Robotics and Control

Anders Robertsson

Department of Automatic Control, LTH
Lund University, Sweden

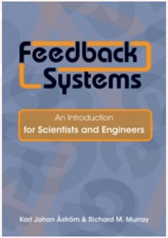
Outline

- Preliminaries, motion control
  - Exact-linearization, cascaded control, ...
  - Servo control
- Jacobians for velocity and force
- Different force control schemes

Literature

- "Springer Handbook of Robotics"
  B. Siciliano; O. Khatib (Eds.)
  Industrial Robotics: Hägele, Nilsson, Pires

- “Feedback Systems:
  An Introduction for Scientists and Engineers”
  `http://www.cds.caltech.edu/~murray/amwiki/Main_Page`
  by Karl J. Åström and Richard M. Murray
  (IFAC 2011, Harald Chestnut Control Engineering Textbook Prize)

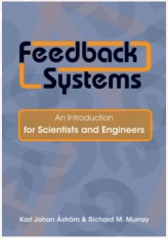


- ...and several others: [Murray, Li, Sastry] [Siciliano *et al*], [Spong, Hutchinson, Vidyasagar], [Craig], [Paul], [Lynch and Park], [Corke] ...

Literature

- "Springer Handbook of Robotics"
  B. Siciliano; O. Khatib (Eds.)
  Industrial Robotics: Hägele, Nilsson, Pires

- “Feedback Systems:
  An Introduction for Scientists and Engineers”
  `http://www.cds.caltech.edu/∼murray/amwiki/Main_Page`
  by Karl J. Åström and Richard M. Murray
  (IFAC 2011, Harald Chestnut Control Engineering Textbook Prize)

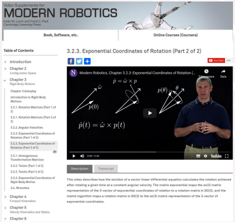


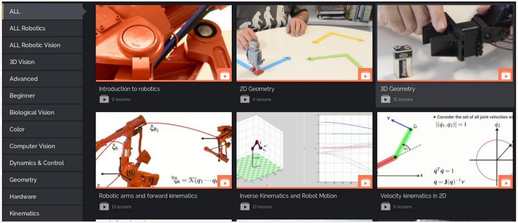
- ...and several others: [Murray, Li, Sastry] [Siciliano *et al*], [Spong, Hutchinson, Vidyasagar], [Craig], [Paul], [Lynch and Park], [Corke] ...

- Kevin Lynch and Frank Park, *Modern Robotics*

  `http://hades.mech.northwestern.edu/index.php/Modern_Robotics`

- Peter Corke
  - Robot Academy
  - Robotics Toolbox (matlab, python, ...)
  - Robotics, Vision & Control
  - `https://petercorke.com/`





Robotics - Application areas

- Industrial robotics
- Mobile Robotics
- Service robotics
- Entertainment
- ...

Multi-disciplinary:
Control, mechatronics, real-time embedded systems...

Robotics - Application areas

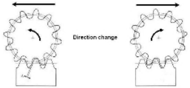
- Industrial robotics
- Mobile Robotics
- Service robotics
- Entertainment
- ...

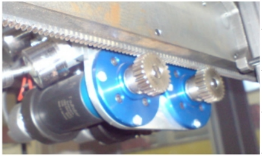
Multi-disciplinary:
Control, mechatronics, real-time embedded systems...

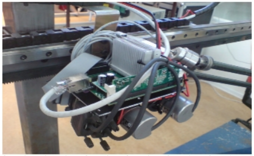
Parallell Kinematic Control

- Stiffness
  - Mechanical bandwidth
- Large open workspace,
- "Little" moving mass
  - Large accelerations

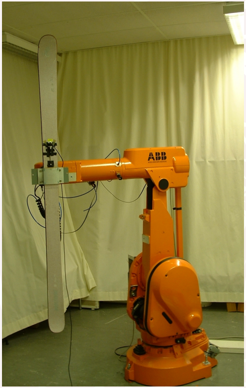
Rack-and-pinion: dual motor control for backlash reduction [module by Stefan S]



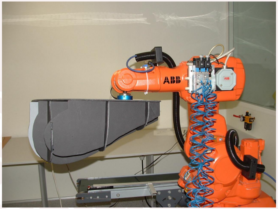




Identification of resonant modes and stiffness of skies [Ridea]



Damping of resonant systems
[R. Olsson *et al*, 2004]

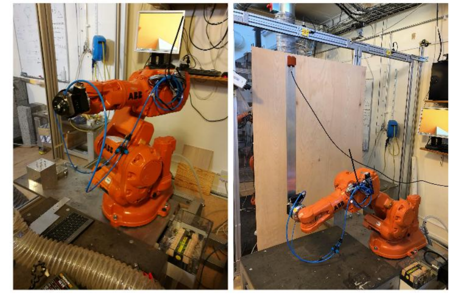


“Desarrollo de un banco de pruebas experimental mediante control de fuerza con robot industrial para el análisis de la respuesta mecánica de asientos de coche”, [Valera *et al* 2009]

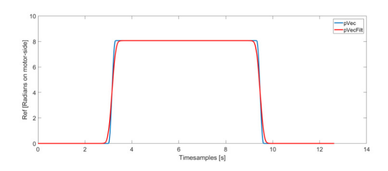
Show video

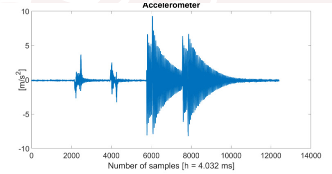
Using force sensor measurement for flexible structure damping

Effect of prefiltering resonance(s)



Notch away resonance fq(s) from reference/input signal:





Master's thesis, M. Peterson [movie]

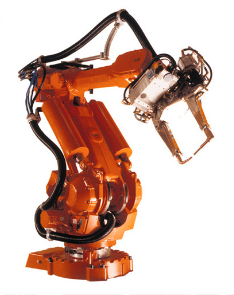
Flexible robots

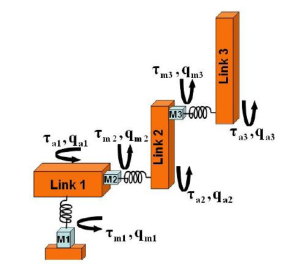
- Nonminimum-phase phenomenon

Modeling

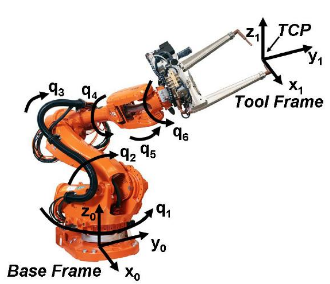
What models to use?

- Rigid body models
- Flexible joints
  - Single direction [Spong *et al*]
  - Multiple directions
- Flexible links (PDEs)





[Moberg, 2007]



How detailed models shall we use?

