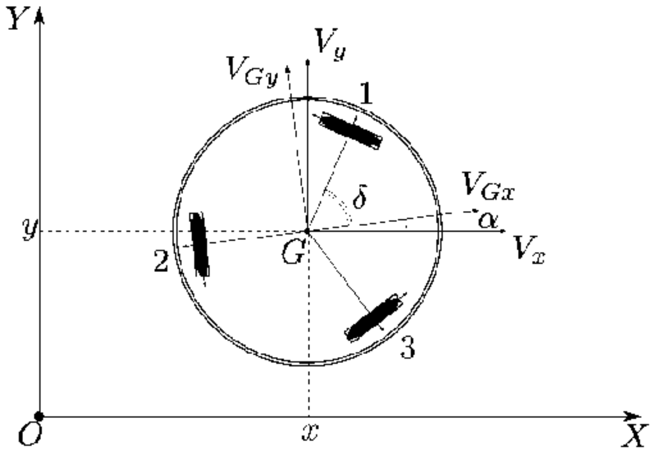
## Omni wheels “Robotino”



## Local Navigation “Northstar”

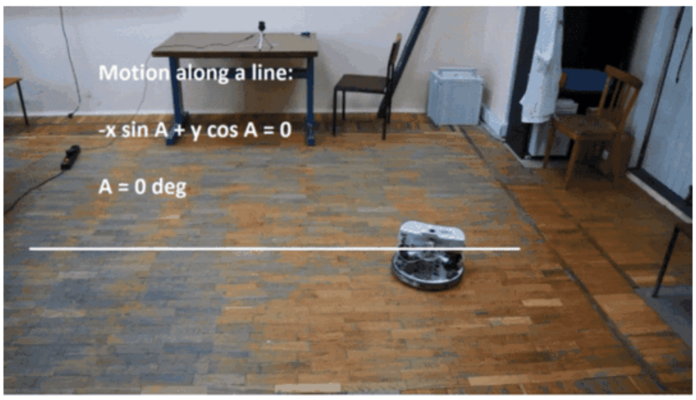


## Mathematical model

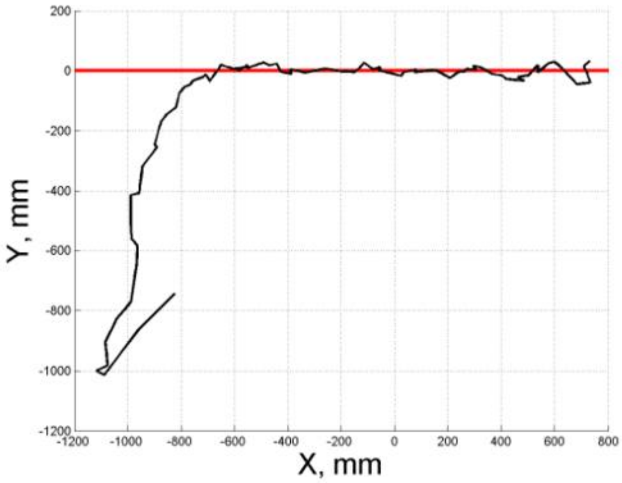



$$\begin{bmatrix} \dot\theta_1 \\ \dot\theta_2 \\ \dot\theta_3 \end{bmatrix} = \begin{bmatrix} -\sin\frac{\pi}{3} & \cos\frac{\pi}{3} & L \\ 0 & -1 & L \\ \sin\frac{\pi}{3} & \cos\frac{\pi}{3} & L \end{bmatrix} \begin{bmatrix} V_{Gx} \\ V_{Gy} \\ \dot\alpha \end{bmatrix}$$

## Motion along a straight line

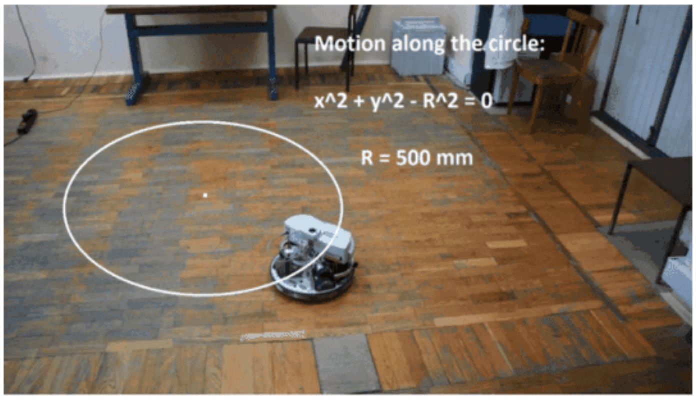


## Motion along a straight line

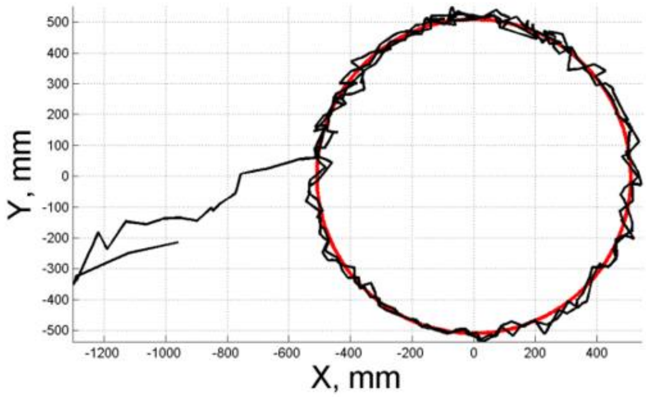



$$\varphi(x, y) = -\sin \alpha x + \cos \alpha y = 0,$$

## Motion along the circle

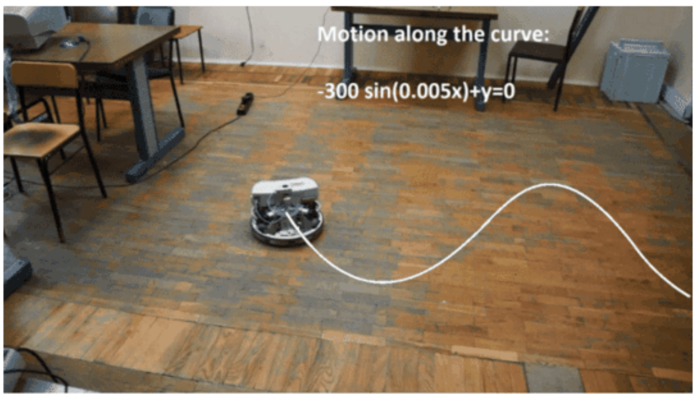


## Motion along the circle

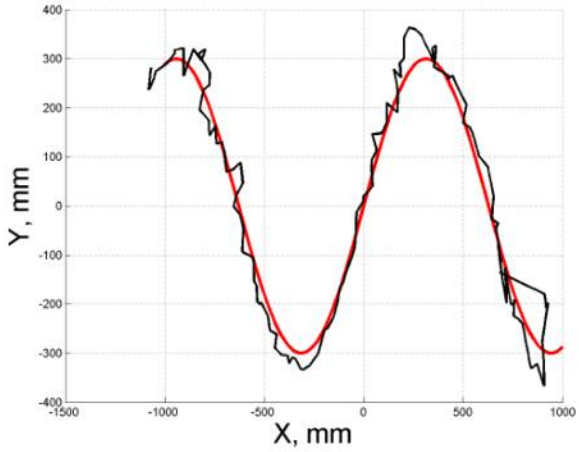



$$\varphi(x,y) = x^2 + y^2 - 2500 = 0$$

## Motion along the sinusoid

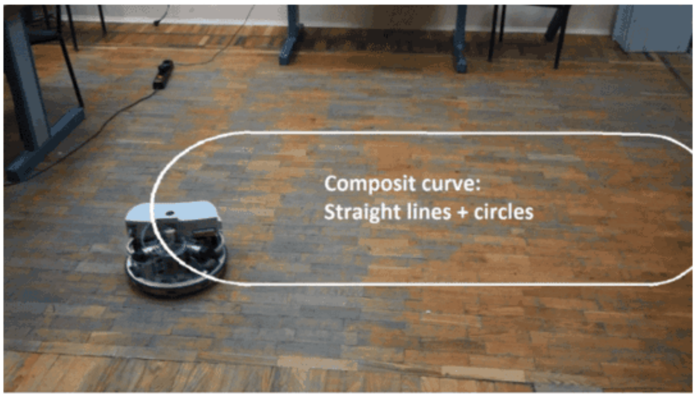


## Motion along the sinusoid

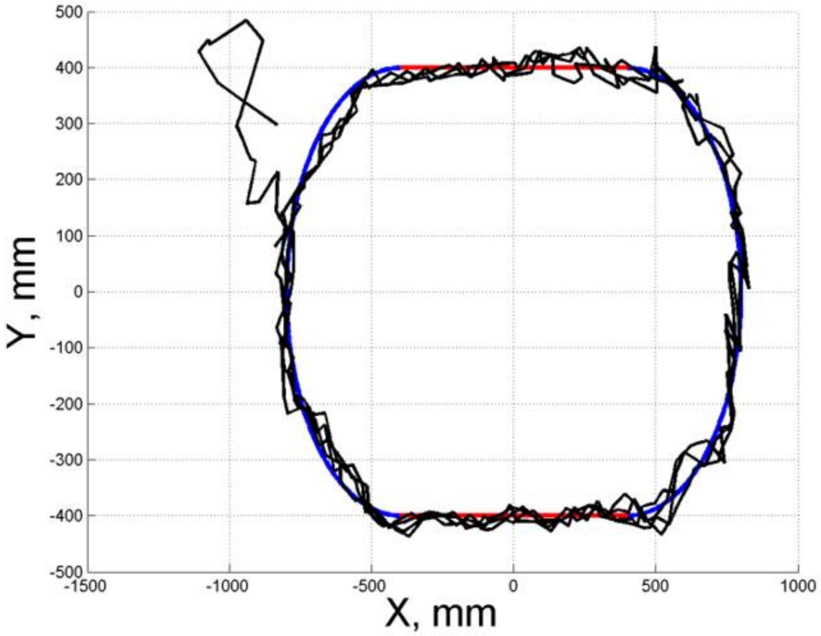



$$\varphi(x,y) = -300\sin 0.005x + y = 0$$

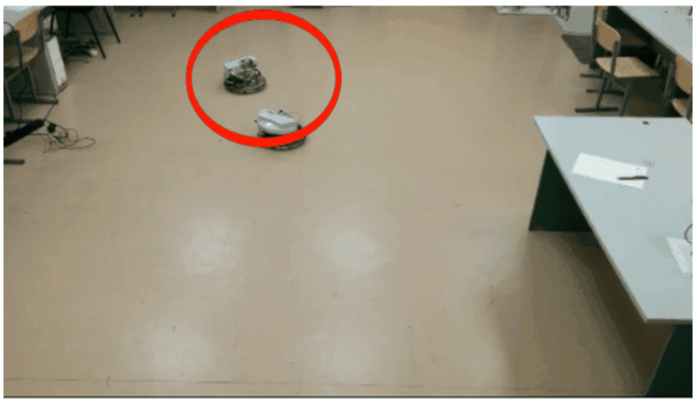
## Motion along a complex curve



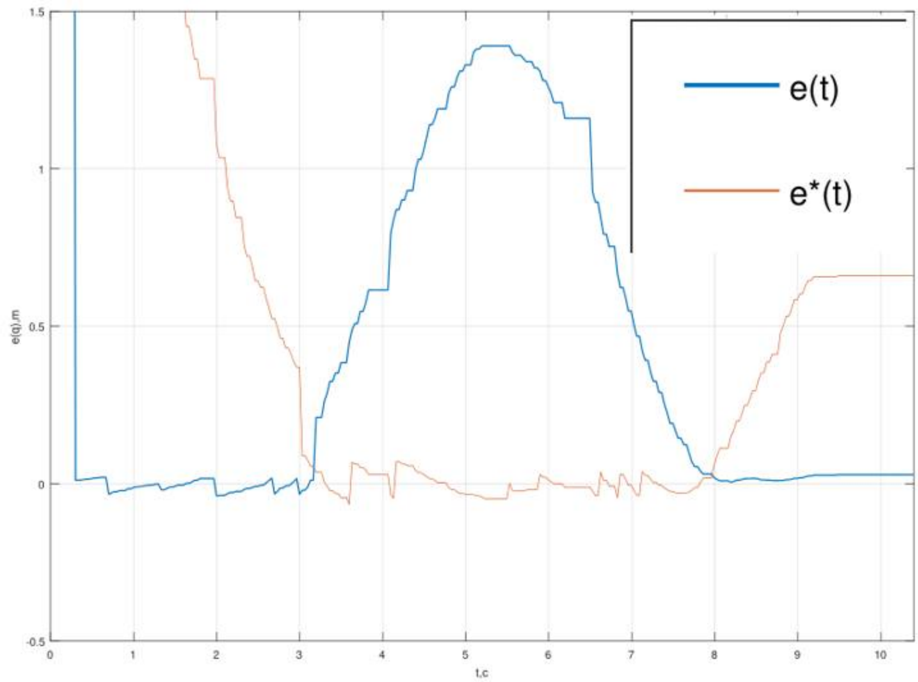
## Motion along a complex curve



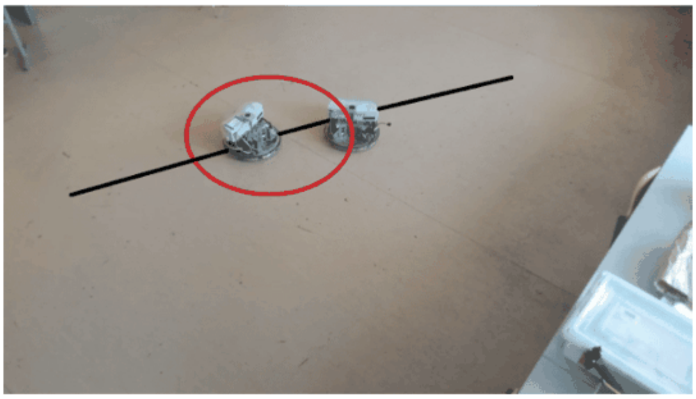

## Avoiding moving obstacle



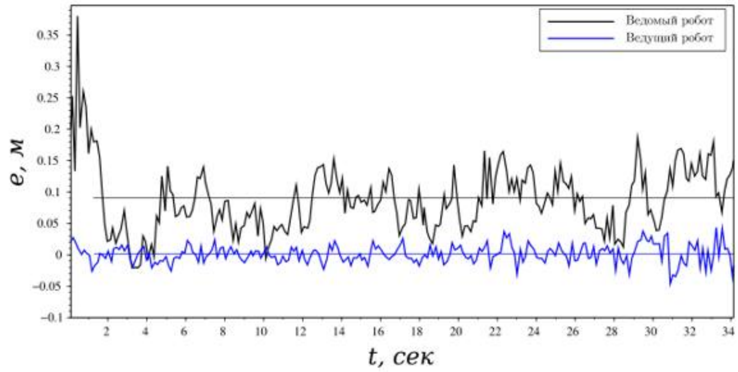
## Avoiding moving obstacle




## Coordinated motion of two robots



## Coordinated motion of two robots

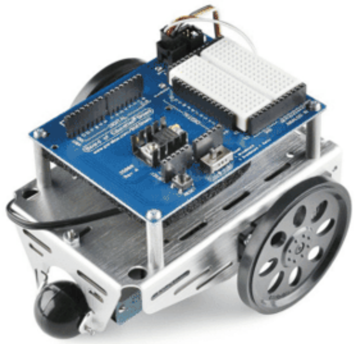


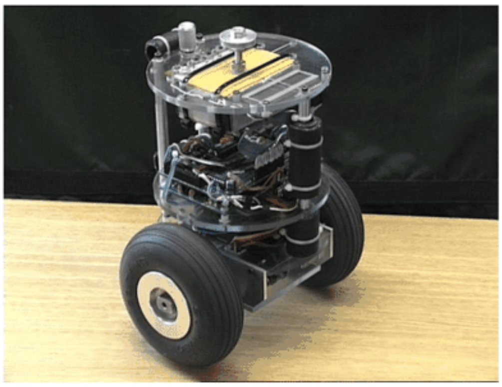

Modeling and structural properties of wheeled mobile robots

# Modeling and structural properties of wheeled mobile robots

Aleksandr Y. Krasnov

## Wheeled mobile robots





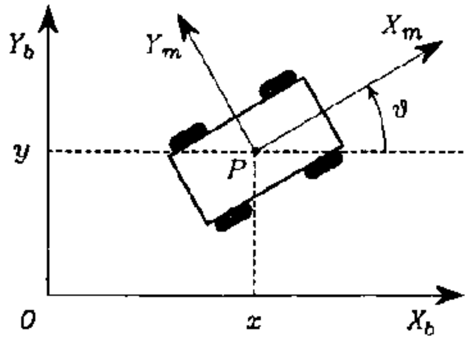
## Four state space models

- The posture kinematic model
- The configuration kinematic model
- The configuration dynamic model
- The posture dynamic model

## Robot description

The robot posture

$$\xi = \begin{bmatrix} x \\ y \\ \vartheta \end{bmatrix}, \qquad (1)$$




Orientation of the base frame with respect to the moving frame

$$R(\vartheta) = \begin{bmatrix} cos\vartheta & sin\vartheta & 0 \\ -sin\vartheta & cos\vartheta & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad (2)$$

## Constrains on different wheels. Fixed or steering wheel.

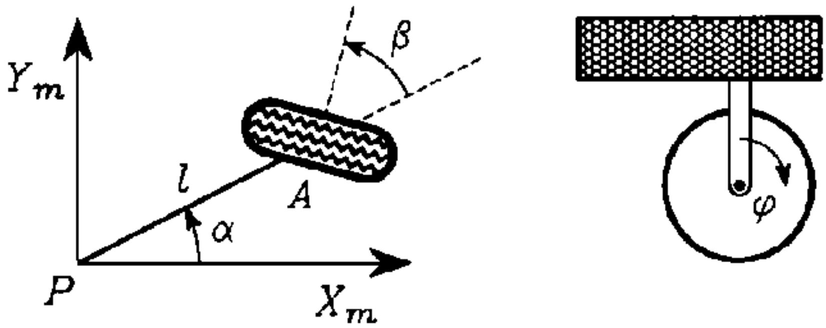



**Figure 1:** Fixed wheel or steering wheel.

Constraint on the wheel plane

$$\begin{bmatrix} -sin(\alpha+\beta) & cos(\alpha+\beta) & lcos\beta \end{bmatrix} R(\vartheta)\dot{\xi} + r\dot{\varphi} = 0; \qquad (3)$$

Constraint orthogonal to the wheel plane

## Constrains on different wheels. Castor wheel.

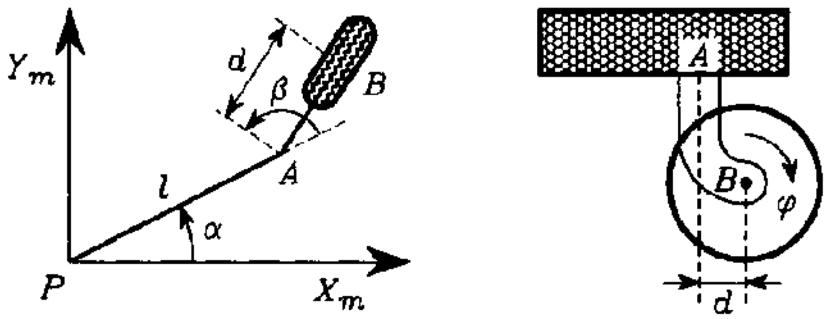



**Figure 2:** Castor wheel.

Constraint on the wheel plane

$$\begin{bmatrix} -sin(\alpha+\beta) & cos(\alpha+\beta) & lcos\beta \end{bmatrix} R(\vartheta)\dot{\xi} + r\dot{\varphi} = 0; \tag{5}$$

Constraint orthogonal to the wheel plane

## Constrains on different wheels. Swedish wheel.

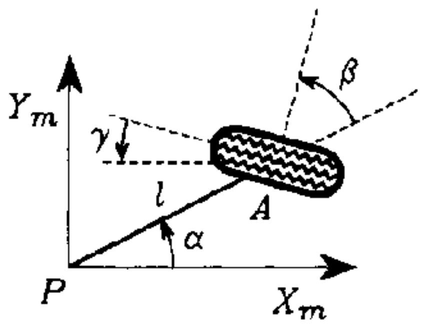



**Figure 3:** Swedish wheel.

The motion constraint

$$\begin{bmatrix} -sin(\alpha+\beta+\gamma) & cos(\alpha+\beta+\gamma) & lcos(\beta+\gamma) \end{bmatrix} R(\vartheta)\dot{\xi} + rcos\gamma\dot{\varphi} = 0. \tag{7}$$

## Restrictions on robot mobility

The configuration of the robot is fully described by the following coordinate vectors:

- posture coordinates $\xi(t) = \begin{bmatrix} x(t) & y(t) & \vartheta(t) \end{bmatrix}^T$ for the position in the plane;
- orientation coordinates $\beta(t) = \begin{bmatrix} \beta_s^T(t) & \beta_c^T(t) \end{bmatrix}^T$ for the orientation angles of the steering and castor wheels, respectively;
- rotation coordinates $\varphi(t) = \begin{bmatrix} \varphi_f(t) & \varphi_s(t) & \varphi_c(t) & \varphi_s w(t) \end{bmatrix}^T$ for the rotation angles of the wheels about their horizontal axle of rotation.

$$J_1(\beta_s, \beta_c)R(\vartheta)\dot{\xi} + J_2\dot{\varphi} = 0, \tag{8}$$

$$C_1(\beta_s, \beta_c)R(\vartheta)\dot{\xi} + C_2\dot{\beta}_c = 0. \tag{9}$$

where $J_1(\beta_s, \beta_c) = \begin{bmatrix} J_{1f} \\ J_{1s}(\beta_s) \\ J_{1c}(\beta_c) \\ J_{1sw} \end{bmatrix}$, $C_1(\beta_s, \beta_c) = \begin{bmatrix} C_{1f} \\ C_{1s}(\beta_s) \\ C_{1c}(\beta_c) \end{bmatrix}$, $C_2 = \begin{bmatrix} 0 \\ 0 \\ C_{2c} \end{bmatrix}$.