As simple as possible,

which captures the behaviour in our "operating range" and intended use (frequency, performance, phenomena of interest / simulation, prediction, control...)

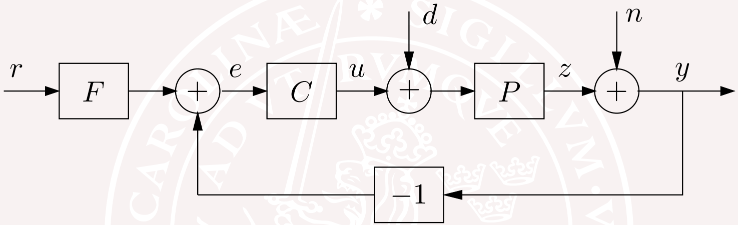
Modelling and control

How detailed models shall we use?

As simple as possible,

which captures the behaviour in our "operating range" and intended use (frequency, performance, phenomena of interest / simulation, prediction, control...)

A Basic Control System



Ingredients:

- Controller: feedback C, feedforward F
- Load disturbance d: Drives the system from desired state
- Measurement noise n: Corrupts information about z
- Process variable z should follow reference r

Key Issues

Find a controller that

A: Reduces effects of load disturbances

B: Does not inject to much measurement noise into the system

C: Makes the closed loop insensitive to variations in the process

D: Makes the output follow command signals

Convenient to use a controller with *two degrees of freedom*, i.e. separate signal transmission from y to u and from r to u. This gives a complete separation of the problem: Use feedback to deal with A, B, and C. Use feedforward to deal with D!

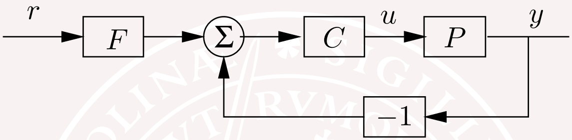
Designing System with Two Degrees of Freedom

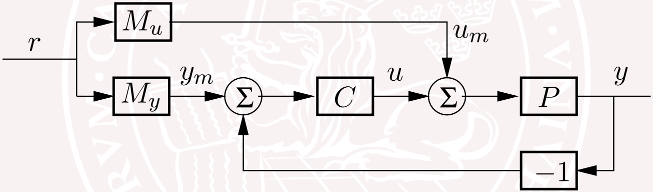
Design procedure:

- Design the feedback C to achieve
  - Small sensitivity to load disturbances d
  - Low injection of measurement noise n
  - High robustness to process variations
- Then design the feedforward F to achieve desired response to command signals r

For many problems in process control the load disturbance response is much more important than the set point response. The set point response is more important in motion control. Few textbooks and papers show more than set point responses.

Many Versions of 2DOF

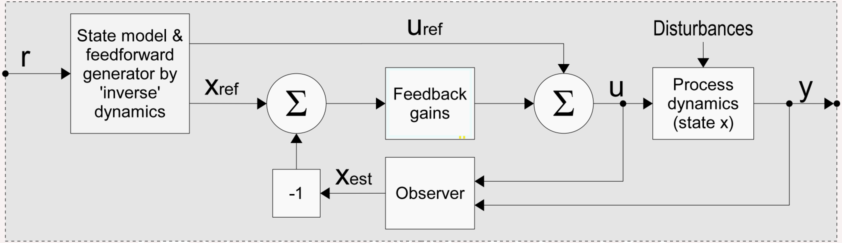




For linear systems all 2DOF configurations have the same properties.
For the systems above we have

$$CF = M_u + CM_y$$

Feedforward (state-space)



- State estimation
- State references
- FF of u_{ref}

Exact Feedback Linearization

Idea: Find state feedback $u = u(x, v)$ so that the nonlinear system

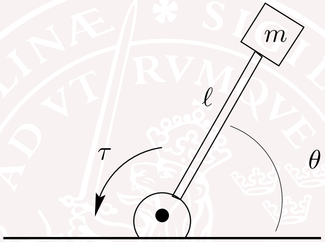
$$\dot{x} = f(x) + g(x)u$$

turns into linear system

$$\dot{x} = Ax + Bv$$

and then apply linear control design method.

Exact linearization: example [one-link robot]



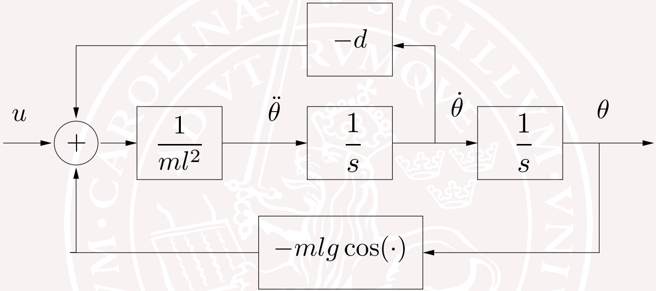
$$m\ell^2\ddot{\theta} + d\dot{\theta} + m\ell g\cos\theta = u$$

where d is the viscous damping.

The control $u = \tau$ is the applied torque

Design state feedback controller $u = u(x)$ with $x = (\theta, \dot{\theta})^T$

Block diagram



$$m\ell^2\ddot{\theta} + d\dot{\theta} + m\ell g\cos\theta = u$$

$$\ddot{\theta} = \frac{1}{m\ell^2}\left(-d\dot{\theta} - m\ell g\cos\theta + u\right)$$

Introduce new control variable v and let

$$u = m\ell^2 v + d\dot{\theta} + m\ell g \cos\theta$$

Then

$$\ddot{\theta} = v$$

Choose e.g. a PD-controller

$$v = v(\theta, \dot{\theta}) = k_p(\theta_{\mathsf{ref}} - \theta) - k_d\dot{\theta}$$

This gives the closed-loop system:

$$\ddot{\theta} + k_d\dot{\theta} + k_p\theta = k_p\theta_{\mathsf{ref}}$$

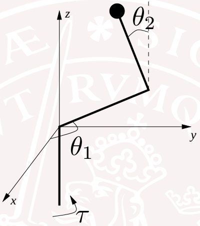
Hence, $u = m\ell^2[k_p(\theta - \theta_{\mathsf{ref}}) - k_d\dot{\theta}] + d\dot{\theta} + m\ell g \cos\theta$

Warning: Exact feedback linearization of non-minimum-phase systems corresponds to unstable pole-zero cancellations!

Example: Linear systems

Both zeros and poles in the right half plane impose *fundamental limitations*, even though zeros "does not affect the stability of the plant".

Multi-link robot (n-joints)



General form

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + G(\theta) = u, \qquad \theta \in R^n$$

Called *fully* actuated if n indep. actuators,

M	$n \times n$ inertia matrix, $M = M^T > 0$
$C\dot{\theta}$	$n \times 1$ vector of centrifugal and Coriolis forces
G	$n \times 1$ vector of gravitation terms

Computed torque

The computed torque
(also known as *exact linearization, dynamic inversion* , etc.)

$$\begin{aligned} u &= M(\theta)v + C(\theta,\dot{\theta})\dot{\theta} + G(\theta) \\ v &= K_p(\theta_{ref} - \theta) - K_d\dot{\theta}, \end{aligned} \tag{1}$$

gives closed-loop system

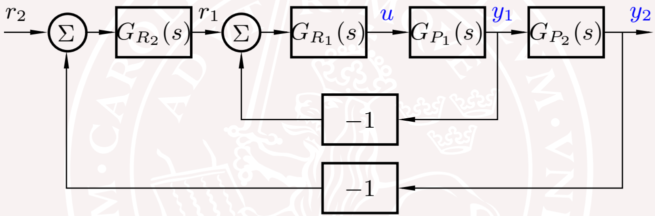
$$\ddot{\theta} + K_d\dot{\theta} + K_p\theta = K_p\theta_{Ref}$$

The matrices K_d and K_p can be chosen diagonal (no cross-terms) and then this decouples into n independent second-order equations.

Identify FF and FB-part!

Cascade control

For systems with one control signal and many outputs:

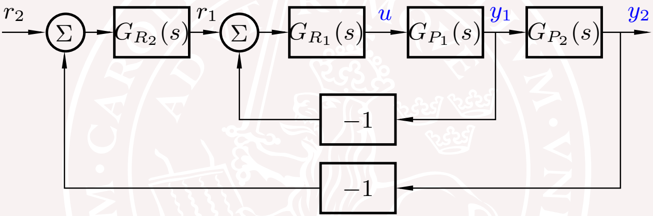


- $G_{R_1}(s)$ controls the subsystem $G_{P_1}(s)$ ($\Rightarrow G_{y_1 r_1}(s) \approx 1$)
- $G_{R_2}(s)$ controls the subsystem $G_{P_2}(s)$

Often used in motion control, e.g., robotics, with cascaded velocity and position controllers, BUT should have velocity reference and acc./torque feedforward!!

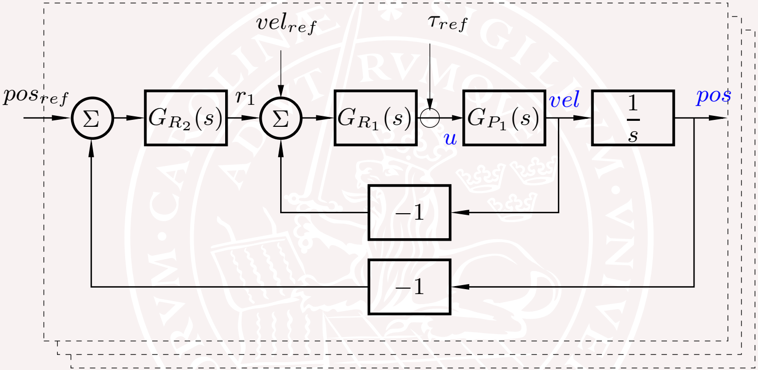
Cascade control

For systems with one control signal and many outputs:



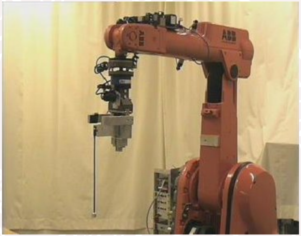
- $G_{R_1}(s)$ controls the subsystem $G_{P_1}(s)$ $(\Rightarrow G_{y_1 r_1}(s) \approx 1)$
- $G_{R_2}(s)$ controls the subsystem $G_{P_2}(s)$

Often used in motion control, e.g., robotics, with cascaded velocity and position controllers, BUT should have velocity reference and acc./torque feedforward!!


pos_{ref}
Σ
$G_{R_2}(s)$
r_1
vel_{ref}
Σ
$G_{R_1}(s)$
τ_{ref}
u
$G_{P_1}(s)$
vel
$\frac{1}{s}$
pos
-1
-1

Example of couplings

Example: Couplings and interaction: "good"/"bad"


[movie]

"Robot Furuta pendulum": Underactuated – coupling as control action

"Ordinary" Robot control:

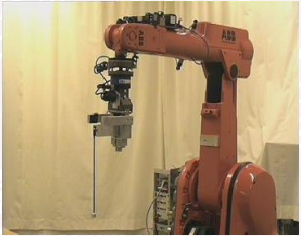
Often cascaded PI-controllers for each joint (inner velocity and outer position loop)

Feedforward for

- **disturbance rejection** between joints
- velocity and torque reference (improved tracking!)

Example of couplings

Example: Couplings and interaction: "good"/"bad"



[movie]

"Robot Furuta pendulum": Underactuated – coupling as control action

"Ordinary" Robot control:

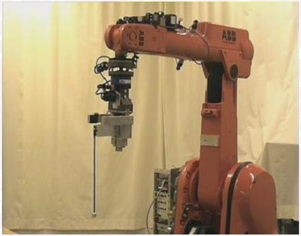
Often cascaded PI-controllers for each joint
(inner velocity and outer position loop)

Feedforward for

- **disturbance rejection** between joints
- velocity and torque reference (improved tracking!)

Example of couplings

Example: Couplings and interaction: "good"/"bad"



[movie]

"Robot Furuta pendulum": Underactuated – coupling as control action

"Ordinary" Robot control:

Often cascaded PI-controllers for each joint

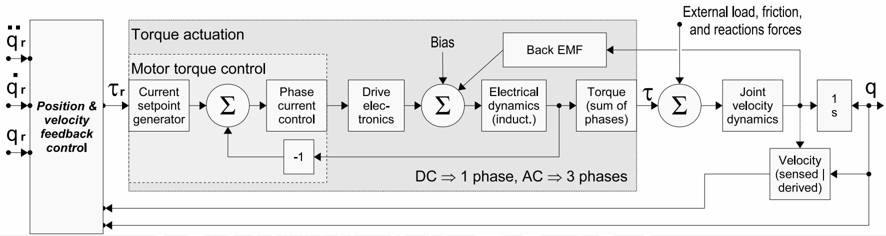
(inner velocity and outer position loop)

Feedforward for

- **disturbance rejection** between joints
- velocity and torque reference (improved tracking!)

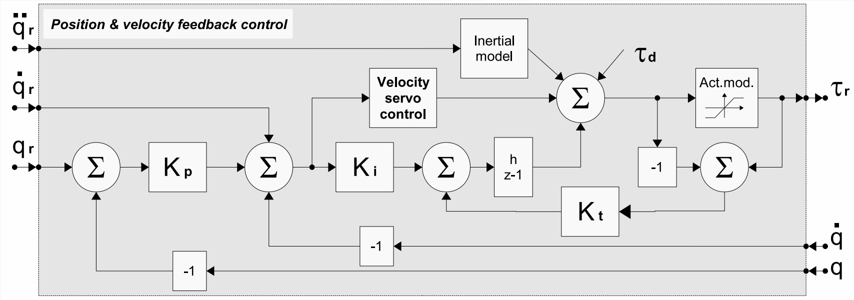
SOLVE PROBLEMS WHERE THEY OCCUR!