## Restrictions on robot mobility

Consider the first $(N_f + N_s)$ non-slipping constrains from (9) and written explicitly as

$$C_{1f}R(\vartheta)\dot{\xi} = 0, \tag{10}$$

$$C_{1s}(\beta_s)R(\vartheta)\dot{\xi} = 0. \tag{11}$$

These constraints imply that the vector $R(\vartheta)\dot{\xi} \in \aleph(C_1^*(\beta_s))$, where

$$C_1^*(\beta_s) = \begin{bmatrix} C_{1f} \\ C_{1s}(\beta_s) \end{bmatrix}. \tag{12}$$

Obviously, it is $rank(C_1^*(\beta_s)) \leq 3$. If $rank(C_1^*(\beta_s)) = 3$, then $R(\vartheta)\dot{\xi} = 0$ and any motion in the plane is impossible.

Define the degree of mobility $\delta_m$ of a mobile robot as

$$\delta_m = dim(\aleph(C_1^*(\beta_s))) = 3 - rank(C_1^*(\beta_s)). \tag{13}$$

## Restrictions on robot mobility

If $rank(C_{1f}) = 2$ the only possible motion is a rotation of the robot about a fixed ICR. Obviously, this limitation is not acceptable in practice and thus we assume that $rank(C_{1f}) \leq 1$. Moreover, we assume that a mobile robot is non-degenerate if

$$rank(C_{1f}) \leq 1 rank(C_1^*(\beta_s)) = rank(C_{1f}) + rank(C_{1s}(\beta_s)) \leq 2.$$

This assumption is equivalent to the following conditions:

- if the robot has more than one fixed wheel ($N_f > 1$), then they are all on a single common axle;
- the centres of the steering wheels do not belong to this common axle of the fixed wheels;
- the number $rank(C_{1s}(\beta_s)) \leq 2$ is the number of steering wheels that can be oriented independently in order to steer the robot.

Define the degree of steerability $\delta_s$ of a mobile robot as

$$\delta_s = rank(C_{1s}(\beta_s)). \tag{14}$$

## Restrictions on robot mobility

It follows that only 5 non-singular structures are of practical interest, which can be inferred by the following conditions.

- The degree of mobility $\delta_m$ satisfies the inequality

$$1 \leq \delta_m \leq 3. \tag{15}$$

- The degree of steerability $\delta_s$ satisfies the inequality

$$0 \leq \delta_s \leq 2. \tag{16}$$

- The following inequality is satisfied:

$$2 \leq \delta_m + \delta_s \leq 3. \tag{17}$$

**Table 1:** Degrees of mobility and steerability for possible wheeled mobile robots.

| $\delta_m$ | 3 | 2 | 2 | 1 | 1 |
|---|---|---|---|---|---|
| $\delta_s$ | 0 | 0 | 1 | 1 | 2 |

## Types of possible wheeled mobile robots

- Type (3, 0) robot. In this case it is

$$\delta_m = dim(\aleph(C_1^*(\beta_s))) = 3 \qquad \delta_s = 0.$$

- Type (2, 0) robot. In this case it is

$$\delta_m = dim(\aleph(C_1^*(\beta_s))) = dim(\aleph(C_{1f})) = 2 \qquad \delta_s = 0.$$

- Type (2, 1) robot. In this case it is

$$\delta_m = dim(\aleph(C_1^*(\beta_s))) = dim(\aleph(C_{1s}(\beta_s))) = 3 \qquad \delta_s = 1.$$

- Type (1, 1) robot. In this case it is

$$\delta_m = dim(\aleph(C_1^*(\beta_s))) = 1 \qquad \delta_s = 1.$$

- Type (1, 2) robot. In this case it is

$$\delta_m = dim(\aleph(C_1^*(\beta_s))) = dim(\aleph(C_{1s}(\beta_s))) = 1 \qquad \delta_s = 2.$$

## Type 3, 0 robot with swedish wheels

The constrains have from (8)

where

$$J_1 = J_{1sw} = \begin{bmatrix} -\sqrt{3}/2 & 1/2 & L \\ 0 & -1 & L \\ \sqrt{3}/2 & 1/2 & L \end{bmatrix},$$

$$J_2 = \begin{bmatrix} r & 0 & 0 \\ 0 & r & 0 \\ 0 & 0 & r \end{bmatrix}.$$

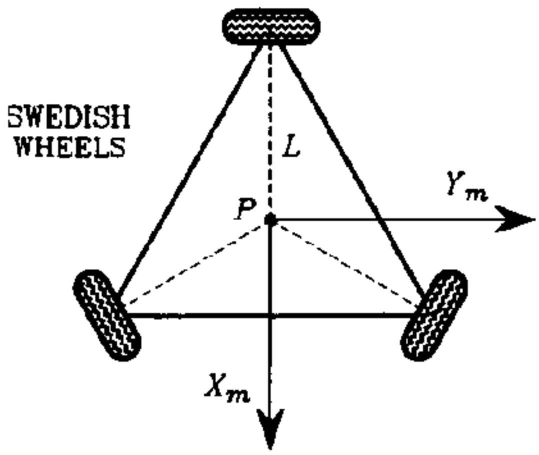



**Table 2:** Characteristic constants of type 3, 0 robot with swedish wheels.

| Wheels | $\alpha$ | $\beta$ | $\gamma$ | $l$ |
|---|---|---|---|---|
| $1sw$ | $\pi/3$ | 0 | 0 | $L$ |
| $2sw$ | $\pi$ | 0 | 0 | $L$ |
| $3sw$ | $5\pi/3$ | 0 | 0 | $L$ |

## Type 3, 0 robot with castor wheels

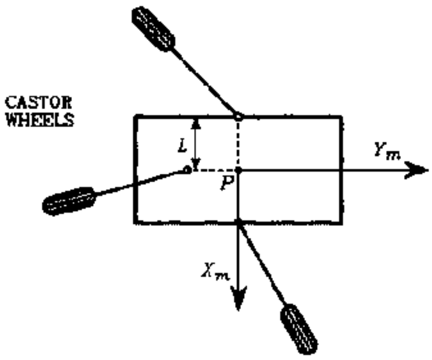
The constrains have from (8) and (9) where

$$J_1 = J_{1c}(\beta_c) = \begin{bmatrix} -sin\beta_{c1} & cos\beta_{c1} & Lcos\beta_{c1} \\ sin\beta_{c2} & -cos\beta_{c2} & Lcos\beta_{c2} \\ cos\beta_{c3} & sin\beta_{c3} & Lcos\beta_{c3} \end{bmatrix}; J_2 = \begin{bmatrix} r & 0 & 0 \\ 0 & r & 0 \\ 0 & 0 & r \end{bmatrix}.$$

$$C_1 = C_{1c}(\beta_c) = \begin{bmatrix} cos\beta_{c1} & sin\beta_{c1} & d + Lsin\beta_{c1} \\ -cos\beta_{c2} & -sin\beta_{c2} & d + Lsin\beta_{c2} \\ sin\beta_{c3} & -cos\beta_{c3} & d + Lsin\beta_{c3} \end{bmatrix}, C_{2c} = \begin{bmatrix} d & 0 & 0 \\ 0 & d & 0 \\ 0 & 0 & d \end{bmatrix}.$$

**Table 3:** Characteristic constants of type 3, 0 robot with castor wheels.

| Wheels | $\alpha$ | $\beta$ | $l$ |
|---|---|---|---|
| $1c$ | $0$ | - | $L$ |
| $2c$ | $\pi$ | - | $L$ |
| $3c$ | $3\pi/2$ | - | $L$ |




## Type 2, 0 robot

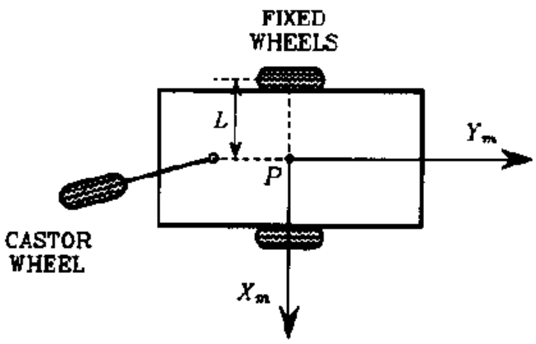
The constrains have from (8) and (9) where

$$J_1 = \begin{bmatrix} J_{1f} \\ J_{1c}(\beta_{c3}) \end{bmatrix} = \begin{bmatrix} 0 & 1 & L \\ 0 & -1 & L \\ cos\beta_{c3} & sin\beta_{c3} & Lcos\beta_{c3} \end{bmatrix}; J_2 = \begin{bmatrix} r & 0 & 0 \\ 0 & r & 0 \\ 0 & 0 & r \end{bmatrix}.$$

$$C_1 = \begin{bmatrix} C_{1f} \\ C_{1c}(\beta_{c3}) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 0 & 0 \\ sin\beta_{c3} & -cos\beta_{c3} & d + Lsin\beta_{c3} \end{bmatrix}, C_2 = \begin{bmatrix} 0 \\ C_{2c} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ d \end{bmatrix}.$$

**Table 4:** Characteristic constants of type 2, 0 robot.

| Wheels | $\alpha$ | $\beta$ | $l$ |
|---|---|---|---|
| $1f$ | $0$ | $0$ | $L$ |
| $2f$ | $\pi$ | $0$ | $L$ |
| $3c$ | $3\pi/2$ | - | $L$ |



## Type 2, 1 robot

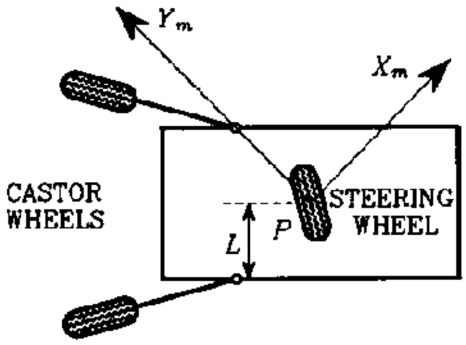
The constrains have from (8) and (9) where

$$J_1 = \begin{bmatrix} J_{1s}(\beta_{s1}) \\ J_{1c}(\beta_{c2}, \beta_{c3}) \end{bmatrix} = \begin{bmatrix} -sin\beta_{s1} & cos\beta_{s1} & 0 \\ -cos\beta_{c2} & -sin\beta_{c2} & \sqrt{2}Lcos\beta_{c2} \\ sin\beta_{c3} & cos\beta_{c3} & \sqrt{2}Lcos\beta_{c3} \end{bmatrix}, J_2 = \begin{bmatrix} r & 0 & 0 \\ 0 & r & 0 \\ 0 & 0 & r \end{bmatrix}.$$

$$C_1 = \begin{bmatrix} C_{1s}(\beta_{s1}) \\ C_{1c}(\beta_{c2}, \beta_{c3}) \end{bmatrix} = \begin{bmatrix} cos\beta_{s1} & sin\beta_{s1} & 0 \\ -sin\beta_{c2} & cos\beta_{c2} & d+\sqrt{2}Lsin\beta_{c2} \\ -cos\beta_{c3} & -sin\beta_{c3} & d+\sqrt{2}Lsin\beta_{c3} \end{bmatrix}, C_2 = \begin{bmatrix} 0 \\ C_{2c} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ d & 0 \\ 0 & d \end{bmatrix}.$$

**Table 5:** Characteristic constants of type 2, 1 robot.

| Wheels | $\alpha$ | $\beta$ | $l$ |
|---|---|---|---|
| 1s | 0 | - | 0 |
| 2c | $\pi/2$ | - | $L\sqrt{2}$ |
| 3c | $\pi$ | - | $L\sqrt{2}$ |




## Type 1, 1 robot

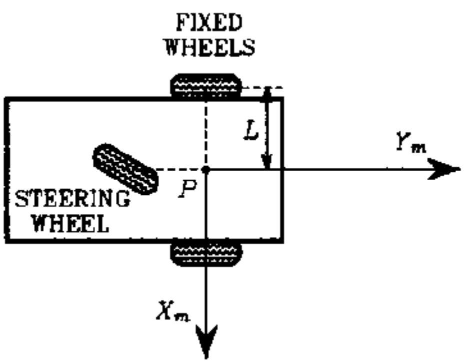
The constrains have from (8) and (9) where

$$J_1 = \begin{bmatrix} J_{1f} \\ J_{1s}(\beta_{s3}) \end{bmatrix} = \begin{bmatrix} 0 & 1 & L \\ 0 & -1 & L \\ cos\beta_{s3} & sin\beta_{s3} & Lcos\beta_{s3} \end{bmatrix}, J_2 = \begin{bmatrix} r & 0 & 0 \\ 0 & r & 0 \\ 0 & 0 & r \end{bmatrix}.$$

$$C_1 = \begin{bmatrix} C_{1f} \\ C_{1s}(\beta_{c3}) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 0 & 0 \\ sin\beta_{s3} & -cos\beta_{s3} & Lsin\beta_{s3} \end{bmatrix}, C_2 = 0.$$

**Table 6:** Characteristic constants of type 1, 1 robot.

| Wheels | $\alpha$ | $\beta$ | $l$ |
|---|---|---|---|
| 1f | 0 | 0 | $L$ |
| 2f | $\pi$ | 0 | $L$ |
| 3s | $3\pi/2$ | - | $L$ |



## Type 1, 2 robot

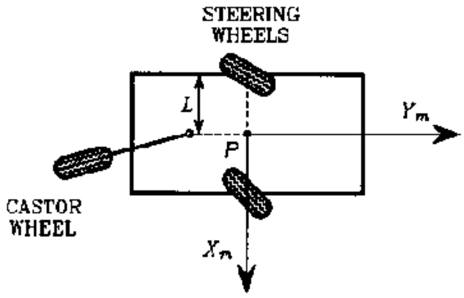
The constrains have from (8) and (9) where

$$J_1 = \begin{bmatrix} J_{1s}(\beta_{s1}, \beta_{s2}) \\ J_{1c}(\beta_{c3}) \end{bmatrix} = \begin{bmatrix} -sin\beta_{s1} & cos\beta_{s1} & Lcos\beta_{s1} \\ sin\beta_{s2} & -cos\beta_{s2} & Lcos\beta_{s2} \\ cos\beta_{c3} & sin\beta_{c3} & Lcos\beta_{c3} \end{bmatrix}, J_2 = \begin{bmatrix} r & 0 & 0 \\ 0 & r & 0 \\ 0 & 0 & r \end{bmatrix}.$$

$$C_1 = \begin{bmatrix} C_{1s}(\beta_{s1}, \beta_{s2}) \\ C_{1c}(\beta_{c3}) \end{bmatrix} = \begin{bmatrix} cos\beta_{s1} & sin\beta_{s1} & Lsin\beta_{s1} \\ -cos\beta_{s2} & -sin\beta_{s2} & Lsin\beta_{s2} \\ sin\beta_{c3} & -cos\beta_{c3} & d + Lsin\beta_{c3} \end{bmatrix}, C_2 = \begin{bmatrix} 0 \\ C_{2c} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ d \end{bmatrix}.$$

**Table 7:** Characteristic constants of type 1, 2 robot.

| Wheels | $\alpha$ | $\beta$ | $l$ |
|---|---|---|---|
| 1s | 0 | - | $L$ |
| 2s | $\pi$ | - | $L$ |
| 3c | $3\pi/2$ | - | $L$ |




## Posture kinematic model

Whatever the type of mobile robot, the velocity $\dot{\xi}(t)$ is restricted to belong to a distribution $\Delta_c$ defined as

$$\dot{\xi}(t) \in \Delta_c = span\{col(R^T(\vartheta)\Sigma(\beta_s))\}\forall t,$$

where columns of matrix $\Sigma(\beta_s)$ form a basis of $\aleph(C_1^*(\beta_s))$, i.e.

$$\aleph(C_1^*(\beta_s)) = span\{col(\Sigma(\beta_s)).$$

This is equivalent to the following statement: for all $t$, there exists a time-varying vector $\eta(t)$ such that

$$\dot{\xi} = R^T(\vartheta)\Sigma(\beta_s)\eta. \qquad (18)$$

The dimension of the vector $\eta(t)$ is the degree of mobility (13) of the robot.

## Posture kinematic model

If robot has no steering wheels ($\delta_s = 0$), the matrix $\Sigma$ is constant and the expression (18) reduces to

$$\dot{\xi} = R^T(\vartheta)\Sigma\eta. \tag{19}$$

In the opposite case ($\delta_s \geq 1$), the matrix $\Sigma$ explicitly depends on the orientation coordinates $\beta_s$ and the expression (18) can be augmented as follows:

$$\dot{\xi} = R^T(\vartheta)\Sigma(\beta_s)\eta. \tag{20}$$

$$\dot{\beta}_s = \zeta. \tag{21}$$

The kinematic state space model is in fact only a subsystem of general dynamic model that will be discussed further.