Generic drive hierarchy



Inner “current/torque loop” extremely important!

Cascade for motion control



- Integral part in inner loop
- Anti-windup

Challenge

ABB-design contest.:

2005 *Robust Control Of A Flexible Manipulator Arm: A Benchmark Problem*

2008 *A Benchmark Problem for Robust Control of a Multivariable Nonlinear Flexible Manipulator*

See more at

`http://www.control.isy.liu.se/~stig/mimo.html`

[SISO.pdf]

Kinematics

Consider an *n*-degree-of-freedom robot:

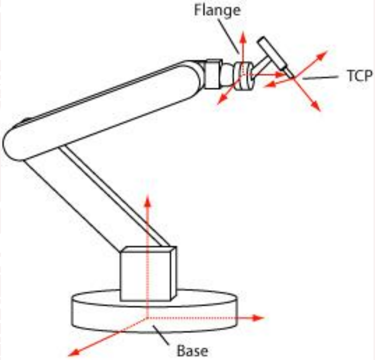
Direct/Forward kinematics

$$T(q) = \begin{bmatrix} R(q) & p(q) \\ \mathbf{0} & 1 \end{bmatrix}$$

where

$$\mathbf{q} = \begin{bmatrix} q_1 & \dots & q_n \end{bmatrix}^T$$

are the joint variables



See e.g. Peter Corke's matlab toolbox

Jacobians

Differental kinematics:

Find relationship between

joint velocities $\dot{q}$
and
the *end-effector linear and angular velocities* $(\dot{p},\ \omega)$
usually w.r.t. base coordinate system

Geometric Jacobian

$$\dot{p} = J_P(q)\dot{q}$$

$$\omega = J_O(q)\dot{q}$$

$$J_G = \begin{bmatrix} J_P(q) \\ J_O(q) \end{bmatrix}$$

Jacobians

Differental kinematics:

Find relationship between

joint velocities $\dot{q}$
and
the *end-effector linear and angular velocities* $(\dot{p},\, \omega)$
usually w.r.t. base coordinate system

Geometric Jacobian

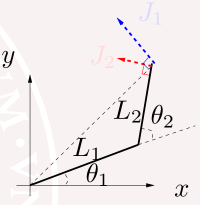
$$\dot{p} = J_P(q)\dot{q}$$
$$\omega = J_O(q)\dot{q}$$

$$J_G = \begin{bmatrix} J_P(q) \\ J_O(q) \end{bmatrix}$$

2 DoF-example

Example:

$$x = L_1\cos(\theta_1) + L_2\cos(\theta_1+\theta_2)$$
$$y = L_1\sin(\theta_1) + L_2\sin(\theta_1+\theta_2)$$



Differentiating wrt time

$$\dot{x} = -L_1\sin(\theta_1)\dot{\theta}_1 - L_2\sin(\theta_1+\theta_2)(\dot{\theta}_1+\dot{\theta}_2)$$
$$\dot{y} = L_1\cos(\theta_1)\dot{\theta}_1 + L_2\cos(\theta_1+\theta_2)(\dot{\theta}_1+\dot{\theta}_2)$$

If J_1 and J_2 are not parallell we can achieve any TCP velocity $(\dot{x}, \dot{y})$ by choosing $(\dot{\theta}_1, \dot{\theta}_2)$

$$\begin{bmatrix}\dot{x}\\ \dot{y}\end{bmatrix} = \underbrace{\begin{bmatrix}*\\ *\end{bmatrix}}_{J_1(\theta_1,\theta_2)}\dot{\theta}_1 + \underbrace{\begin{bmatrix}*\\ *\end{bmatrix}}_{J_2(\theta_1,\theta_2)}\dot{\theta}_2$$

$$= \begin{bmatrix}J_1 & J_2\end{bmatrix}\begin{bmatrix}\dot{\theta}_1\\ \dot{\theta}_2\end{bmatrix}$$

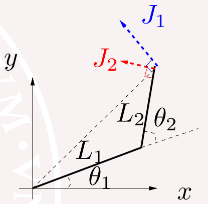
2 DoF-example

Example:

$$x = L_1\cos(\theta_1) + L_2\cos(\theta_1+\theta_2)$$
$$y = L_1\sin(\theta_1) + L_2\sin(\theta_1+\theta_2)$$

Differentiating wrt time

$$\dot{x} = -L_1\sin(\theta_1)\dot{\theta}_1 - L_2\sin(\theta_1+\theta_2)(\dot{\theta}_1+\dot{\theta}_2)$$
$$\dot{y} = L_1\cos(\theta_1)\dot{\theta}_1 + L_2\cos(\theta_1+\theta_2)(\dot{\theta}_1+\dot{\theta}_2)$$

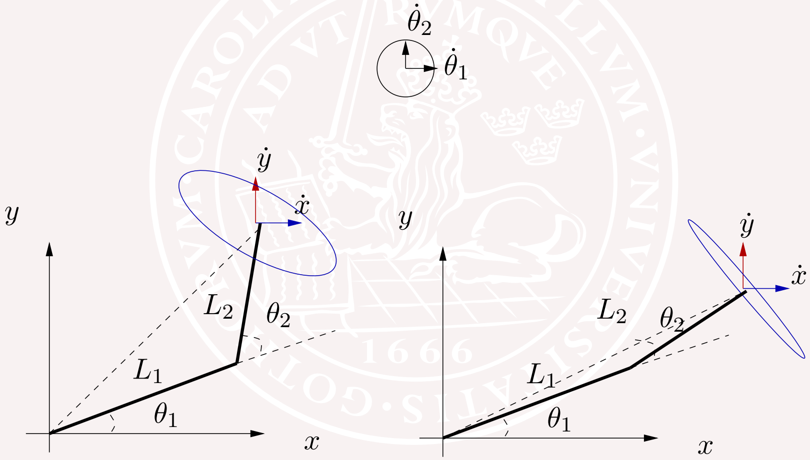


If J_1 and J_2 are not parallell we can achieve any TCP velocity $(\dot{x}, \dot{y})$ by choosing $(\dot{\theta}_1, \dot{\theta}_2)$

$$\begin{bmatrix}\dot{x}\\ \dot{y}\end{bmatrix} = \underbrace{\begin{bmatrix}*\\ *\end{bmatrix}}_{J_1(\theta_1,\theta_2)}\dot{\theta}_1 + \underbrace{\begin{bmatrix}*\\ *\end{bmatrix}}_{J_2(\theta_1,\theta_2)}\dot{\theta}_2$$
$$= \begin{bmatrix}J_1 & J_2\end{bmatrix}\begin{bmatrix}\dot{\theta}_1\\ \dot{\theta}_2\end{bmatrix}$$

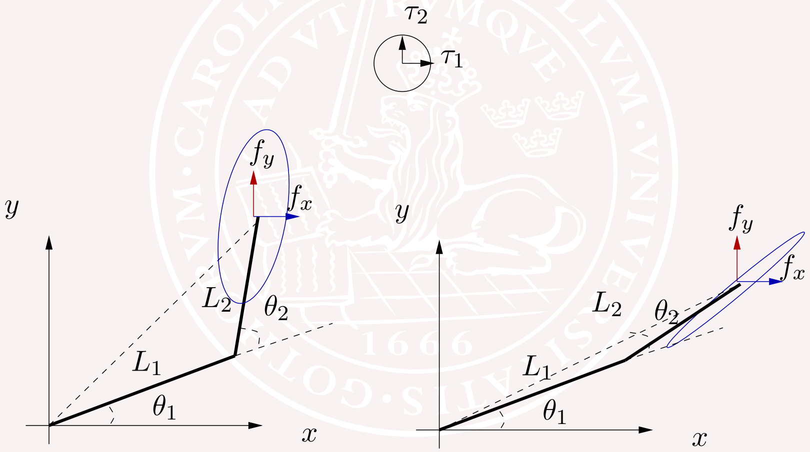
Manipulability ellipsoid

Depending on configuration (θ_1, θ_2) the joint velocities $(\dot{\theta}_1, \dot{\theta}_2)$ map very differently to $(\dot{x}, \dot{y})$.



Manipulability ellipsoid - forces

Depending on configuration (θ_1, θ_2) the motor torqus (τ_1, τ_2) map very differently to (f_x, f_y).



Jacobians cont'd

Analytical Jacobian

Assume end-effector *pose* $x = k(q)$ is expressed as position p and minimal representation of *orientation* ϕ (e.g., Euler angles ZYZ).

Note: In general $\dot{\phi} \neq \omega$!!

$$\dot{x} = \begin{bmatrix} \dot{p} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} J_P(q) \\ J_\phi(q) \end{bmatrix} \dot{q} = J_A(q)\dot{q} = \frac{\partial k(q)}{\partial q}\dot{q}$$

Jacobians cont'd

Analytical Jacobian

Assume end-effector *pose* $x = k(q)$ is expressed as position p and minimal representation of *orientation* ϕ (e.g., Euler angles ZYZ).

Note: In general $\dot{\phi} \neq \omega$!!

$$\dot{x} = \begin{bmatrix} \dot{p} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} J_P(q) \\ J_\phi(q) \end{bmatrix} \dot{q} = J_A(q)\dot{q} = \frac{\partial k(q)}{\partial q}\dot{q}$$

Kinematic singularity

$$v = \begin{bmatrix} \dot{p} \\ \omega \end{bmatrix} \dot{q} = J_G(q)\dot{q}$$

is said to have a *kinematic singularity* at q when $J_G(q)$ is rank-deficient.

Close to a singularity small velocities in *operational space* may map to very large velocities in *joint space*

- Boundary singularities
  - Example: stretched out
- Internal singularities
  - Example: axis aligend

[movie]

Kinematic singularity

$$v = \begin{bmatrix} \dot{p} \\ \omega \end{bmatrix} \dot{q} = J_G(q)\dot{q}$$

is said to have a *kinematic singularity* at q when $J_G(q)$ is rank-deficient.

Close to a singularity small velocities in *operational space* may map to very large velocities in *joint space*

- Boundary singularities
  - Example: stretched out
- Internal singularities
  - Example: axis aligend

[movie]

Kinematic singularity

$$v = \begin{bmatrix} \dot{p} \\ \omega \end{bmatrix} \dot{q} = J_G(q)\dot{q}$$

is said to have a *kinematic singularity* at q when $J_G(q)$ is rank-deficient.

Close to a singularity small velocities in *operational space* may map to very large velocities in *joint space*

- Boundary singularities
  - Example: stretched out
- Internal singularities
  - Example: axis aligend

[movie]

Kineto-static duality: velocity - force

At static equilibrium:

By the principle of virtual work the similar relations between

joint torques $\tau = [\tau_1, ..., \tau_n]^T$

and

end-effector forces and torques $\gamma = [f^T, \mu^T]^T$ hold according to

$$\tau = J_G^T(q)\gamma$$

Velocity manipulability ellipsoid

$$\dot{q}^T \dot{q} = 1 \leftrightarrow v^T \left(J_G(q) J_G(q)^T \right)^{-1} v = 1$$

Force manipulability ellipsoid

$$\tau \tau^T = 1 \leftrightarrow \gamma^T \left(J_G(q) J_G(q)^T \right) \gamma = 1$$

Same principal axes, whereas "inverse magnitudes"

Caveat: Don't only look at linearization [?]

Velocity manipulability ellipsoid

$$\dot{q}^T \dot{q} = 1 \leftrightarrow v^T \left(J_G(q) J_G(q)^T \right)^{-1} v = 1$$

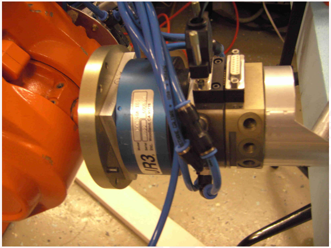
Force manipulability ellipsoid

$$\tau \tau^T = 1 \leftrightarrow \gamma^T \left(J_G(q) J_G(q)^T \right) \gamma = 1$$

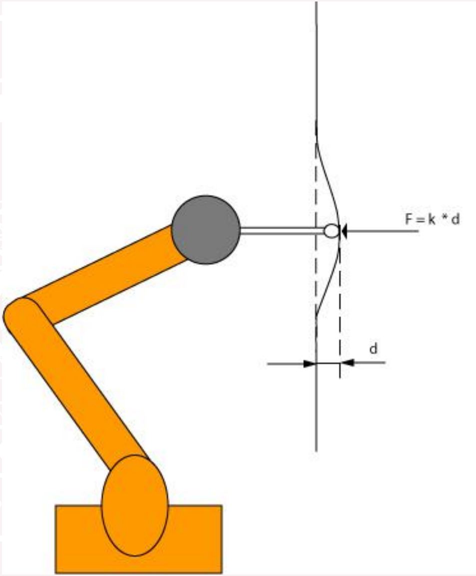
Same principal axes, whereas "inverse magnitudes"

Caveat: Don't only look at linearization [**?**]

Force interaction with environment



Wrist mounted 6DOF force/torque sensor



Force interaction with environment

Course overview:

- Compliant motion ("Soft servo")
  - Detune control parameters
  - Contact forces as load disturbances
- Direct force control
  - Example: close force control loop outside pos/vel controlled robot
- Impedance/admittance control

  [Hogan, 1985]
  - specify e.g., damping and stiffness
- Hybrid force/position
- etc.