## Generic models of wheeled robots

- **Type** $(3, 0)$ **robot.** The matrix $\Sigma$ can always be chosen as a $(3 \times 3)$ identity matrix, so the equation (19) reduces to

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\vartheta} \end{bmatrix} = \begin{bmatrix} cos\vartheta & -sin\vartheta & 0 \\ sin\vartheta & cos\vartheta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \eta_1 \\ \eta_2 \\ \eta_3 \end{bmatrix}. \tag{22}$$

- **Type** $(2, 0)$ **robot.** The matrix $\Sigma$ is selected as $\Sigma = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$, so the equation (19) reduces to

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\vartheta} \end{bmatrix} = \begin{bmatrix} -sin\vartheta & 0 \\ cos\vartheta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix}. \tag{23}$$

## Generic models of wheeled robots

- **Type** $(2,1)$ **robot.** The matrix $\Sigma(\beta_s)$ is selected as

$$\Sigma(\beta_s) = \begin{bmatrix} sin\beta_{s1} & 0 \\ cos\beta_{s1} & 0 \\ 0 & 1 \end{bmatrix},$$

so the equations (20) and (21) reduces to

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\vartheta} \end{bmatrix} = \begin{bmatrix} -sin(\vartheta+\beta_{s2}) & 0 \\ cos(\vartheta+\beta_{s1}) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix}, \tag{24}$$

$$\dot{\beta}_{s3} = \zeta_1. \tag{25}$$

## Generic models of wheeled robots

- **Type** $(1,1)$ **robot.** The matrix $\Sigma(\beta_s)$ is selected as

$$\Sigma(\beta_s) = \begin{bmatrix} 0 \\ Lsin\beta_{s3} \\ cos\beta_{s3} \end{bmatrix},$$

so the equations (20) and (21) reduces to

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\vartheta} \end{bmatrix} = \begin{bmatrix} -Lsin\vartheta sin\beta_{s3} \\ Lcos\vartheta sin\beta_{s3} \\ cos\beta_{s3} \end{bmatrix} \eta_1, \tag{26}$$

$$\dot{\beta}_{s3} = \zeta_1. \tag{27}$$

## Generic models of wheeled robots

- **Type (1, 2) robot.** The matrix $\Sigma(\beta_s)$ is selected as

$$\Sigma(\beta_s) = \begin{bmatrix} -2Lsin\beta_{s1}sin\beta_{s2} \\ Lsin(\beta_{s1} + \beta_{s2}) \\ sin(\beta_{s2} - \beta_{s1}) \end{bmatrix},$$

so the equations (20) and (21) reduces to

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\vartheta} \end{bmatrix} = \begin{bmatrix} -L(sin\beta_{s1}sin(\vartheta + \beta_{s2}) + sin\beta_{s2}sin(\vartheta + \beta_{s1})) \\ L(sin\beta_{s1}cos(\vartheta + \beta_{s2}) + sin\beta_{s2}cos(\vartheta + \beta_{s1})) \\ sin(\beta_{s2} - \beta_{s1}) \end{bmatrix} \eta_1, \quad (28)$$

$$\dot{\beta}_{s1} = \zeta_1. \quad (29)$$

$$\dot{\beta}_{s2} = \zeta_2. \quad (30)$$

## Mobility, steerability and manoeuvrability

Rewrite the posture kinematic model in the compact form

$$\dot{z} = B(z)u, \quad (31)$$

where either $(\delta_s)$

$$z = \xi, \quad B(z) = R^T(\vartheta)\Sigma, \quad u = \eta$$

or $(\delta_s \geq 1)$

$$z = \begin{bmatrix} \xi \\ \beta_s \end{bmatrix}, \quad B(z) = \begin{bmatrix} R^T(\vartheta)\Sigma(\beta_s) & 0 \\ 0 & I \end{bmatrix}, \quad u = \begin{bmatrix} \eta \\ \zeta \end{bmatrix}.$$

Consider degree of manoeuvrability

$$\delta_M = \delta_m + \delta_s.$$

The ideal situation is that of omnidirectional mobile robots where $\delta_m = \delta_M = 3$.

## Irreducibility

A well-known consequence of Frobenius theorem is that the system is reducible only if $dim(\bar{\Delta}) < dim(z)$, where $\bar{\Delta}$ is the involutive closure of the following distribution $\Delta$, expressed in local coordinates as $\Delta(z) = span\{col(B(z))\}$.

For the posture kinematic model (31) of a wheeled mobile robot, the input matrix $B(z)$ has full rank, i.e.

$$rank(B(z)) = \delta_m + \delta_s \quad \forall z,$$

and the involutive distribution $\bar{\Delta}(z)$ has constant maximal dimension, i.e.

$$dim(\bar{\Delta}(z)) = 3 + \delta_s \quad \forall z.$$

As a consequence, the posture kinematic model (31) of a wheeled mobile robot is irreducible. This is a coordinate-free property.

## Controllability

The controllability rank of the linear approximation of the posture kinematic model (31) around an equilibrium configuration $\bar{z} = \begin{bmatrix} \bar{\xi}^T & \bar{\beta}_s^T \end{bmatrix}^T$ is $\delta_m + \delta_s$.

This property follows from the fact that the linear approximation around $(\bar{z} = 0, \bar{u} = 0)$ can be written as

$$\frac{d}{dt}(z - \bar{z}) = B(\bar{z})u.$$

It follows that the controllability matrix reduces to $B(\bar{z})$ whose rank is $\delta_m + \delta_s$ for all $\bar{z}$ as was shown before.

This implies that the posture kinematic model (31) of a wheeled mobile robot is controllable (completely controllable for type $(3, 0)$ robots).

## Stabilizability

For omnidirectional robots feedback control

$$u(z) = B^{-1}(z)A(z - z^*),$$

with $A$ an arbitrary Hurwitz matrix is clearly a linearizing smooth feedback control law that drives robot exponentially to $z^*$. Indeed, the closed loop is described by the freely assignable linear dynamics

$$\frac{d}{dt}(z - z^*) = A(z - z^*).$$

Hence, omnidirectional mobile robots are full state feedback linearizable.

For restricted mobility robots the posture kinematic model (31) is not stabilizable by a continuous static time-invariant state feedback $u(z)$, but is stabilizable by a continuous time-varying static state feedback $u(z, t)$.

## Configuration kinematic model

From (8) and (9) it follows directly that

$$\dot{\beta}_c = -C_{2c}^{-1} C_{1c}(\beta_c) R(\vartheta)\dot{\xi}, \qquad (32)$$

$$\dot{\varphi} = -J_2^{-1} J_1(\beta_s, \beta_c) R(\vartheta)\dot{\xi}. \qquad (33)$$

By combining with the posture kinematic model (20), equations (32) and (33) become

$$\dot{\beta}_c = D(\beta_c)\Sigma(\beta_s)\eta, \qquad (34)$$

$$\dot{\varphi} = E(\beta_s, \beta_c)\Sigma(\beta_s)\eta, \qquad (35)$$

where $D(\beta_c) = -C_{2c}^{-1} C_{1c}(\beta_c)$ and $E(\beta_s, \beta_c) = -J_2^{-1} J_1(\beta_s, \beta_c)$.

## Configuration kinematic model

Define the configuration kinematic model as

$$\dot{q} = S(q)u, \tag{36}$$

where

$$q = \begin{bmatrix} \xi \\ \beta_s \\ \beta_c \\ \varphi \end{bmatrix}, S(q) = \begin{bmatrix} R^T(\vartheta)\Sigma(\beta_s) & 0 \\ 0 & I \\ D(\beta_c)\Sigma(\beta_s) & 0 \\ E(\beta_s, \beta_c)\Sigma(\beta_s) & 0 \end{bmatrix}, q = \begin{bmatrix} \eta \\ \zeta \end{bmatrix}.$$

## Configuration kinematic model

Reducibility of (36) is directly related to the dimension of the involutive closure of the distribution $\Delta_1(q) = span\{col(S(q))\}$. It follows immediately that

$$\delta_m + N_s = dim(\Delta_1) \leq dim(inv(\Delta_1)) \leq dim(q) = 3 + N + N_c + N_s.$$

Define the degree of nonholonomy $M$ of a mobile robot as

$$M = dim(inv(\Delta_1)) - (\delta_m + N_s). \tag{37}$$

The configuration kinematic model (36) of all types of wheeled mobile robot is nonholonomic, i.e. $M > 0$, but is reducible, i.e. $dim(q) > dim(inv(\Delta_1))$.

## Configuration kinematic model for type $(3,0)$ robot

For this robot $\delta_m = 3$ and the configuration coordinates are
$q = \begin{bmatrix} x & y & \vartheta & \varphi_1 & \varphi_2 & \varphi_3 \end{bmatrix}^T$.

The configuration model is characterised by

$$S(q) = \begin{bmatrix} cos\vartheta & -sin\vartheta & 0 \\ sin\vartheta & cos\vartheta & 0 \\ 0 & 0 & 1 \\ \sqrt{3}/2r & -1/2r & -L/r \\ 0 & 1/r & L/r \\ -\sqrt{3}/2r & -1/2r & -L/r \end{bmatrix}.$$

It is easy to check that $dim(\Delta_1) = 3$ and $dim(inv(\Delta_1)) = 5$. The structure of the configuration model implies that

$$\dot{\varphi}_1 + \dot{\varphi}_2 + \dot{\varphi}_3 = -\frac{3L}{r}\dot{\vartheta}.$$

## Configuration kinematic model for type $(2,0)$ robot

For this robot $\delta_m = 2$ and the configuration coordinates are
$q = \begin{bmatrix} x & y & \vartheta & \beta_{c3} & \varphi_1 & \varphi_2 & \varphi_3 \end{bmatrix}^T$.

The configuration model is characterised by

$$S(q) = \begin{bmatrix} -sin\vartheta & 0 \\ cos\vartheta & 0 \\ 0 & 1 \\ \frac{1}{d}cos\beta_{c3} & -\frac{1}{d}(d + Lsin\beta_{c3}) \\ -1/r & -L/r \\ 1/r & -L/r \\ -\frac{1}{r}sin\beta_{c3} & -\frac{L}{r}cos\beta_{c3} \end{bmatrix}.$$

It can be checked that $dim(\Delta_1) = 2$ and $dim(inv(\Delta_1)) = 6$. From the the configuration model it is

$$\dot{\varphi}_1 + \dot{\varphi}_2 = -\frac{2L}{r}\dot{\vartheta}.$$

## Model derivation

Using the Lagrange formulation, the dynamics of wheeled mobile robots is described by the following $(3 + N_c + N + N_s)$ Lagrange's equations:

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial T}{\partial \dot{\xi}}\right)^T - \left(\frac{\partial T}{\partial \xi}\right)^T = R^T(\vartheta)J_1^T(\beta_s, \beta_c)\lambda + R^T(\vartheta)C_1^T(\beta_s, \beta_c)\mu, \quad (38)$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial T}{\partial \dot{\beta}_c}\right)^T - \left(\frac{\partial T}{\partial \beta_c}\right)^T = C_2^T\mu + \tau_c, \quad (39)$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial T}{\partial \dot{\varphi}}\right)^T - \left(\frac{\partial T}{\partial \varphi}\right)^T = J_2^T\lambda + \tau_\varphi, \quad (40)$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial T}{\partial \dot{\beta}_s}\right)^T - \left(\frac{\partial T}{\partial \beta_s}\right)^T = \tau_s, \quad (41)$$

where $T$ represents the kinetic energy and $\lambda$, $\mu$ are the Lagrange multipliers associated with the constraints (8) and (9) respectively.

## Model derivation

By multiplying (38), (39) and (40) by $\Sigma^T(\beta_s)R(\vartheta)$, $\Sigma^T(\beta_s)D(\beta_c)$ and $\Sigma^T(\beta_s)E(\beta_s, \beta_c)$ respectively and summing them up one can obtain

$$\begin{aligned}\Sigma^T(\beta_s)R(\vartheta)[T]_\xi + D(\beta_c)[T]_{\beta_c} + E(\beta_s, \beta_c)[T]_\varphi = \\ = \Sigma^T(\beta_s)(D^T(\beta_c)\tau_c + E^T(\beta_s, \beta_c)\tau_\varphi),\end{aligned} \quad (42)$$

$$[T]_{\beta_s} = \tau_s, \quad (43)$$

where

$$[T]_\psi = \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial T}{\partial \dot{\psi}}\right)^T - \left(\frac{\partial T}{\partial \psi}\right)^T.$$

The kinetic energy of wheeled mobile robots can be expressed as follows:

$$T = \dot{\xi}^T R^T(\vartheta)(M(\beta_c)R(\vartheta)\dot{\xi} + 2V(\beta_c)\dot{\beta}_c + 2W\dot{\beta}_s) + \dot{\beta}_c^T I_c\dot{\beta}_c + \dot{\varphi}^T I_\varphi\dot{\varphi} + \dot{\beta}_s^T I_s\dot{\beta}_s.$$

## Configuration dynamic model

The configuration dynamic model of wheeled mobile robots in the state space takes on the following general form:

$$\dot{\xi} = R^T(\vartheta)\Sigma(\beta_s)\eta, \tag{44}$$

$$\dot{\beta}_s = \zeta, \tag{45}$$

$$\dot{\beta}_c = D(\beta_c)\Sigma(\beta_s)\eta, \tag{46}$$

$$H_1(\beta_s, \beta_c)\dot{\eta} + \Sigma^T(\beta_s)V(\beta_c)\dot{\zeta} + f_1(\beta_s, \beta_c, \eta, \zeta) = \\ = \Sigma^T(\beta_s)(D^T(\beta_c)\tau_c + E^T(\beta_s, \beta_c)\tau_\varphi), \tag{47}$$

$$V^T(\beta_c)\Sigma(\beta_s)\dot{\eta} + I_s\dot{\zeta} + f_2(\beta_s, \beta_c, \eta, \zeta) = \tau_s, \tag{48}$$

$$\dot{\varphi} = E(\beta_c, \beta_s)\Sigma(\beta_s)\eta, \tag{49}$$

where $H_1(\beta_s, \beta_c) - \Sigma^T(\beta_s)(M(\beta_c) + D^T(\beta_c)V^T(\beta_c) + V(\beta_c)D(\beta_c) + D^T(\beta_c)I_cD(\beta_c) + E^T(\beta_s, \beta_c)I_\varphi E(\beta_s, \beta_c))\Sigma(\beta_s)$.

## Actuator configuration

All steering wheels must be provided with an actuator for their orientation, and to ensure a full robot mobility $N_m$ additional actuators for either the rotation of some wheels or the orientation of some castor wheels.

$$\begin{bmatrix} \tau_c \\ \tau_\varphi \end{bmatrix} - P\tau_m, \tag{50}$$

where $P$ is an $((N_c + N) \times N_m)$ elementary matrix which selects the components of $\tau_c$ and $\tau_\varphi$ that are effectively used as control inputs.

Using (50) we can recognize that (47) becomes

$$H_1(\beta_s, \beta_c)\dot{\eta} + \Sigma^T(\beta_s)V(\beta_c)\dot{\zeta} + f_1(\beta_s, \beta_c, \eta, \zeta) = B(\beta_s, \beta_c)P\tau_m, \tag{51}$$

where $B(\beta_s, \beta_c) = \Sigma^T(\beta_s)\begin{bmatrix} D^T(\beta_c) & E^T(\beta_s, \beta_c) \end{bmatrix}$.

The actuator configuration is such that the matrix $B(\beta_s, \beta_c)P$ has full rank for all $(\beta_s, \beta_c) \in R^{N_s+N_c}$.

## Actuator configuration for type $(3, 0)$ robot

In case of swedish wheels the matrix $B$ is constant and nonsingular, so the only admissible configuration is to equip each wheel with an actuator.

In case of castor wheels the matrix $B(\beta_c)$ is

$$B(\beta_c) = \Sigma^T \begin{bmatrix} D^T(\beta_c) & E^T(\beta_c) \end{bmatrix}$$

with

$$\Sigma^T D^T(\beta_c) = -\frac{1}{d}\begin{bmatrix} cos\beta_{c1} & cos\beta_{c2} & sin\beta_{c3} \\ sin\beta_{c1} & -sin\beta_{c2} & -cos\beta_{c3} \\ d + Lsin\beta_{c1} & d + Lsin\beta_{c2} & d + Lsin\beta_{c3} \end{bmatrix},$$

$$\Sigma^T E^T(\beta_c) = -\frac{1}{r}\begin{bmatrix} -sin\beta_{c1} & sin\beta_{c2} & cos\beta_{c3} \\ cos\beta_{c1} & -cos\beta_{c2} & sin\beta_{c3} \\ Lcos\beta_{c1} & Lcos\beta_{c2} & Lcos\beta_{c3} \end{bmatrix}.$$

## Actuator configuration for type $(2, 0)$ robot

For this robot the matrix $B(\beta_c)$ is

$$B(\beta_{c3}) = \begin{bmatrix} \frac{1}{d}cos\beta_{c3} & -\frac{1}{r} & \frac{1}{r} & -\frac{1}{r}sin\beta_c \\ -\frac{1}{d}(d + Lsin\beta_{c3}) & -\frac{L}{r} & -\frac{L}{r} & -\frac{L}{r}cos\beta_{c3} \end{bmatrix}.$$

Several configurations with 2 actuators is admissible: 2 rotation actuators on wheels 1 and 2 with $P = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$; 1 actuator for the orientation of wheel 3 and 1 actuator for the rotation of wheel 2 (or 3), provided that $d > L$ with $P = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$; 2 actuators (orientation and rotation) on castor wheel 3, provided that $d < L$ with $P = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}$.