Contact force control

[movie]

Deburring: Force control on aluminum castings

Impedance control

[Hogan, 1985]: *"...unified approach to the control of a manipulator applicable to free motions, kinematically constrained motions, and dynamic interaction of the manipulator and its environment"*

Relates output velocity (position) with input force

"Use feedback to get error dynamics behave as a tunable spring-damper system"

$$Z(s) = \frac{F(s)}{V(s)} = \frac{F(s)}{sX(S)}$$

$$F(s) = (M_d s^2 + K_v s + K_p)\tilde{X}(s)$$

Movie:

Impedance control

[Hogan, 1985]: *“...unified approach to the control of a manipulator applicable to free motions, kinematically constrained motions, and dynamic interaction of the manipulator and its environment"*

Relates output velocity (position) with input force

"Use feedback to get error dynamics behave as a tunable spring-damper system"

$$Z(s) = \frac{F(s)}{V(s)} = \frac{F(s)}{sX(S)}$$
$$F(s) = (M_d s^2 + K_v s + K_p)\tilde{X}(s)$$

Movie:

[movie]

Admittance control used for assembley [ABB CRC, US]

Constrained motions

Interaction control specifications

Natural constraints: The environment imposes constraints on forces or motions

Artificial constraints: The manipulator can control only those directions which are not natural constraints. Artificial constraints are the desired reference values

Constrained motions

Interaction control specifications

Natural constraints: The environment imposes constraints on forces or motions

Artificial constraints: The manipulator can control only those directions which are not natural constraints. Artificial constraints are the desired reference values

Contrained motions

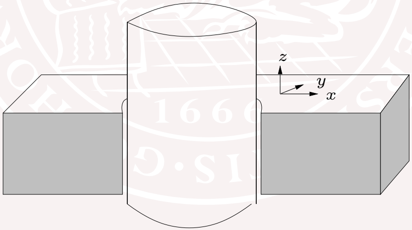
Example: Peg-in-the-hole problem

Natural	Artificial
$\dot{p}_x$	f_x
$\dot{p}_y$	f_y
ω_x	μ_x
ω_y	μ_y
f_z	$\dot{p}_z$
μ_z	ω_x

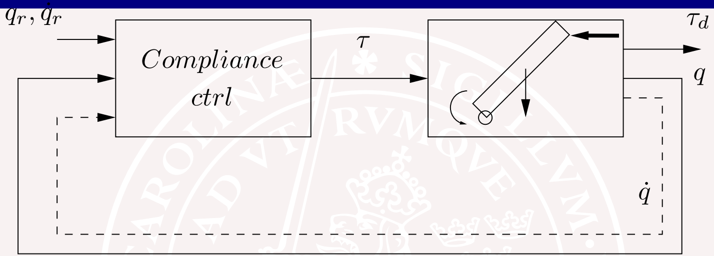
With

$\dot{p}$: linear velocity $\quad \omega$: angular velocity

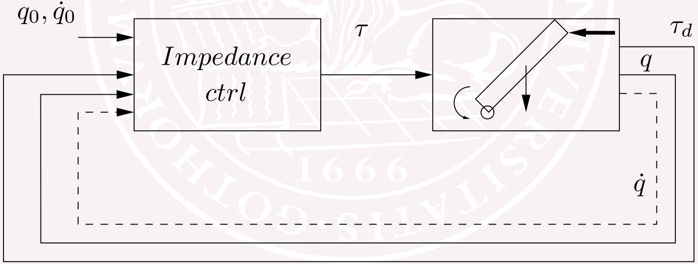
f: contact force $\quad \mu$: torque



Comparison

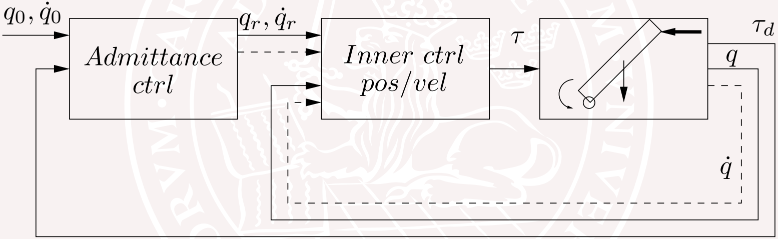


$q_r, \dot{q}_r$
$Compliance$
$ctrl$
τ
τ_d
q
$\dot{q}$



$q_0, \dot{q}_0$
$Impedance$
$ctrl$
τ
τ_d
q
$\dot{q}$

Comparison



$q_0, \dot{q}_0$
Admittance ctrl
$q_r, \dot{q}_r$
Inner ctrl pos/vel
τ
τ_d
q
$\dot{q}$

Bibliography — Force control

- Robot Force Control B. Siciliano and L. Villani, Kluwer academic publishers, 1999

Bilateral Teleoperation:

- P.F. Hokayem and M.W. Spong, "Bilateral Teleoperation: An Historical Survey," Automatica, December, 2006.

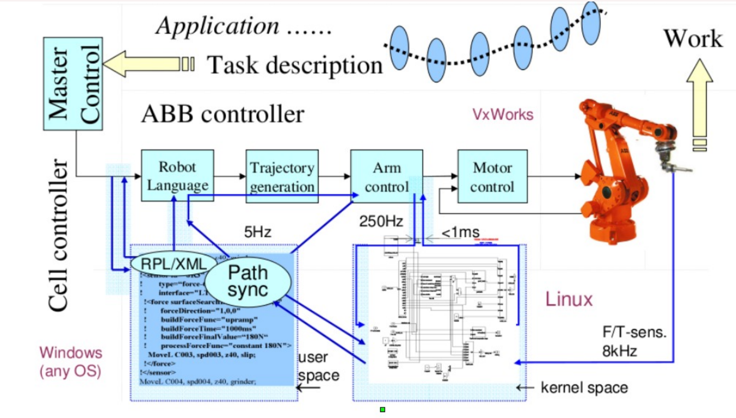
Path generation

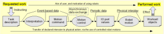
Need both geometric and dynamic path planning.

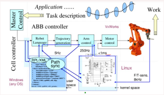
Different constraints on path generation.