## Actuator configuration for type $(2, 1)$ robot

For this robot we first need an orientation actuator for the steering wheel. The matrix $B(\beta_s, \beta_c)$ is then

$$B(\beta_s, \beta_c) = \Sigma^T(\beta_s)\begin{bmatrix} D^T(\beta_c) & E^T(\beta_s, \beta_c)\end{bmatrix}$$

with

$$\Sigma^T(\beta_s) = \begin{bmatrix} -sin\beta_{s1} & cos\beta_{s1} & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

$$D^T(\beta_c) = -\frac{1}{d}\begin{bmatrix} -sin\beta_{c2} & -cos\beta_{c3} \\ cos\beta_{c2} & -sin\beta_{c3} \\ d + \sqrt{2}Lsin\beta_{c2} & d + \sqrt{2}Lsin\beta_{c3} \end{bmatrix},$$

$$E^T(\beta_s, \beta_c) = -\frac{1}{r}\begin{bmatrix} cos\beta_{s1} & -sin\beta_{c2} & -cos\beta_{c3} \\ sin\beta_{s1} & cos\beta_{c2} & -sin\beta_{c3} \\ 0 & d + \sqrt{2}Lsin\beta_{c2} & d + \sqrt{2}Lsin\beta_{c3} \end{bmatrix}.$$

## Actuator configuration for type $(2, 1)$ robot

Hence two admissible actuator configurations are obtained by using a second actuator for the rotation of the steering wheel (number 1) and a third actuator for the orientation of either wheel 2 or wheel 3. The two corresponding matrices $P$ are:

$$P = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad P = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

## Actuator configuration for type $(1, 1)$ robot

For this robot we first need an orientation actuator for the steering wheel. The matrix $B(\beta_s)$ reduces to the vector

$$B = -\frac{L}{r}\begin{bmatrix} sin\beta_{s3} + cos\beta_{s3} & -sin\beta_{s3} + cos\beta_{s3} & 1 \end{bmatrix}.$$

Since $\delta_m = 1$ a second actuator should be provided for the rotation of the third wheel. The matrix $P$ is then

$$P = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

## Actuator configuration for type $(1, 2)$ robot

For this robot we first need 2 orientation actuators for 2 steering wheels. The matrix $B(\beta_s, \beta_c)$ is then

$$B(\beta_s, \beta_c) = \Sigma^T(\beta_s)\begin{bmatrix} D^T(\beta_c) & E^T(\beta_s, \beta_c) \end{bmatrix}$$

with

$$\Sigma^T(\beta_s) = \begin{bmatrix} -2Lsin\beta_{s1}sin\beta_{s2} & Lsin(\beta_{s1} + \beta_{s2}) & sin(\beta_{s2} - \beta_{s1}) \end{bmatrix},$$

$$D^T(\beta_c) = \begin{bmatrix} -\frac{1}{d}sin\beta_{c3} \\ \frac{1}{d}cos\beta_{c3} \\ -\frac{1}{d}(d + Lsin\beta_{c3}) \end{bmatrix},$$

$$E^T(\beta_s, \beta_c) = \frac{1}{r}\begin{bmatrix} -sin\beta_{s1} & sin\beta_{s2} & cos\beta_{c3} \\ cos\beta_{s1} & -cos\beta_{s2} & sin\beta_{c3} \\ Lcos\beta_{s1} & Lcos\beta_{s2} & Lcos\beta_{c3} \end{bmatrix}.$$

## Actuator configuration for type $(1,2)$ robot

Since $\delta_m = 1$, it would be sufficient to have one column of $B(\beta_s, \beta_c)$ being nonzero for all possible configurations. However, there is no such a column. It is therefore necessary to use 2 additional actuators, for instance for the rotation of wheels 1 and 2 giving the matrix $P$ as

$$P = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}.$$

## Posture dynamic model

The configuration dynamic model in compact form

$$\dot{q} = S(q)u, \tag{52}$$

$$H(\beta)\dot{u} + f(\beta, u) = F(\beta)\tau_0, \tag{53}$$

where $\beta = \begin{bmatrix} \beta_s \\ \beta_c \end{bmatrix}$, $q = \begin{bmatrix} \xi \\ \beta \\ \varphi \end{bmatrix}$, $u = \begin{bmatrix} \eta \\ \zeta \end{bmatrix}$,

$$H(\beta) = \begin{bmatrix} H_1(\beta_s, \beta_c) & \Sigma^T(\beta_s)V(\beta_c) \\ V^T(\beta_{0c})\Sigma(\beta_s) & I_s \end{bmatrix}, \; f(\beta, u) = \begin{bmatrix} f_1(\beta_s . \beta_c, \eta, \zeta) \\ f_2(\beta_s . \beta_c, \eta, \zeta) \end{bmatrix},$$

$$F(\beta) = \begin{bmatrix} B(\beta_s, \beta_c) & 0 \\ 0 & I \end{bmatrix}, \; \tau_0 = \begin{bmatrix} \tau_m \\ \tau_s \end{bmatrix}.$$

## Posture dynamic model

The configuration dynamic model (52)-(53) is feedback equivalent (by a smooth static time-invariant state feedback) to the following system:

$$\dot{q} = S(q)u, \tag{54}$$

$$\dot{u} = v, \tag{55}$$

where $v$ represents a set of $\delta_m$ auxiliary control inputs.

The following smooth static time-invariant state feedback is well defined everywhere in the state space, i.e.

$$\tau_0 = F^{\dagger}(\beta)(H(\beta)\dot{u} - f(\beta, u)), \tag{56}$$

where $F^{\dagger}$ denotes an arbitrary left inverse of $F(\beta, u)$.

## Posture dynamic model

We restrict our attention to the following posture dynamic model:

$$\dot{z} - B(z)u, \tag{57}$$

$$\dot{u} = v, \tag{58}$$

where $z = \begin{bmatrix}\xi^T & \beta_s^T\end{bmatrix}^T$ and $u = \begin{bmatrix}\eta^T & \zeta^T\end{bmatrix}^T$.

The coordinates $\beta_c$ and $\varphi$ have apparently disappeared but it is important to notice that they are in fact hidden in the feedback (56).

The posture dynamic model is generic and irreducible, and small-time-locally-controllable; further, for restricted mobility robots, it is not stabilizable by a continuous static time-invariant state feedback, but is stabilizable by a time-varying static state feedback.

**Digital control systems**

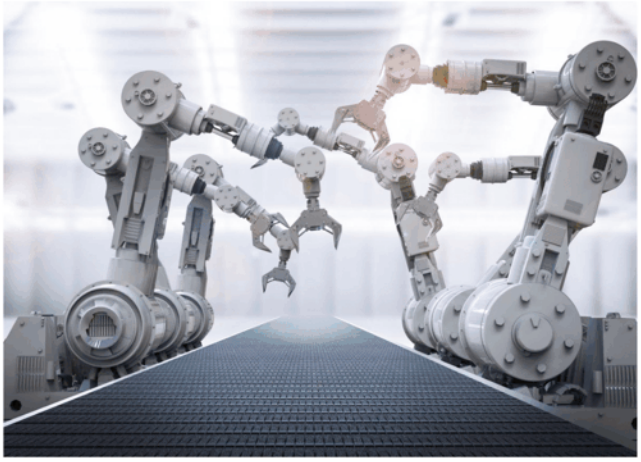
Digital and microcontroller devices

# Digital and Microcontroller Devices

Vlasov Sergei

## Robots, what is it?

# Robots, what is it?

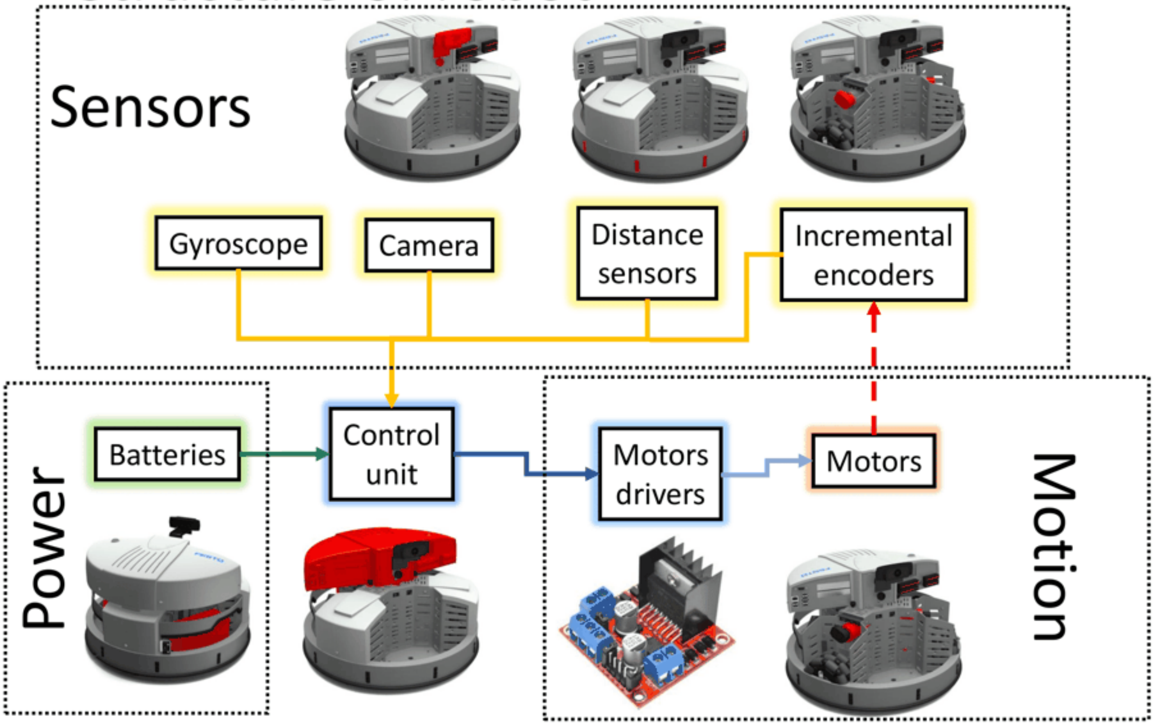


# Structure of robot

Hardware

| Control | Drive systems | Sensors | Interfaces | Supply |
|---|---|---|---|---|
| → Power switch | → Omnidrive | → Bumper | → WLAN | → Batteries |
| → Control unit | → Motors | → Distance sensors | → I/O-Interfaces | → Power supply unit |
| → Embedded PC | → Incremental encoder | → Gyroscope | → Motor/encoder | → Charging electronics |
| → Microcontroller | → Gear units | → Camera | → USB | → Pedestal |
| → Reset button | → Wheels | → Opto-electronic sensors | → PCI Express | |
| | | → Inductive sensors | → Ethernet | |
| | | | → VGA | |

## Structure of robot




## Power supply

- Primary Batteries
- Secondary Batteries
  - Lithium (Li-ion, Li-pol)
  - Nickel Cadmium (Ni-Cd)
  - Nickel-Metal Hydride (Ni-MH)
  - Lead-Acid

Schematic symbols

Single cell Multi-cell

# Power supply

**Terminology**

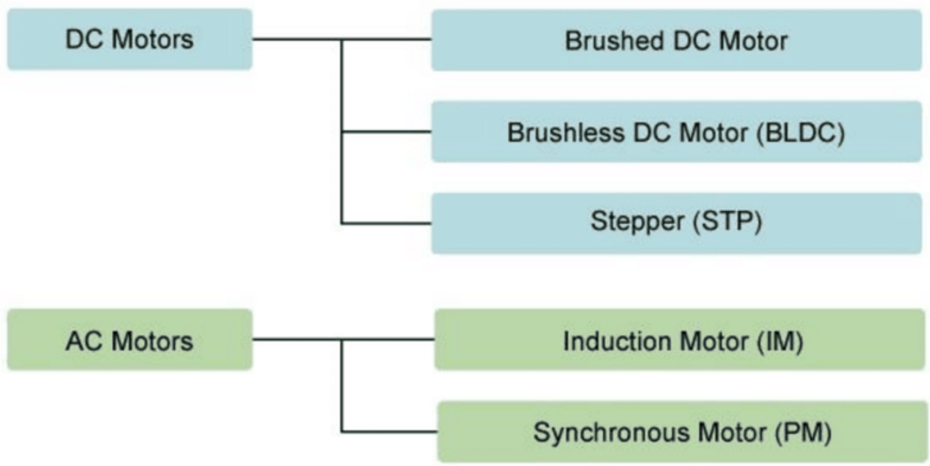
- **Capacity** - Batteries have different ratings for the amount of power a given battery can store. When a battery is fully charged, the capacity is the amount of power it contains. Batteries of the same type will often be rated by the amount of current they can output over time. For example, there are 1000mAh (milli-Amp Hour) and 2000mAh batteries.
- **Nominal Cell Voltage** - The average voltage a cell outputs when charged. The nominal voltage of a battery depends on the chemical reaction behind it. A lead-acid car battery will output 12V. A lithium coin cell battery will output 3V.
- The key word here is "nominal", the actual measured voltage on a battery will decrease as it discharges. A fully charged LiPo battery will produce about 4.23V, while when discharged its voltage may be closer to 2.7V.
- **Shape** - Batteries come in many sizes and shapes. The term 'AA' references a specific shape and style of a cell. There are a large variety.

# Power supply

Common batteries, their chemistry, and their nominal voltage

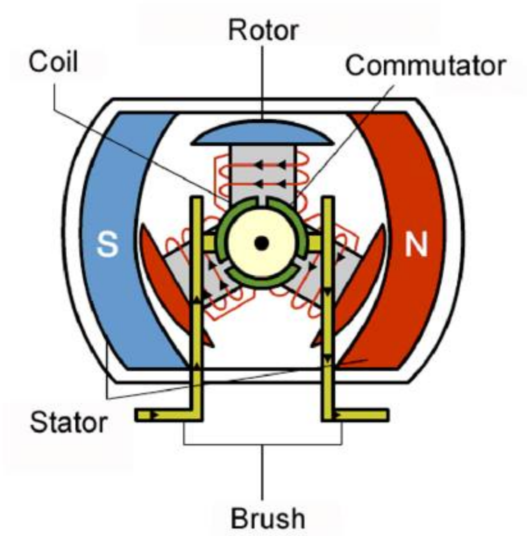
| Battery Shape | Chemistry | Nominal Voltage | Rechargeable? |
|---|---|---|---|
| AA, AAA, C, and D | Alkaline or Zinc-carbon | 1.5V | No |
| 9V | Alkaline or Zinc-carbon | 9V | No |
| Coin Cell | Lithium | 3V | No |
| Silver Flat Pack | Lithium Polymer (LiPo) | 3.7V | Yes |
| AA, AAA, C, D (Rechargeable) | NiMH or NiCd | 1.2V | Yes |
| Car Battery | Six-cell lead acid | 12.6V | Yes |

# Motion – motors




# Brushed DC electric motor

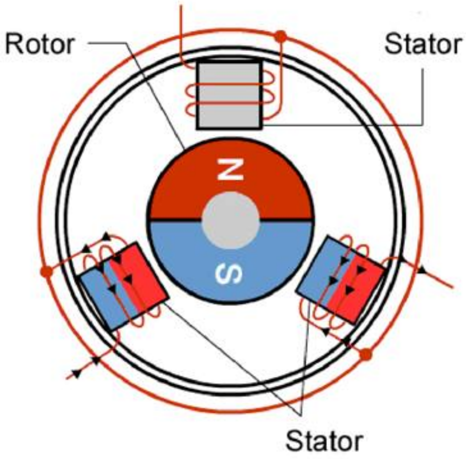
Fixed brushes supply electric energy to the rotating commutator. As the commutator rotates, it continually flips the direction of the current into the coils, reversing the coil polarities so that the coils maintain rightward rotation. The commutator rotates because it is attached to the rotor on which the coils are mounted.



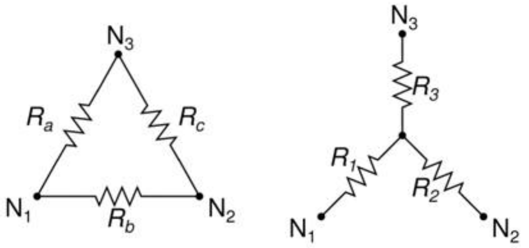

Schematic symbols

# Brushless DC electric motor

Since the rotor is a permanent magnet, it needs no current, eliminating the need for brushes and commutator. Current to the fixed coils is controlled from the outside.

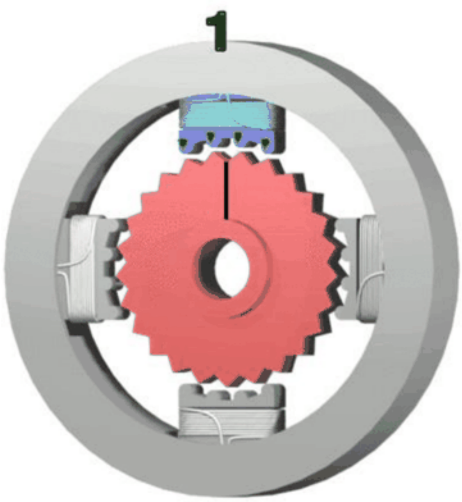



Schematic for delta and wye winding styles. (This image does not illustrate the motor's inductive and generator-like properties)

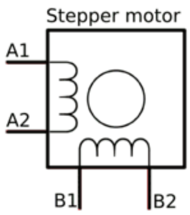



# Stepper motor

A stepper motor, also known as step motor or stepping motor, is a brushless DC electric motor that divides a full rotation into a number of equal steps. The motor's position can then be commanded to move and hold at one of these steps without any position sensor for feedback (an open-loop controller), as long as the motor is carefully sized to the application in respect to torque and speed.

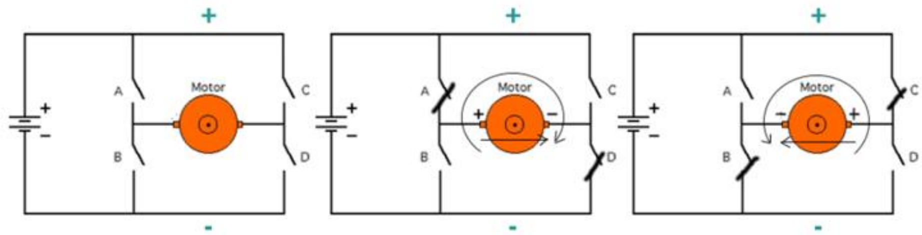



Schematic symbols

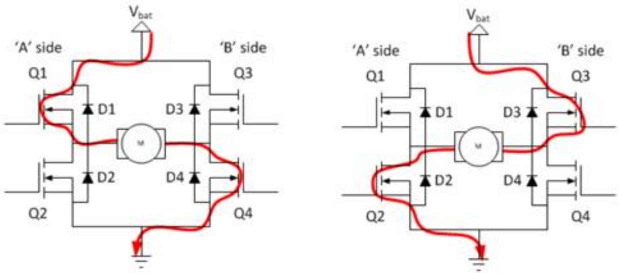


# Controlling Brushed DC Motors

Rotation in different directions

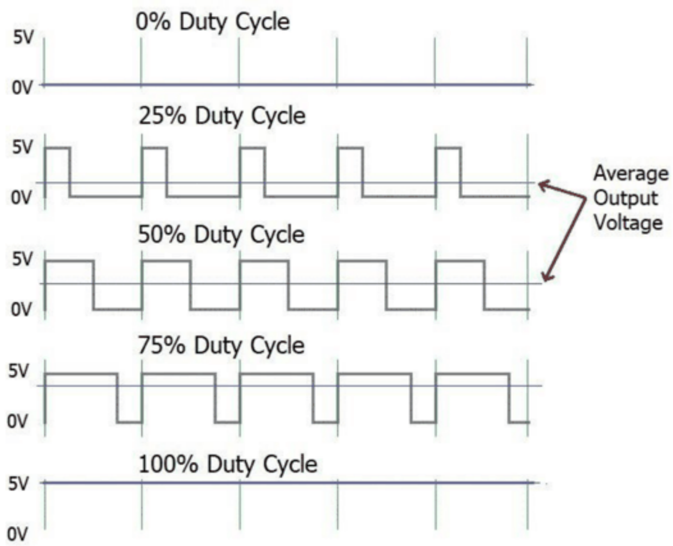



Schematic for microcontroller use

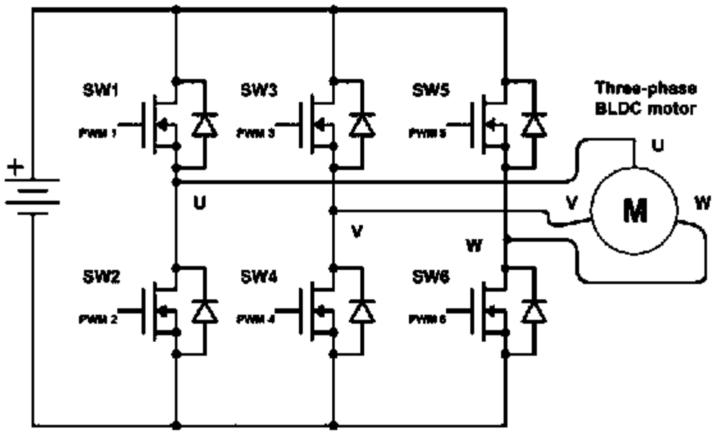



# Controlling Brushed DC Motors

Control speed by PWM (Pulse-Wide Modulation)