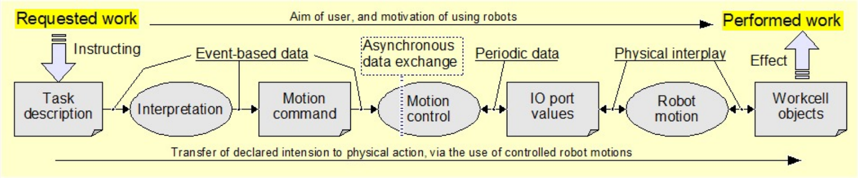
- Smooth (little excitation of resonant modes)
  Example: want *e*.g., continuous accelaration
- As fast as possible, but must stay on path
- Constraints on velocity/accelation
  - Joint space
  - Cartesian space
- Constraints on maximum torque

See also "Planning Algorithms" by S. M. LaValle
`http://planning.cs.uiuc.edu/`


Application ......
Master Control
Task description
Work
ABB controller
VxWorks
Robot Language
Trajectory generation
Arm control
Motor control
Cell controller
5Hz
250Hz
<1ms
RPL/XML
Path sync
Linux
F/T-sens. 8kHz
Windows (any OS)
user space
kernel space
forceDirection="1,0,0"
buildForceFunc="upramp"
buildForceTime="1000ms"
buildForceFinalValue="180N"
processForceFunc="constant 180N">
MoveL C003, spd003, z40, slip;
</force>
</sensor>
MoveL C004, spd004, z40, grinder;







Need to cross the “boundary” when feedback and HRI

Application: Path generation

This work is conducted at the Robotics Lab, Dept of Automatic Control, Lund, in cooperation with Rolf Johansson, Magnus Linderoth, Martin Hast, Andreas Stolt, Klas Nilsson, Henrik Nilsson and Björn Olofsson.

Motivation

Industrial robots are used in many application areas where fast path tracking is required.

“Traditional robotics”/”free motion” entitles

Separation between

- Path/Trajectory generation
  - modeling
  - constraints
  - path/trajectory
- Path/Trajectory tracking

“Fanta Challenge” [movie]

Motivation

Industrial robots are used in many application areas where fast path tracking is required.

“Traditional robotics”/”free motion” entitles

Separation between

- Path/Trajectory generation
  - modeling
  - constraints
  - path/trajectory

- Path/Trajectory tracking

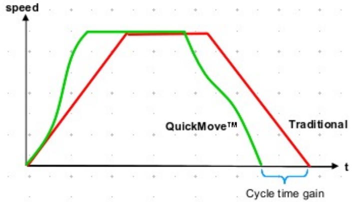
“Fanta Challenge” [movie]

Commercial arguments

World best Motion Control

- **TrueMove™** ensures that the motion path followed by the robot is the same as the programmed path – regardless of the robot speed

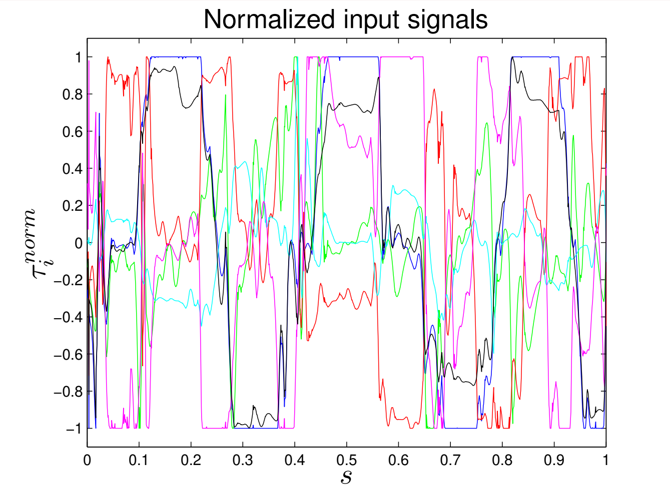
- **QuickMove™** minimizes cycle time by maximizing acceleration at every moment.



Robotics - 5 Oct 2006

ABB

Experimental results



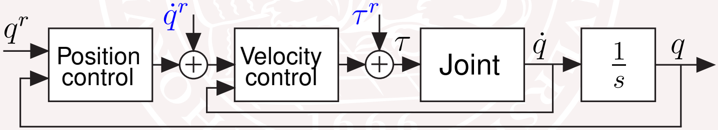
One of the actuator signals 'always' maximized ('not saturated')

Servo control structure

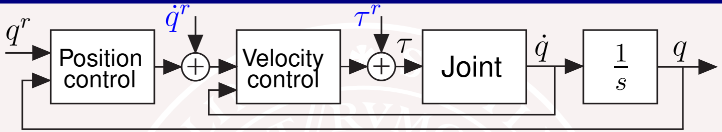
Most industrial robot systems have a cascaded control structure (for each joint)

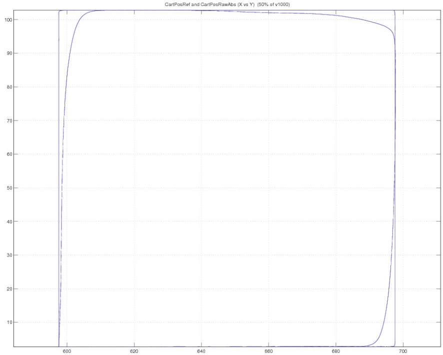
Performance comes from

- Modeling (flexible modes, interaction, decoupling)
- Trajectory generation



Servo control structure (with/without FF)

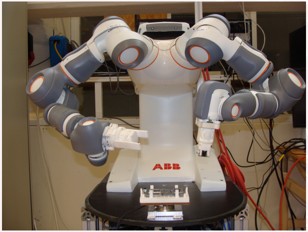


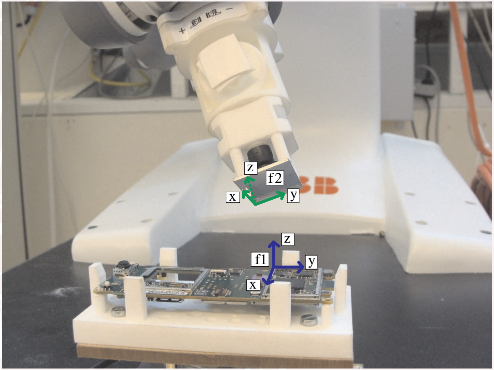


v=500mm/s

Sensor-based task execution

Example: Force feedback control for assembly





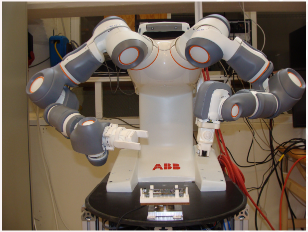
Differs a lot from free motion

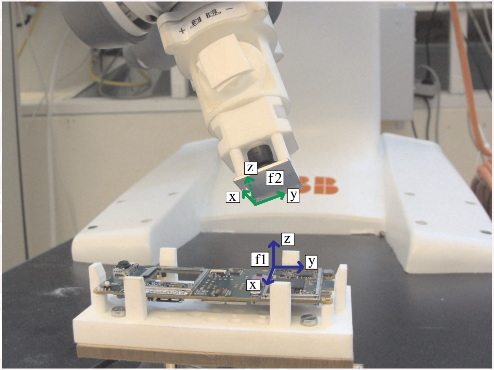
Performance relies on feedback to larger extent