# Controlling Brushless DC Motors

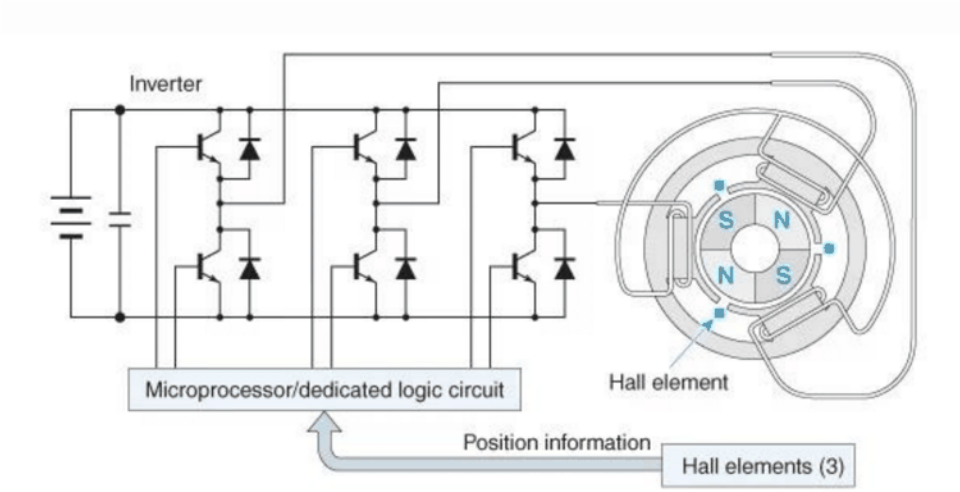



# Controlling Brushless DC Motors

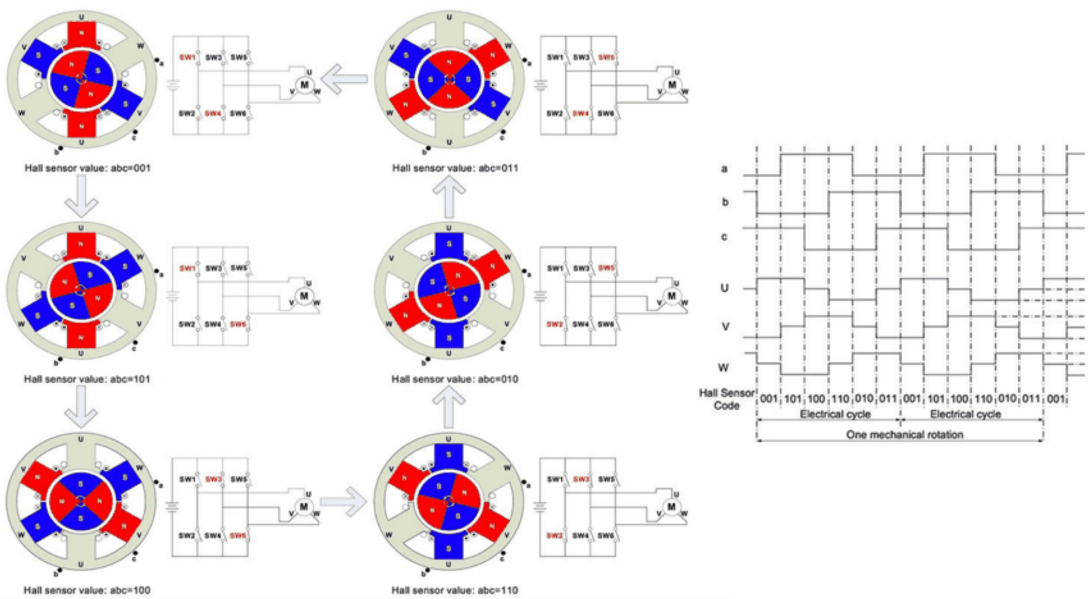
**Sensored vs. sensorless**

- Two technologies offer a solution for positional feedback. The first and most common uses three Hall-effect sensors embedded in the stator and arranged at equal intervals, typically 60° or 120°. A second, ‘sensorless’ control technology comes into its own for BLDC motors that require minimal electrical connections.

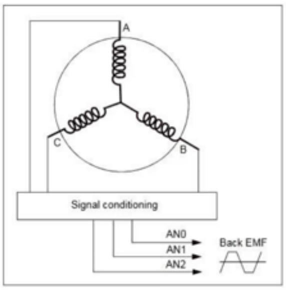
# Controlling Brushless DC Motors with sensors

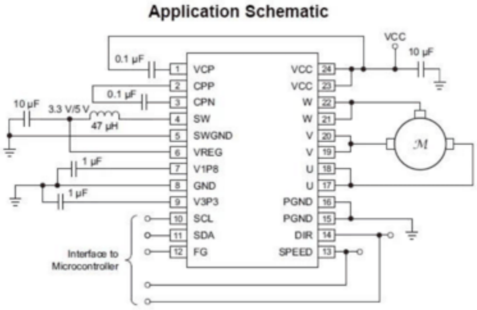


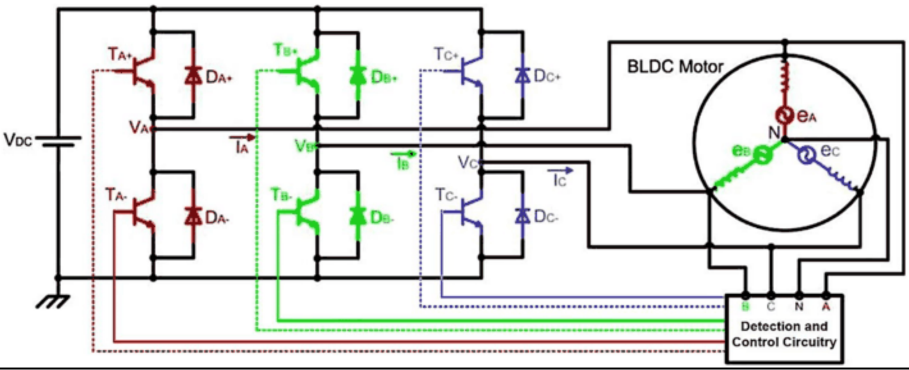

# Controlling Brushless DC Motors with sensors



# Controlling Brushless Sensorless DC Motors

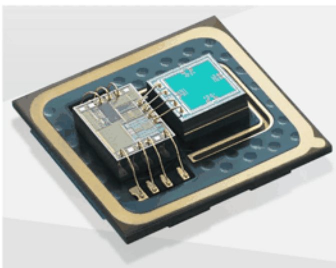


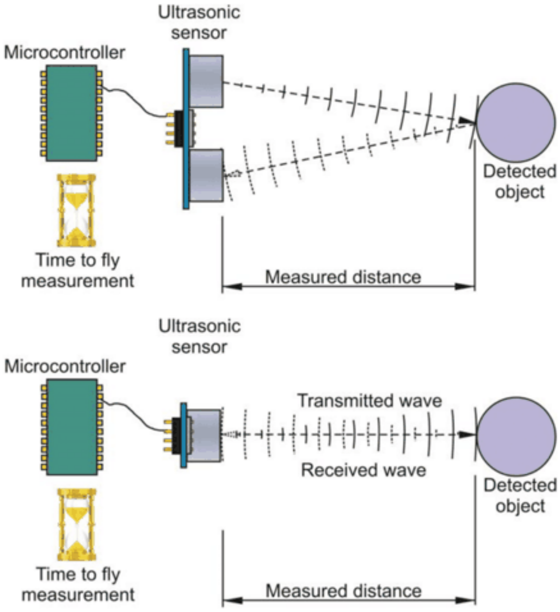

# Sensors

- Distance
- Position
- Velocity
- Temperature



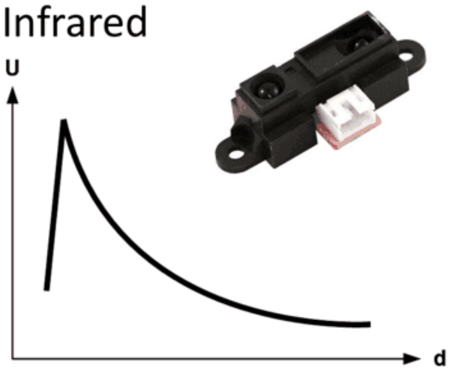
# Distance sensors

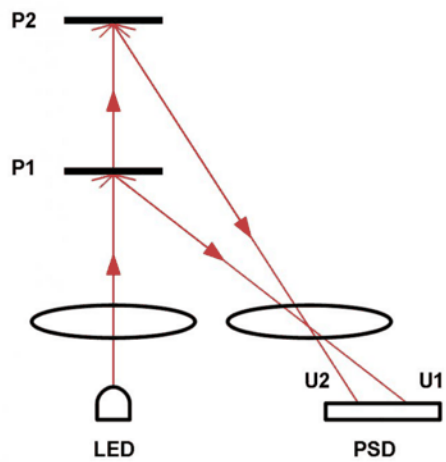
- Ultrasonic



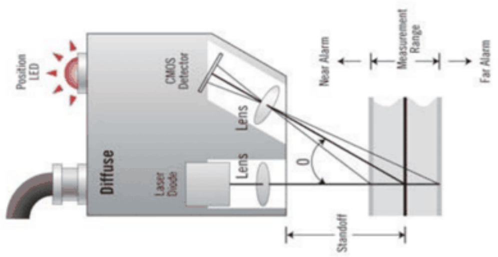

# Distance sensors

- Infrared




- Laser

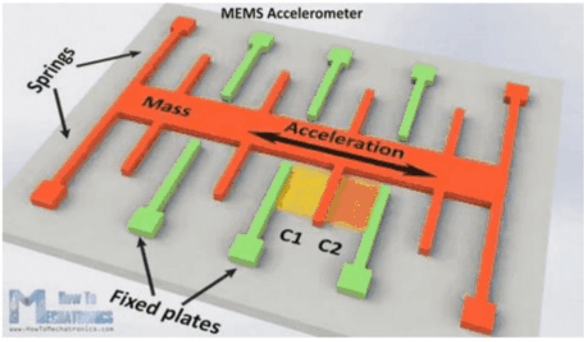


# Position/Velocity sensors

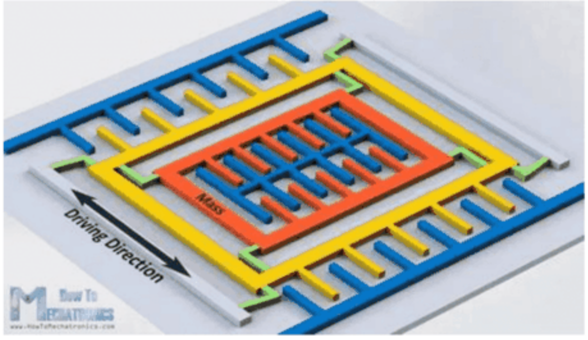
- MEMS (Micro Electro Mechanical Systems)
  - Accelerometer
  - Gyroscope
  - Magnetometer
- Encoder
- Potentiometer

# MEMS sensors

- Accelerometer

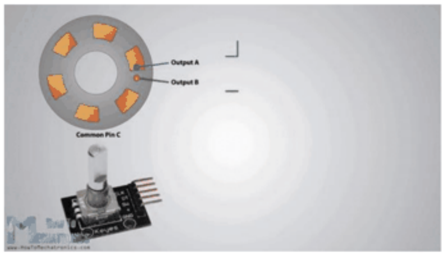



- Gyroscope

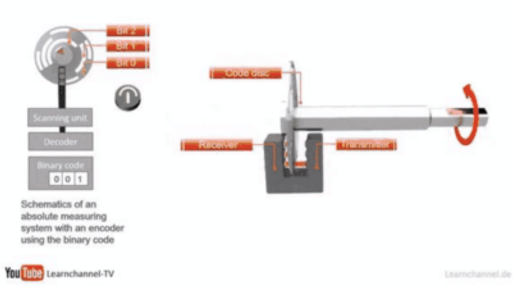


# Encoders

- Mechanical
- Optical

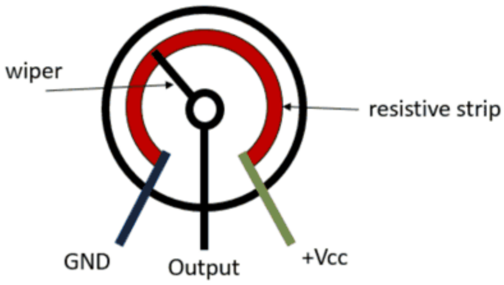


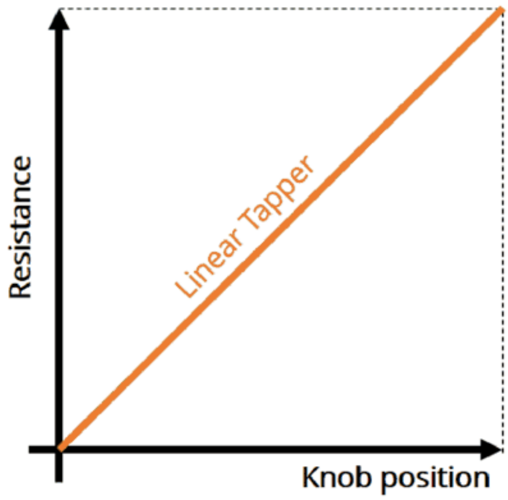

- Incremental
- Absolute





# Potentiometer



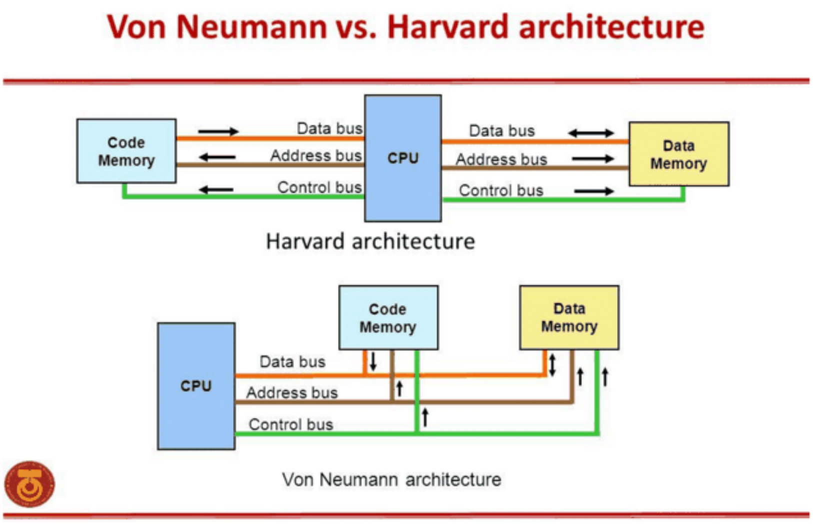
# Microcontrollers

A micro-controller can be comparable to a little stand alone computer; it is an extremely powerful device, which is able of executing a series of pre-programmed tasks and interacting with extra hardware devices. Being packed in a tiny integrated circuit (IC) whose size and weight is regularly negligible, it is becoming the perfect controller for as robots or any machines required some type of intelligent automation. A single microcontroller can be enough to manage a small mobile robot, an automatic washer machine or a security system. Several microcontrollers contains a memory to store the program to be executed, and a lot of input/output lines that can be a used to act jointly with other devices, like reading the state of a sensor or controlling a motor.





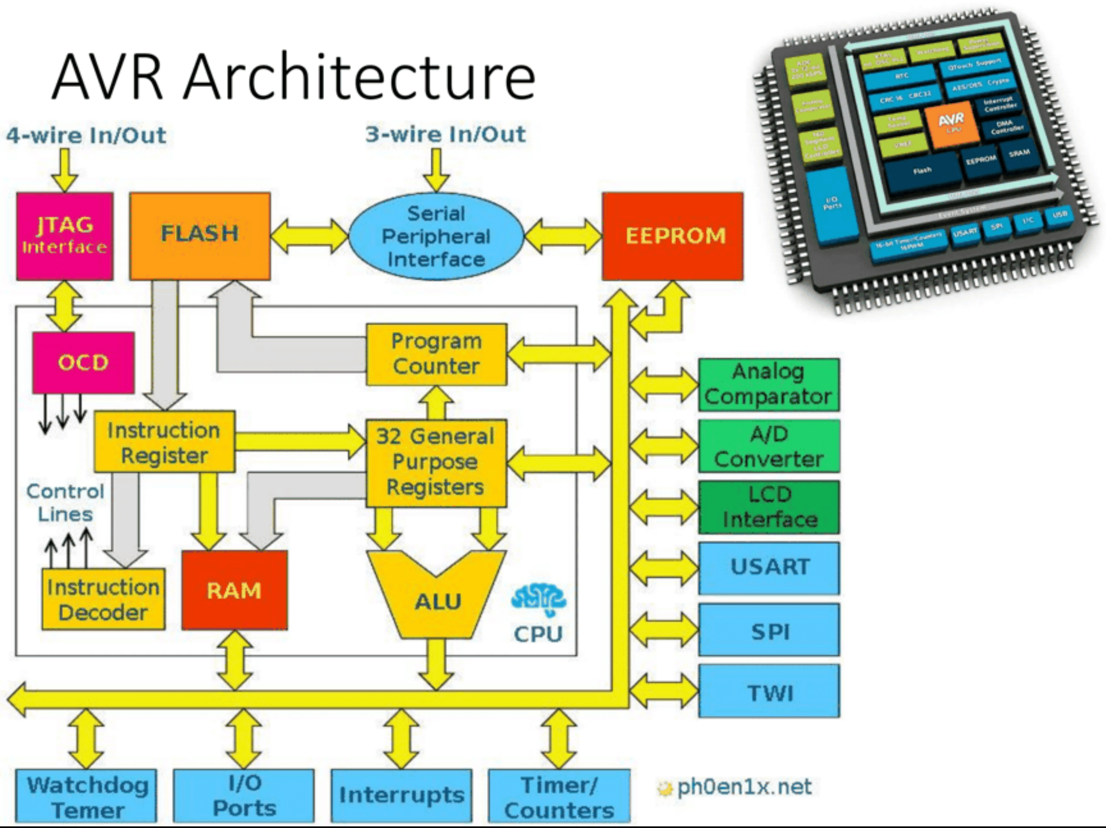
# Microcontroller’s architecture



# Microcontrollers

| | 8051 | PIC | AVR | ARM |
|---|---|---|---|---|
| Bus width | 8-bit for standard core | 8/16/32-bit | 8/32-bit | 32-bit mostly also available in 64-bit |
| Communication Protocols | UART, USART,SPI,I2C | PIC, UART, USART, LIN, CAN, Ethernet, SPI, I2S | UART, USART, SPI, I2C, (special purpose AVR support CAN, USB, Ethernet) | UART, USART, LIN, I2C, SPI, CAN, USB, Ethernet, I2S, DSP, SAI (serial audio interface), IrDA |
| Speed | 12 Clock/instruction cycle | 4 Clock/instruction cycle | 1 clock/ instruction cycle | 1 clock/ instruction cycle |
| Memory | ROM, SRAM, FLASH | SRAM, FLASH | Flash, SRAM, EEPROM | Flash, SDRAM, EEPROM |
| ISA | CLSC | Some feature of RISC | RISC | RISC |
| Memory Architecture | Von Neumann architecture | Harvard architecture | Modified | Modified Harvard architecture |
| Power Consumption | Average | Low | Low | Low |
| Families | 8051 variants | PIC16,PIC17, PIC18, PIC24, PIC32 | Tiny, Atmega, Xmega, special purpose AVR | ARMv4,5,6,7 and series |
| Community | Vast | Very Good | Very Good | Vast |
| Manufacturer | NXP, Atmel, Silicon Labs, Dallas, Cyprus, Infineon, etc. | Microchip Average | Atmel | Apple, Nvidia, Qualcomm, Samsung Electronics, and TI etc. |
| Cost (as compared to features provide) | Very Low | Average | Average | Low |
| Other Feature | Known for its Standard | Cheap | Cheap, effective | High speed operation Vast |
| Popular Microcontrollers | AT89C51, P89v51, etc. | PIC18fXX8, PIC16f88X, PIC32MXX | Atmega8, 16, 32, Arduino Community | LPC2148, ARM Cortex-M0 to ARM Cortex-M7, etc. |

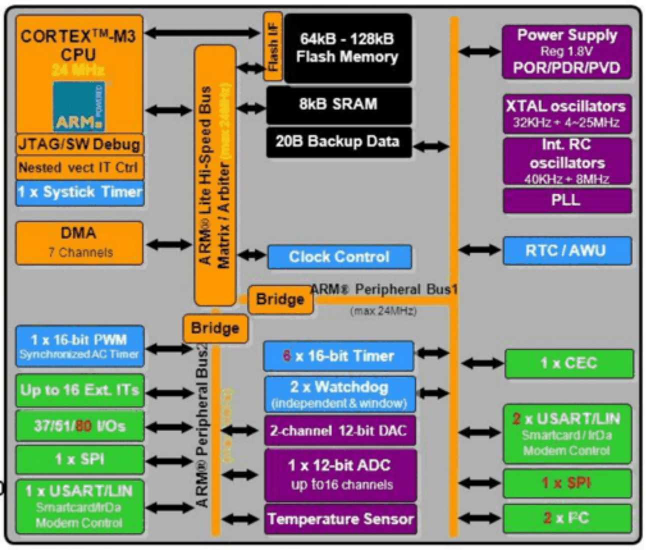
# AVR Architecture



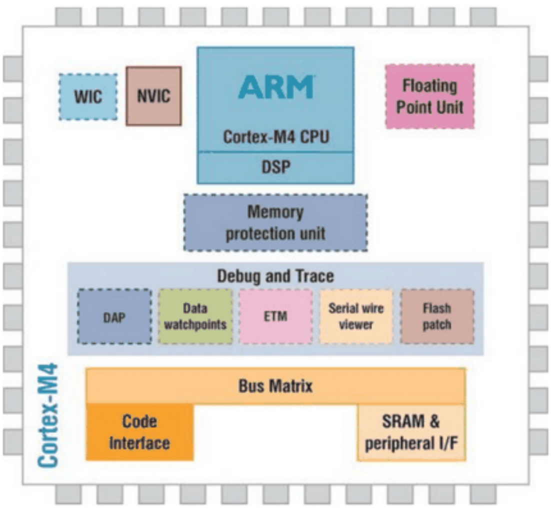
# ARM – STM32 Architecture

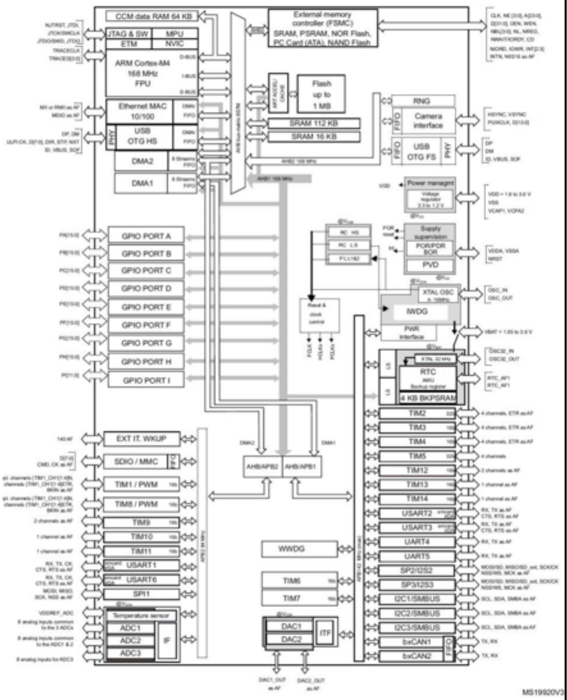
## STM32 Value line 64K-128KBytes block diagram

- **Core and operating conditions**
  - ARM® Cortex™-M3 1.25 DMIPS/MHz up to 24 MHz
  - 2.0 V to 3.6 V range
  - -40 to +105 °C
- **Rich connectivity**
  - 8 communications peripherals
- **Advanced analog**
  - 12-bit1.2 µs conversion time ADC
  - Dual channel 12-bit DAC
- **Enhanced control**
  - 16-bit motor control timer
  - 6x 16-bit PWM timers
- **LQFP48, LQFP/BGA64, LQFP100**

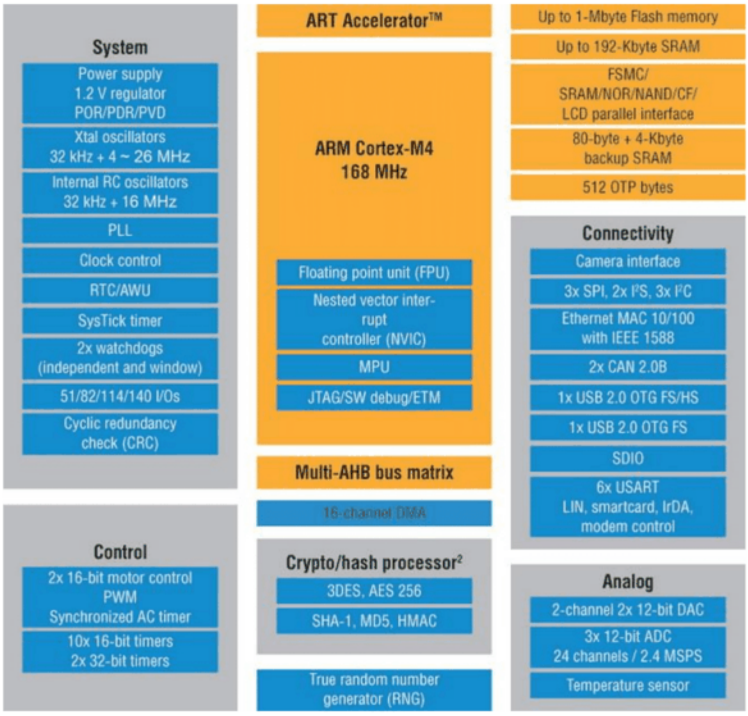



# STM32F40xxx architecture

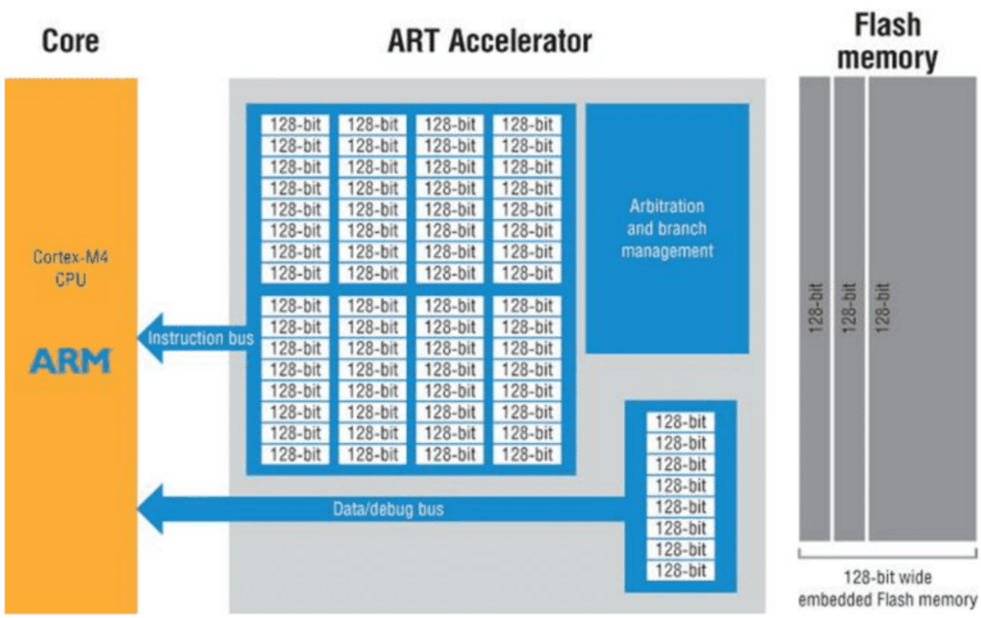


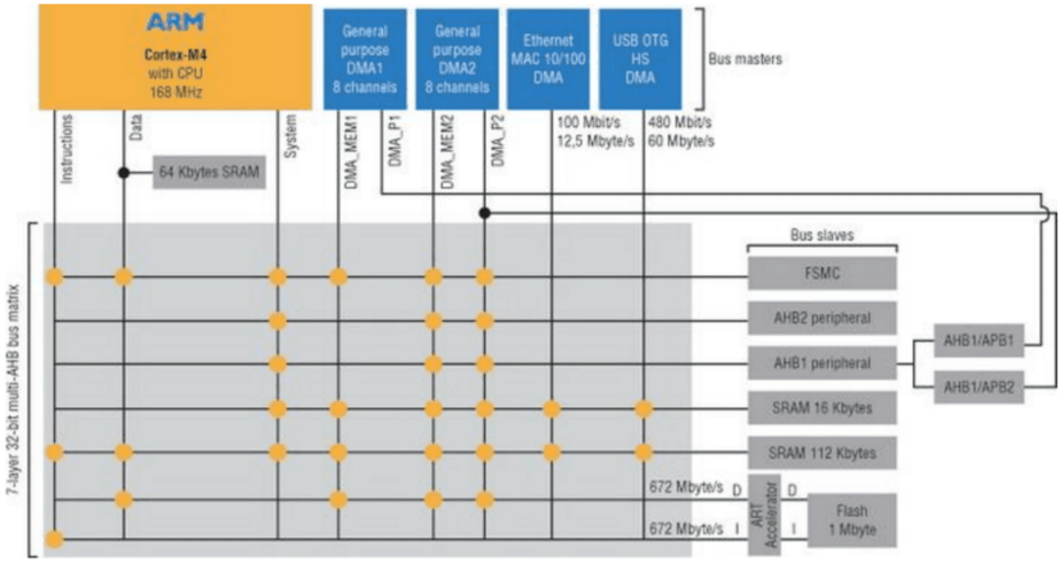
# STM32F4xxx Structure



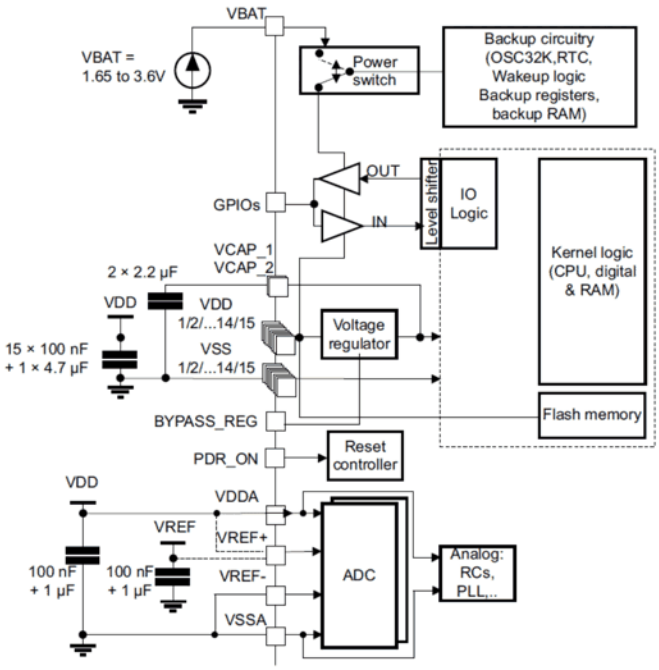

# STM Adaptive Real Time Memory

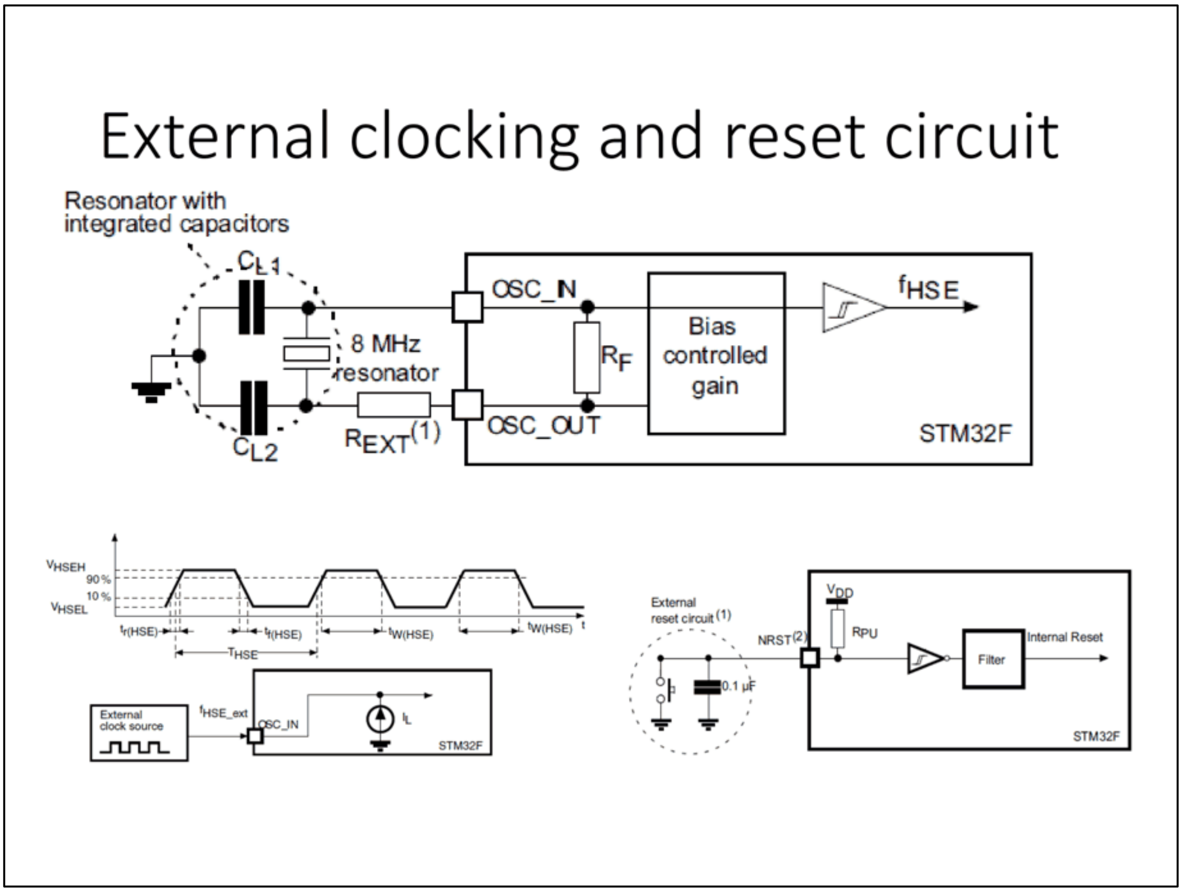


# STM Bus matrix




# Power supply scheme




External clocking and reset circuit
Resonator with integrated capacitors
CL1
8 MHz resonator
CL2
REXT(1)
OSC_IN
OSC_OUT
RF
Bias controlled gain
fHSE
STM32F
VHSEH
90 %
10 %
VHSEL
tr(HSE)
tf(HSE)
tW(HSE)
tW(HSE)
THSE
t
External clock source
fHSE_ext
OSC_IN
IL
STM32F
External reset circuit(1)
NRST(2)
VDD
RPU
0.1 µF
Filter
Internal Reset
STM32F

Actuators and mobile robots control

Mathematical model of DC motor

# Control and modeling of mobile robots

Mathematical model of DC motor

Alexander A. Kapitonov

## Constructon of DC motor

**Figure 1.** DC motor assembled.

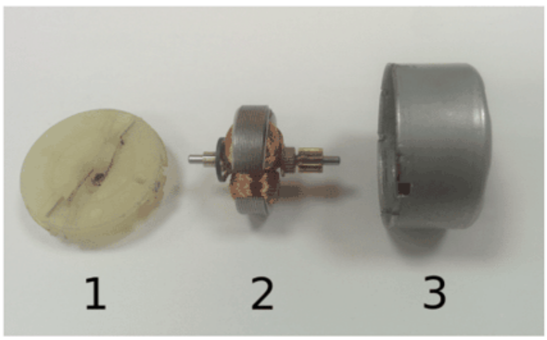



**Figure 2.** DC motor disassembled: 1 — cap, 2 — rotor, 3 — stator.

## Constructon of DC motor

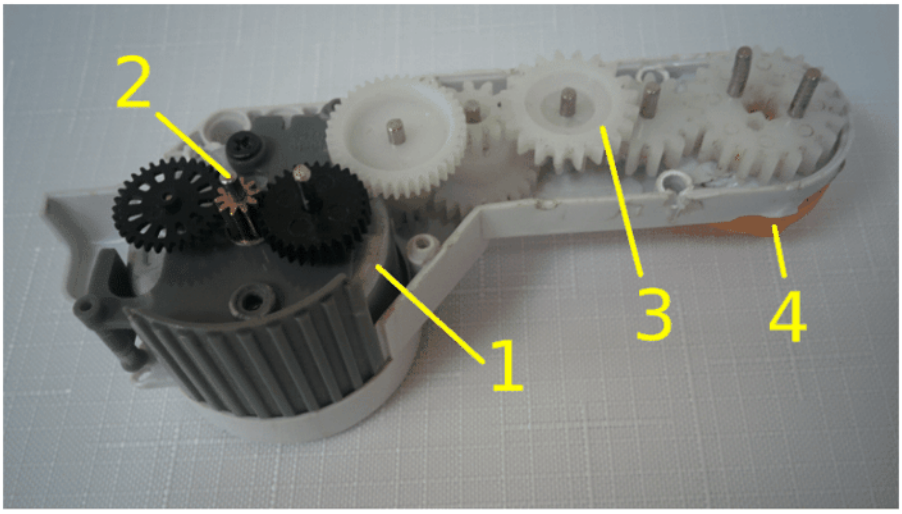



**Figure 3.** NXT motor disassembled: 1 — DC motor 2 — it's shaft, 3 — reducer, 4 — NXT motor external shaft; (chip with encoder isn't shown).

## Mathematical model

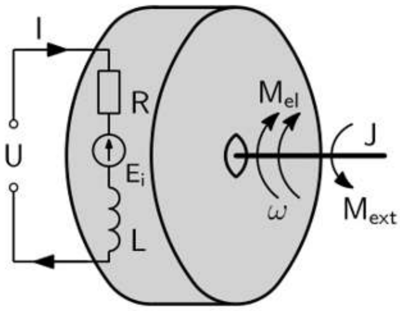



**Figure 4.** Physical scheme of DC motor.

$$\begin{cases} M_{el} - M_{ext} = J\dot{\omega}, \\ U = RI + E_i + L\dot{I}, \end{cases} \tag{1}$$

$$M_{el} = k_m I, \tag{2}$$

$$E_i = k_e \omega, \tag{3}$$

$$\begin{cases} k_m I - M_{ext} = J\dot{\omega}, \\ U = RI + k_e\omega + L\dot{I}, \end{cases} \tag{4}$$

where $M_{el}$ — motor torque; $M_{ext}$ — torque of external forces; $J$ — total moment of inertia of the rotor and reducer's gears; $\omega$ — rotor speed; $U$ — motor supply voltage; $R$, $L$ — resistance and an inductance of rotor's wires; $I$ — the current flowing through the latter; $E_i$ — EMF which appeared in rotor's wires due to its rotation in magnetic field of stator's magnets; $k_m$, $k_e$ — torque and back EMF motor constants.

## Mathematical model

If $L \approx 0$ H, then

$$I = \frac{1}{R}U - \frac{k_e}{R}\omega, \tag{5}$$

therefore

$$k_m\left(\frac{1}{R}U - \frac{k_e}{R}\omega\right) - M_{ext} = J\dot{\omega}, \tag{6}$$

hence

$$\frac{JR}{k_m k_e}\dot{\omega} + \omega - \frac{1}{k_e}U - \frac{R}{k_m k_e}M_{ext}, \tag{7}$$

$$T_m\dot{\omega} + \omega = \frac{1}{k_e}U - \frac{T_m}{J}M_{ext}, \tag{8}$$

where $T_m = \dfrac{JR}{k_m k_e}$ is a motor mechanical constant.

## Mathematical model

Also we can get differential equation which contains $I$, not $\omega$:

1. differentiating (5):

$$\dot{\omega} = \frac{1}{k_e}\dot{U} - \frac{R}{ke}\dot{I} \tag{9}$$

2. putting (9) to the first equation from (4):

$$k_m I - M_{ext} = J\left(\frac{1}{k_e}\dot{U} - \frac{R}{ke}\dot{I}\right) \tag{10}$$

3. transforming (10):

$$\frac{JR}{k_m k_e}\dot{I} + I = \frac{J}{k_m k_e}\dot{U} + \frac{1}{k_m}M_{ext}, \tag{11}$$

$$T_m\dot{I} + I - \frac{T_m}{R}\dot{U} + \frac{1}{k_m}M_{ext}. \tag{12}$$

## Mathematical model

For a situation when

$$\begin{cases} U = const, \\ M_{ext} = 0\ N \cdot m, \end{cases} \tag{13}$$

and

$$\begin{cases} \theta(0) = 0, \\ \omega(0) = 0\ s,^{1} \end{cases} \tag{14}$$

where $\theta$ — angle of rotor's rotation ($\dot{\theta} = \omega$), next expressions for $\omega(t)$ and $\theta(t)$ can be obtained from (8):

$$\omega(t) = \omega_{nls}\left(1 - \exp\left(\frac{t}{T_m}\right)\right), \tag{15}$$

$$\theta(t) = \omega_{nls}t - \omega_{nls}T_m + \omega_{nls}T_m \exp\left(\frac{t}{T_m}\right), \tag{16}$$

where $\omega_{nls} = U/k_e$ — no-load speed of the rotor.

## Mathematical model

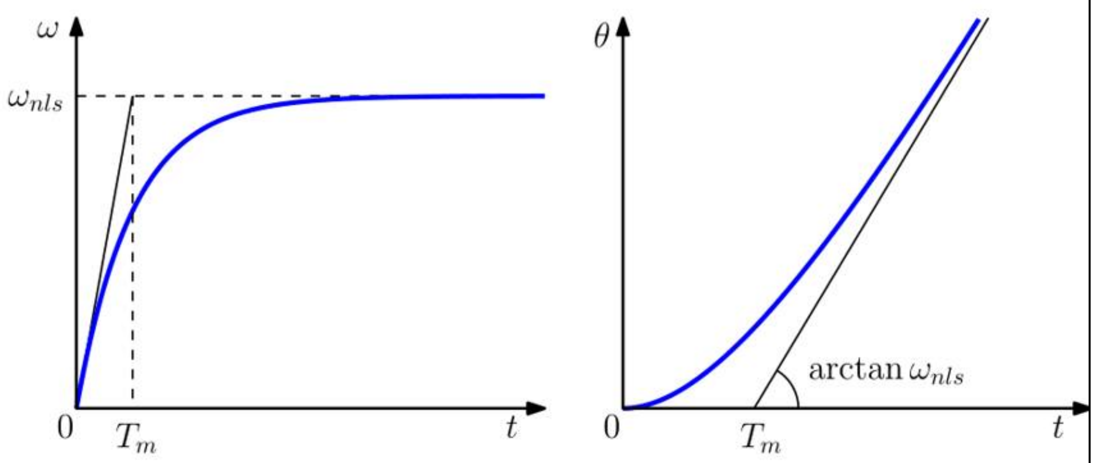



**Figure 5.** Graphs of $\omega(t)$ and $\theta(t)$ from (15) and (16) in case $\omega_{nls} > 0$.

## Description of experiment

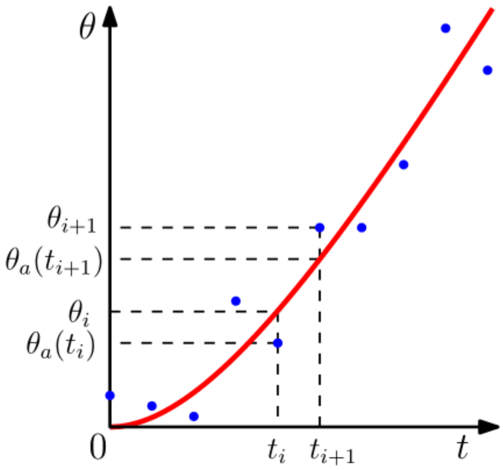



**Figure 6.** Approximation curve.

**Least squares method:**

find values for $\omega_{nls}$ and $T_m$ such that the sum $S$:

$$S = \sum_{j=1}^{N} \left(\theta_a(t_j) - \theta_j\right)^2 \qquad (17)$$

would have a minimal possible value.

There
$N$ — number of pairs $(t_j, \theta_j)$ which were recorded during the experiment,
$\theta_a(t_j)$ — value of (16) when $t = t_j$.

Modeling scheme of DC motor in Scilab

# Control and modeling of mobile robots

Modeling scheme of DC motor in Scilab

Alexander A. Kapitonov

## Scilab

Scilab is free and open source software for numerical computation providing a powerful computing environment for engineering and scientific applications.[1]

**Figure 1.** Scilab logo.

At this course we will use Scilab's:

- Xcos — hybrid dynamic systems modeler and simulator;
- some mathematical algorithms.

[1] Logo and some text on this slide were taken from www.scilab.org.

## System modeling

Figure 2 demonstrates example of modeling scheme for device which is desribed by this system of equations:

$$\begin{cases} \dot{x} = Ax + Bu, \\ y = Cx + Du, \\ u = K(x_d - x) + Lx_g. \end{cases} \tag{1}$$

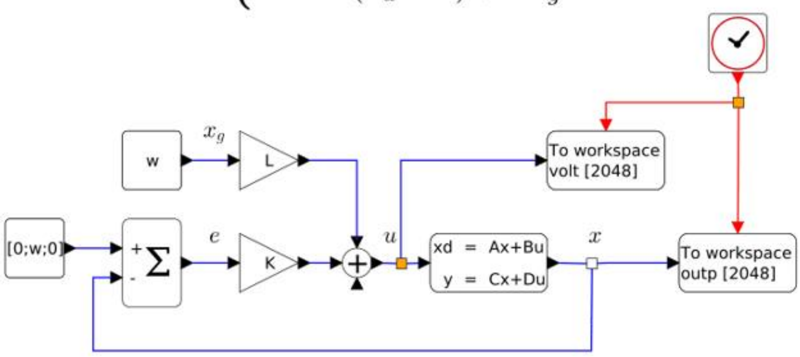



**Figure 2.** Example of modeling scheme.

## System modeling

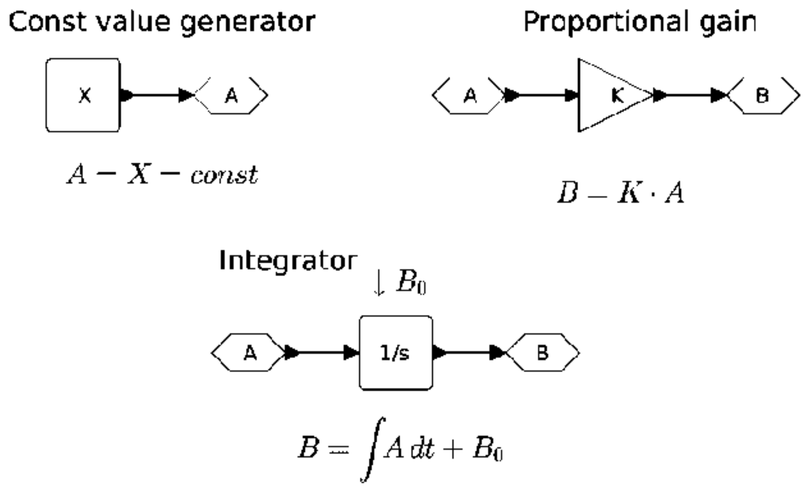



**Figure 3.** Some standard blocks.

## System modeling

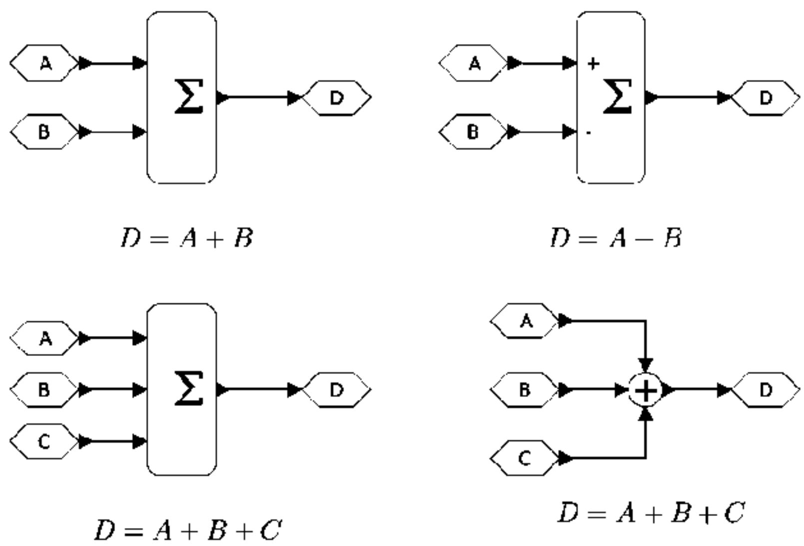



**Figure 4.** Summator block.

## System modeling

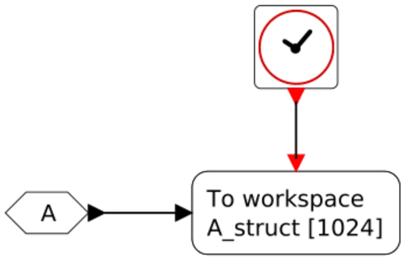



**Figure 5.** Some service blocks.

This subscheme saves values of $A$ and appropriate moments of time into two matrices: `A_struct.values` and `A_struct.time`.

## System modeling

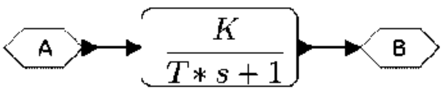



**Figure 6.** Transfer function block.

$$T \cdot \dot{B}(t) + B(t) = K \cdot A(t), \quad (2)$$

$$\mathcal{L}\{T \cdot \dot{B}(t) + B(t)\} = \mathcal{L}\{K \cdot A(t)\}, \quad (3)$$

$$T \cdot s \cdot B(s) - B(s) = K \cdot A(s), \quad (4)$$

$$\frac{B(s)}{A(s)} = \frac{K}{Ts+1}, \quad (5)$$

where $L\{\,\}$ — Laplace transform.

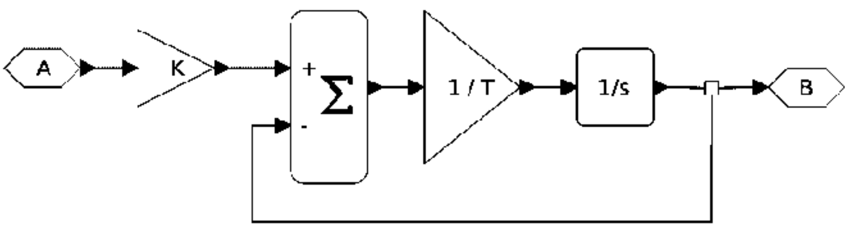



**Figure 7.** An equivalent sheme.

## Modeling scheme of DC motor

Model of DC motor is described by two diffential equations:

$$\begin{cases} T_m\dot{\omega} + \omega = \dfrac{1}{k_e}U - \dfrac{T_m}{J}M_{ext}, \\ T_m\dot{I} + I = \dfrac{T_m}{R}\dot{U} + \dfrac{1}{k_m}M_{ext}, \end{cases} \tag{6}$$

therefore its modeling scheme is equal to one which is demonstrated by figure 8.

## Modeling scheme of DC motor

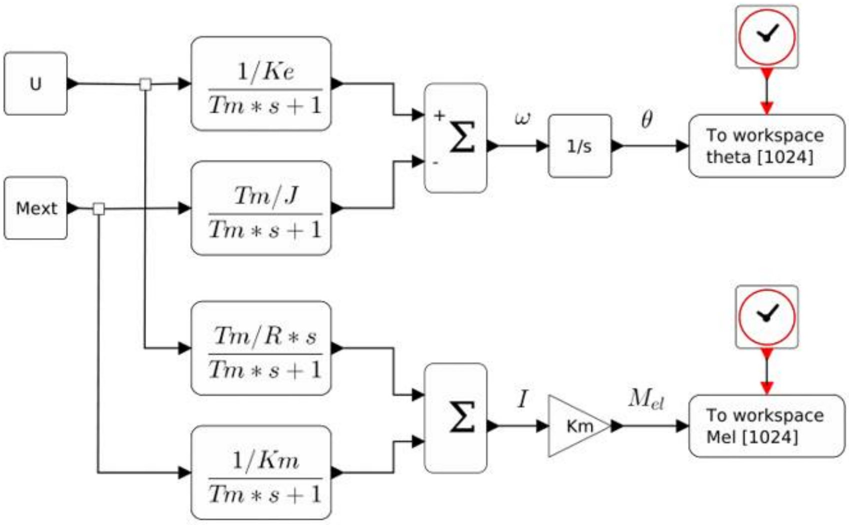



**Figure 8.** Modeling scheme of DC motor.

Control of DC motor using PID controller

# Control and modeling of mobile robots

## Control of DC motor using PID regulator

Alexander A. Kapitonov

## Some basics of control theory

In a control theory all systems are considered as a single object or a "box" which has some number of input and output signals.

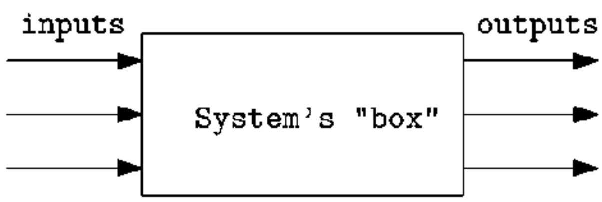



**Figure 1.** One of a possible representation of every system in a control theory.

**input signals** — some impacts which change system state

**output signals** — some physical quantities which describe system state

## Some basics of control theory

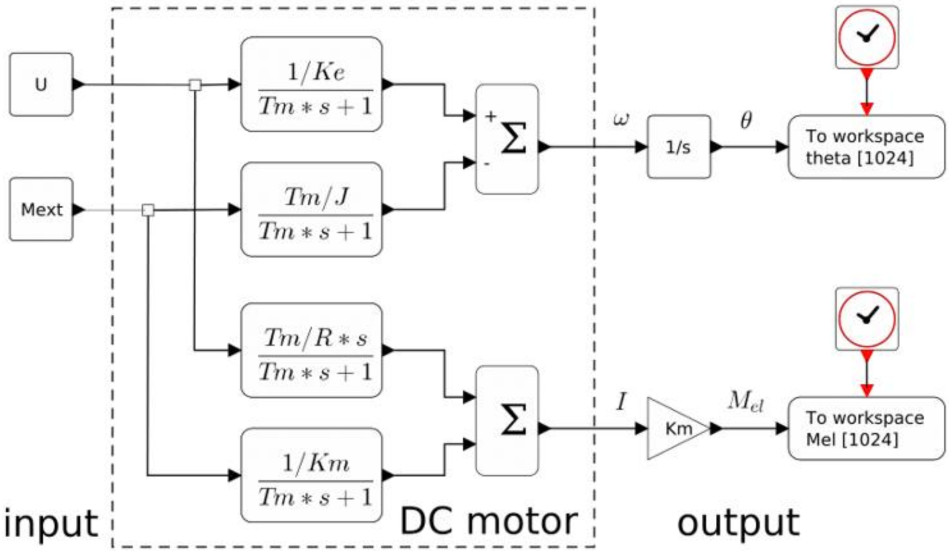



**Figure 2.** Structure of the model of a DC motor.

## Some basics of control theory

Some important definitions:

**Control** is a process of changing in a desired way values of some output signals using some input signals.

**Controller** is a special device and/or algorithm which creates required input signals.

Methods of control:

- forward
- using feedback

## Some basics of control theory

Forward control — a method of control when a controller doesn't use information about values of system's output signals.

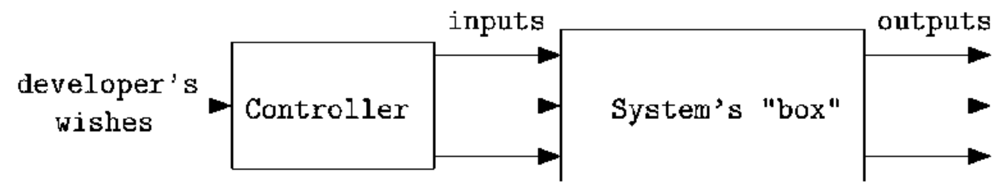



**Figure 3.** Scheme of forward control.

## Some basics of control theory

Control with feedback — a method of control when a controller use information about values of system's output signals.

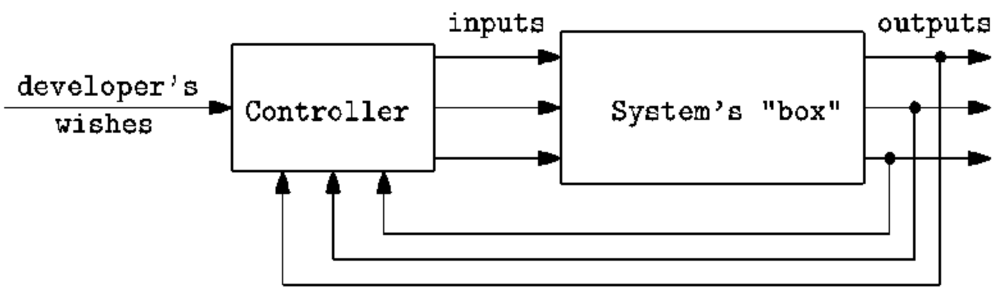



**Figure 4.** Scheme of control with feedback.

## PID controller

PID controller is an algorithm of feedback control which calculate value for input signal in accordance to the formulas:

$$e(t) = x_d(t) - x(t), \quad (1)$$

$$u = K_p \cdot e + K_i \cdot \int e\,dt + K_d \dot{e}, \quad (2)$$

where $x$ — controllable output signal; $x_d$ — desired value of signal; $e$ — error of control; $u$ — used system's input signal; $K_p$, $K_i$, $K_d$ — constant coefficients of PID controller.

## PID controller

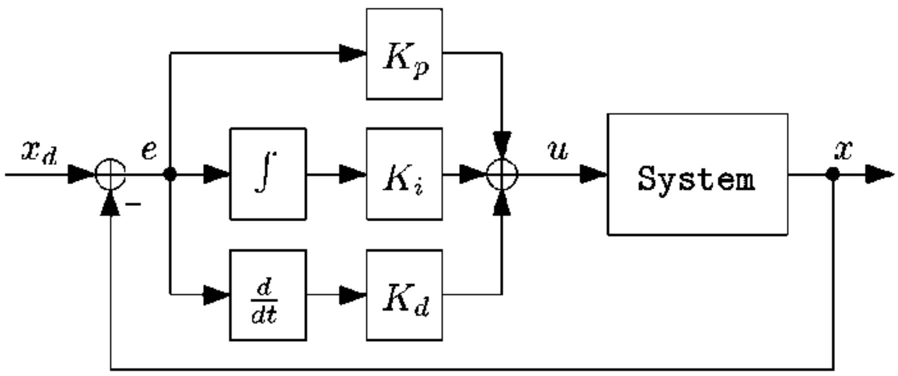



**Figure 5.** Scheme of PID controller sctructure.

# PID controller

**P controlller** or a proportional piece of PID which is calculated as

$$u = K_p \cdot e, \tag{3}$$

does the main part of a controller's job;

**I controlller** or a piece of PID with integral which is calculated as

$$u = K_i \cdot \int e \, dt, \tag{4}$$

prevents errors (makes $e$ is being equal to 0);

**D controlller** or a piece of PID with derivative which is calculated as

$$u = K_d \cdot \dot{e}, \tag{5}$$

dampens oscillations.

# PID controller

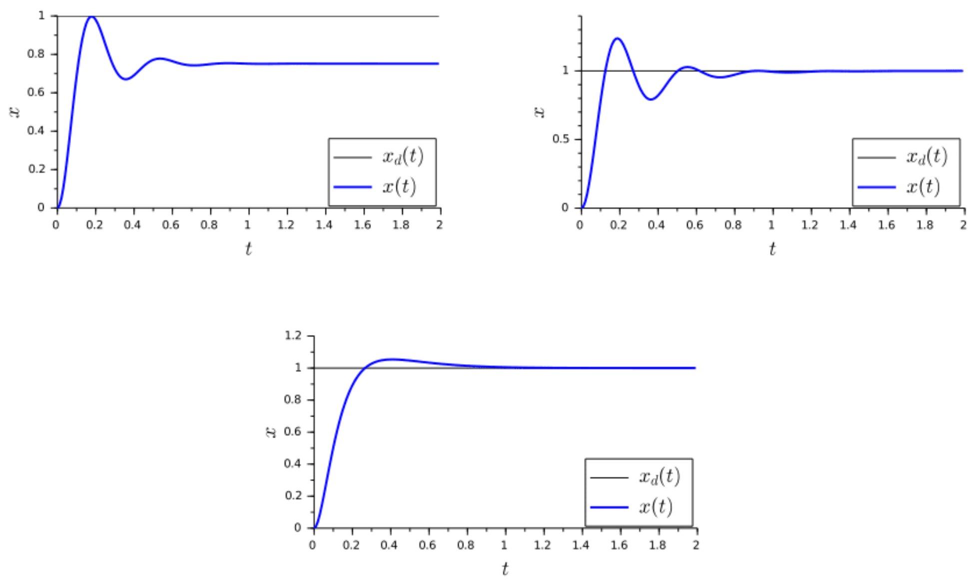



**Figure 6.** System with P, PI and PID controller respectively.

## PID controller

Methods of tuning controller's coefficients:

- calculations using mathematical model of a controllable object;
- setting with according to one of a special algorithm;
- fully manual setting.

## Ziegler–Nichols method

Algorithm of tuning values for coefficients of PID controller:

1. make $K_i$ and $K_d$ is being equal to 0;
2. increase value of $K_p$ until $x(t)$ starts making undamped oscillations; remember this value of $K_p$ as $K_u$ and a period of the oscillations as $T_u$;
3. calculate coefficients of PID controller using these formulas:

$$K_p = 0.6K_u, \qquad K_i = \frac{2K_p}{T_u}, \qquad K_d = \frac{K_pT_u}{8}. \qquad (6)$$

## Ziegler–Nichols method

This method's strengths:

- it is quite simple.

This method's weaknesses:

- it doesn't work for all systems;
- it doesn't give the best value of coefficients.

## Numerical methods

$$\int_{t_1}^{t_2} f\,dt = S_{1,2} \quad (7)$$

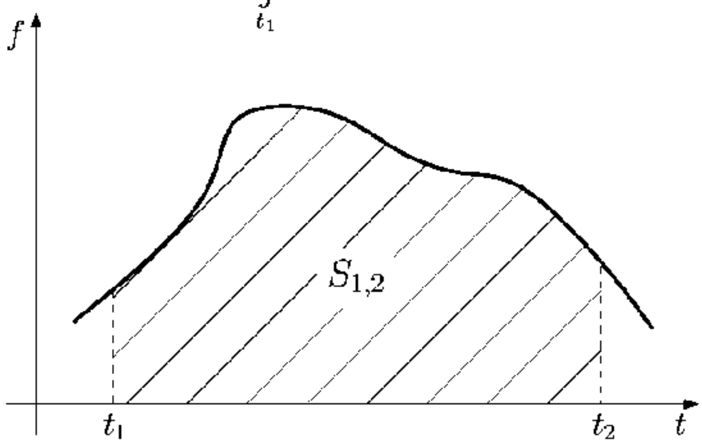



**Figure 7.** Geometry meaning of integrals.

## Numerical methods

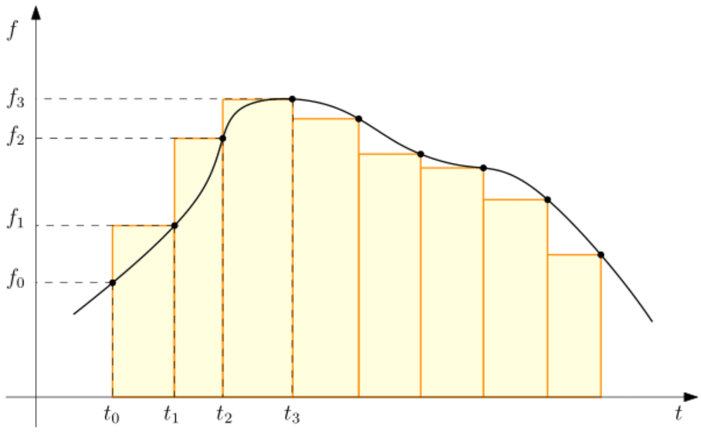



**Figure 8.** One of numerical methods for calculating value of integral.

$$\int_{t_m}^{t_n} f\,dt \approx \sum_{i=m+1}^{n} f_i(t_i - t_{i-1}), \quad m < n, \;\; m, n \in \mathbb{Z} \tag{8}$$

## Numerical methods

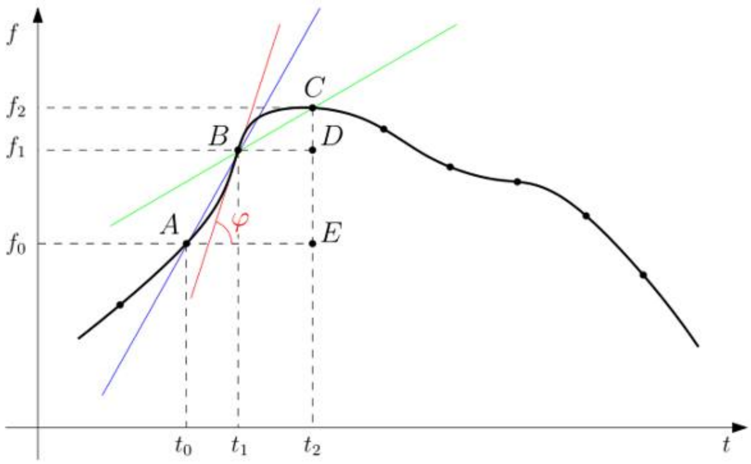



**Figure 9.** Numerical methods for derivative calculating.

$$f'(t_1) = \lim_{\Delta t \to 0} \frac{f(t_1 + \Delta t) - f(t_1)}{\Delta t} = \operatorname{tg}\varphi, \quad f'(t_1) \approx \frac{f_1 - f_0}{t_1 - t_0} = \operatorname{tg}\angle BAE$$

A controller for to-point motion for a mobile robot with differential drive type

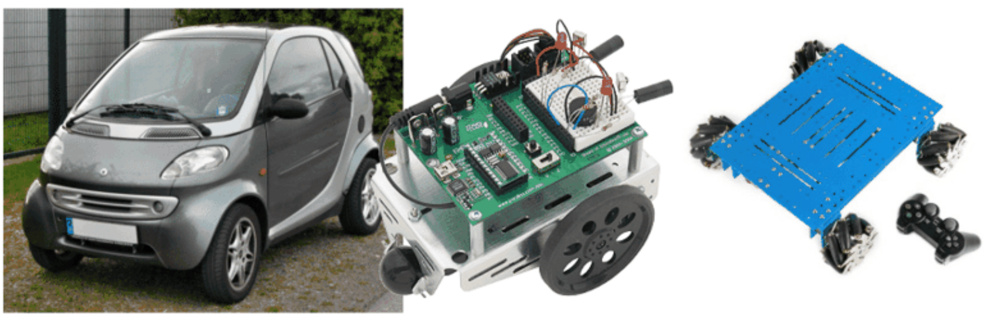
# Control and modeling of mobile robots

## A controller for to-point motion for a mobile robot with differential drive type

Alexander A. Kapitonov

## Robots' drive types

| Drive type | Controllable velocities |
|---|---|
| Car-like type | $v_x,\ \omega(v_x)$ |
| Differential | $v_x,\ \omega$ |
| Omnidirectional | $v_x,\ v_y,\ \omega$ |



a) b) c)

**Figure 1.** Examples of "robots" with different drive types: a—car-like, b—differential, c—omnidirectional

## General view of the robot

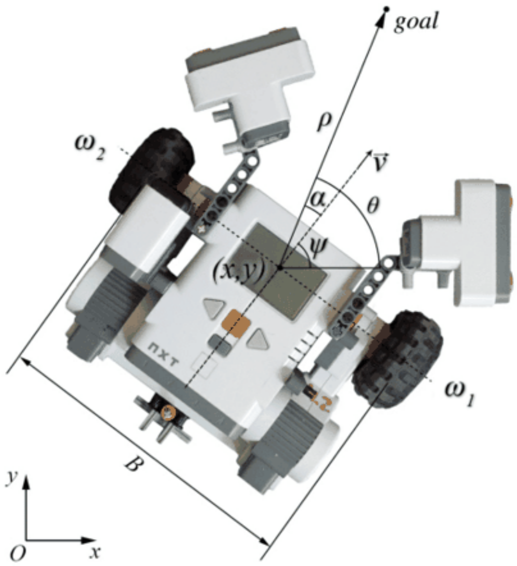



**Figure 2.** General view of a considered robot.

## Structure of the control system

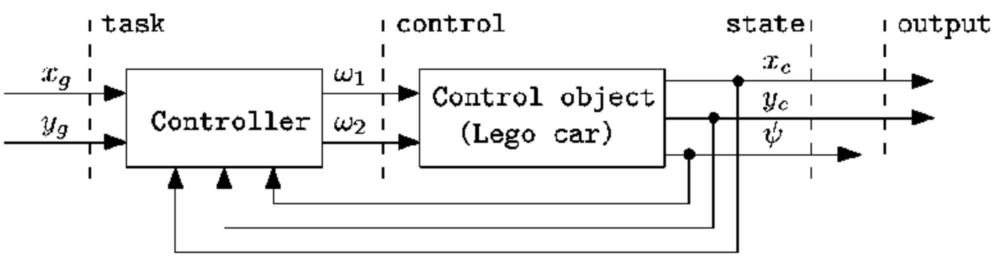



**Figure 3.** Structure of the control system.

$x_g$, $y_g$ — coordinates of goal point;

$\omega_1$, $\omega_2$ — angular velocities of robot's wheels;

$x_c$, $y_c$, $\psi$ coordinates and rotation angle of the robot.

## Mathematical model of the robot

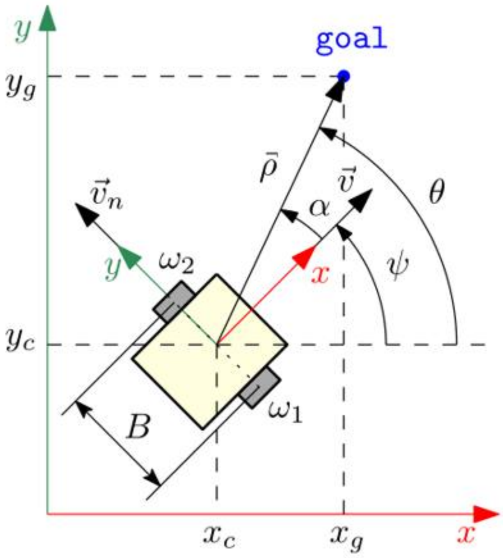



**Figure 4.** Useful drawing.

Kinematic model:

$$\begin{cases} \dot{x}_c = |\vec{v}|\cos\psi \\ \dot{y}_c = |\vec{v}|\sin\psi \\ \dot{\psi} = \omega \end{cases} \tag{1}$$

where

$$|\vec{v}| = R \cdot \frac{\omega_1 + \omega_2}{2}, \tag{2}$$

$$\omega = \frac{R}{B} \cdot (\omega_1 - \omega_2), \tag{3}$$

where $R$ — wheel radius.

## Mathematical model of the robot

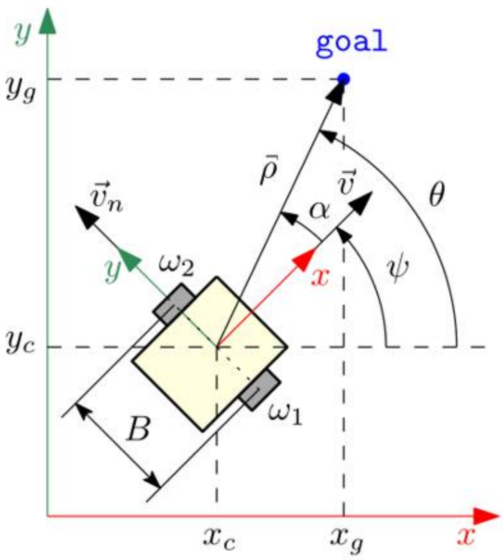



**Figure 5.** Useful drawing.

Some important variables:

$$\vec{\rho} = \left\{ x_g - x_c \quad y_g - y_c \right\}, \tag{4}$$

$$\theta = \arctan \frac{y_g - y_c}{x_g - x_c}, \tag{5}$$

$$\alpha = \theta - \psi, \tag{6}$$

$$|\vec{v}_n| = |\vec{v}|. \tag{7}$$

## Mathematical model of the robot

$$|\vec{\rho}| - \sqrt{(x_g - x_c)^2 + (y_g - y_c)^2} \tag{8}$$

$$\begin{aligned}
\frac{d|\vec{\rho}|}{dt} &= \frac{1}{2\sqrt{(x_g - x_c)^2 + (y_g - y_c)^2}} \cdot \left((x_g - x_c)^2 + (y_g - y_c)^2\right)' = \\
&= \frac{1}{2|\vec{\rho}|}(-2\dot{x}_c(x_g - x_c) - 2\dot{y}_c(y_g - y_c)) = \\
&= -\frac{1}{|\vec{\rho}|} \cdot \{\dot{x}_c \;\; \dot{y}_c\} \cdot \{x_g - x_c \;\; y_g - y_c\} = -\frac{1}{|\vec{\rho}|} \cdot \vec{v} \cdot \vec{\rho} = -|\vec{v}| \cos\alpha
\end{aligned} \tag{9}$$

$$\dot{\alpha} = \dot{\theta} - \dot{\psi} \tag{10}$$

## Mathematical model of the robot

$$\begin{aligned}
\dot{\theta} &= \left(\arctan\frac{y_g - y_c}{x_g - x_c}\right)' = \frac{1}{1 + \left(\frac{y_g - y_c}{x_g - x_c}\right)^2} \cdot \left(\frac{y_g - y_c}{x_g - x_c}\right)' = \\
&= \frac{(x_g - x_c)^2}{(x_g - x_c)^2 + (y_g - y_c)^2} \cdot \frac{-\dot{y}_c(x_g - x_c) + \dot{x}_c(y_g - y_c)}{(x_g - x_c)^2} = \\
&= \frac{\{-\dot{y}_c \;\; \dot{x}_c\} \cdot \{x_g - x_c \;\; y_g - y_c\}}{|\vec{\rho}|^2} = \frac{\vec{v}_n \cdot \vec{\rho}}{|\vec{\rho}|^2} = \\
&= \frac{|\vec{v}_n| \cos(90^\circ - \alpha)}{|\vec{\rho}|} = \frac{|\vec{v}| \sin\alpha}{|\vec{\rho}|}
\end{aligned} \tag{11}$$

$$\dot{\alpha} = \frac{|\vec{v}| \sin\alpha}{|\vec{\rho}|} - \omega \tag{12}$$

## Mathematical model of the robot

Robot's mathematical model:

$$\begin{cases} \dfrac{d|\vec{\rho}|}{dt} = -|\vec{v}|\cos\alpha \\ \dfrac{d\alpha}{dt} = \dfrac{|\vec{v}|\sin\alpha}{|\vec{\rho}|} - \omega \end{cases} \quad \text{or} \quad \dot{x} = f(x), \text{ where } x = \begin{bmatrix} |\vec{\rho}| \\ \alpha \end{bmatrix} \tag{13}$$

Let's use for it this control law:

$$\begin{cases} |\vec{v}| = v_{max} \cdot \tanh|\vec{\rho}| \cdot \cos\alpha \\ \omega = K_\omega \alpha + v_{max} \cdot \dfrac{\tanh|\vec{\rho}|}{|\vec{\rho}|} \cdot \sin\alpha \cdot \cos\alpha \end{cases} \tag{14}$$

where $v_{max}$ and $K_\omega$ are constant positive coefficients.

## Mathematical model of the robot

Some theoretical information:

- Stability is an ability of a controlled system to run to particular state and stay in it.
- For checking system for stability Lyapunov functions are used.
- If time derivative of Lyapunov functions for considered system is always negative, the system is stable.

## Mathematical model of the robot

Possible Lyapunov function for our system:

$$V(x) = \frac{1}{2}|\vec{\rho}|^2 + \frac{1}{2}\alpha^2 \tag{15}$$

Its derivative:

$$\frac{dV}{dt} = \frac{d|\vec{\rho}|}{dt} \cdot |\vec{\rho}| + \frac{d\alpha}{dt} \cdot \alpha = -|\vec{v}||\vec{\rho}|\cos\alpha + \alpha\left(\frac{|\vec{v}|\sin\alpha}{|\vec{\rho}|} - \omega\right) \tag{16}$$

or after using equations (14) for control law:

$$\frac{dV}{dt} = -v_{max} \cdot |\vec{\rho}| \cdot \tanh\ \vec{\rho}\ \cdot \cos^2\alpha - K_\omega\alpha^2 < 0. \tag{17}$$

Due to $\dot{V}$ is always negative the system is stable.

## Mathematical model of the robot

Note that:

- angular speeds of robot's wheels can be found using these formulas:

$$\omega_1 = \frac{1}{R} \cdot (2|\vec{v}| + B\omega), \qquad \omega_2 = \frac{1}{R} \cdot (2|\vec{v}| - B\omega). \tag{18}$$

- in the steady state angular speeds of robot's motors are proportional to voltages which are applied to them; so we will make the latters are being proportional to values obtained from equations (18).

## Sources for pictures

- slide 2:
    - https://en.wikipedia.org/wiki/Car
    - https://www.parallax.com/product/boe-bot-robot
    - http://www.makeblock.com/mecanum-wheel-robot-kit

ITMO UNIVERSITY
Faculty of Control systems and robotics

per. Grivtsova, 14,
Saint Petersburg, Russia, 190000

+7 (812) 595-41-28
csi.ifmo.ru/en/