[Stolt *et al* (2011) (movie)]

Sensor-based task execution

Example: Force feedback control for assembly





Differs a lot from free motion

Performance relies on feedback to larger extent

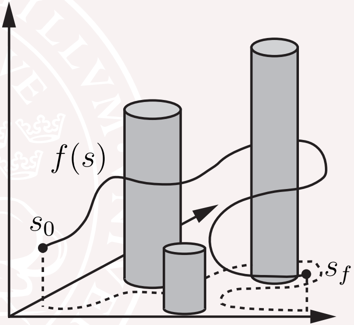
[Stolt *et al* (2011) (movie)]

Previous work

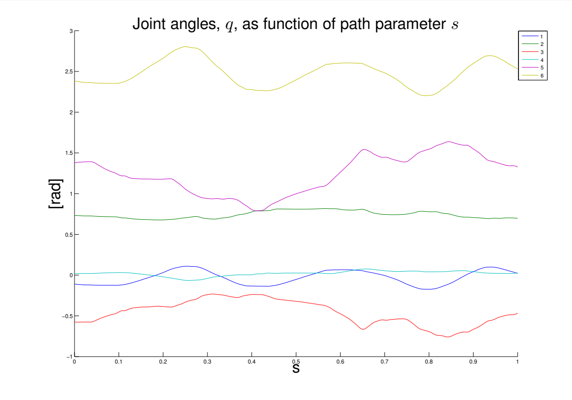
- Industrial robotics, modeling and optimal path generation/tracking has long tradition at Depts. of Automatic Control
- M. Norrlöf and S. Moberg, LiU and ABB Robotics
- Follow path despite disturbances and uncertainties
- Constraints from external sensors
- Based on PhD thesis *Path Constrained Robot Control* by O. Dahl in the 1990s
  - Formulation of optimization problem and path velocity control
  - Optimization with Optimica/Jmodelica.org
  - Relaxed model formulations for convex optimization problem (real-time aspects)
  - Experiments in ORCA
  - External sensing / “allowed deviations”

Robot dynamics and path parametrization

- Rigid body model of the robot:
  $\tau = M\ddot{q} + C(q,\dot{q})\dot{q} + D\dot{q} + g(q)$
- Introduce path f parametrized in path parameter $s \in [s_0, s_f]$
- Path tracking requirement $q = f(s)$
  $\Longrightarrow$ robot dynamics can be expressed in s, $\dot{s}$ and $\ddot{s}$:
  $\tau(s) = \Gamma_1(s)\ddot{s} + \Gamma_2(s,\dot{s})$



Path Representation

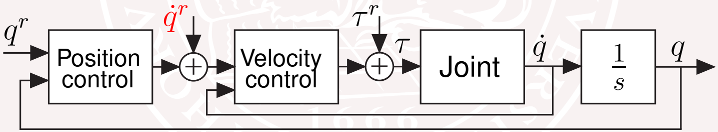


Introduce the scalar path parameter $s(t)$

$$q_{ref}(t) = f(s(t)), \qquad s(0) = s_0 = 0 \leq s \leq s(t_f) = s_f = 1$$

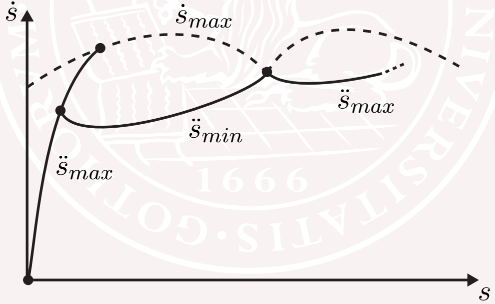
Robot model

- Identification of rigid body model with torques as input non-trivial task
- Experiments on the robot system showed nonlinear effects when controlling with the torques as control signals
- Velocity references to each joint can be used in the cascaded control structure in the robot system



Time-optimal path tracking

- The phase-plane method is the traditional method for time-optimal path tracking with input constraints [Bobrow *et al.*, 1985; Shin & McKay, 1985]
- Utilizes integration in the s–$\dot{s}$ phase plane
- More general cost functions and constraints require more general solution methods



Standard Minimum Time Optimization

- Dynamics (Here simplistic model)

$$K_i \tau_i = \ddot{q}_i T_i + \dot{q}_i, \qquad i = 1, 2, \ldots, 6$$

- Constraints

$$\tau_i^{min} \le \tau_i \le \tau_i^{max}$$

$$f(s(t)) = \begin{bmatrix} f_1(s(t)) \\ \vdots \\ f_6(s(t)) \end{bmatrix}, \quad q(t) = f(s(t))$$

- Boundary conditions

$$q_i(0) = f(0), \quad \dot{q}_i(0) = 0, \qquad q_i(t_f) = f(t_f), \quad \dot{q}_i(t_f) = 0$$

- Cost function

$$\min_{\tau} t_f = \min_{\tau} \int_0^{t_f} 1 \mathrm{d}t$$

Reformulating the Original Problem

Using the chain rule $\frac{df}{dt} = \frac{df}{ds}\frac{ds}{dt}$ to reduce the number of states

- Dynamics

$$\frac{ds}{dt} = \dot{s}, \quad \frac{d\dot{s}}{dt} = \ddot{s}$$

- Constraints

$$\tau_i^{min} \leq \tau_i \leq \tau_i^{max}$$

$$K_i\tau_i = (f_i''(s)\dot{s}^2 + f_i'(s)\ddot{s})T_i + f_i'(s)\dot{s}, \qquad i = 1, 2, \ldots, 6$$

- Boundary conditions

$$s(0) = s_0, \quad \dot{s}(0) = 0, \qquad s(t_f) = s_f, \quad \dot{s}(t_f) = 0$$

- Cost function

$$\min_{\ddot{s}(t)} t_f = \min_{\ddot{s}(t)} \int_0^{t_f} 1 \mathrm{d}t$$

Reformulating the Original Problem

Using the chain rule $\frac{df}{dt} = \frac{df}{ds}\frac{ds}{dt}$ to reduce the number of states

- Dynamics

$$\frac{ds}{dt} = \dot{s}, \quad \frac{d\dot{s}}{dt} = \ddot{s}$$

- Constraints

$$\tau_i^{min} \leq \tau_i \leq \tau_i^{max}$$

$$K_i \tau_i = (f_i''(s)\dot{s}^2 + f_i'(s)\ddot{s})T_i + f_i'(s)\dot{s}, \qquad i = 1, 2, \ldots, 6$$

- Boundary conditions

$$s(0) = s_0, \quad \dot{s}(0) = 0, \qquad s(t_f) = s_f, \quad \dot{s}(t_f) = 0$$

- Cost function

$$\min_{\ddot{s}(t)} t_f = \min_{\ddot{s}(t)} \int_0^{t_f} 1 \mathrm{d}t$$

Reformulating the Original Problem

Using the chain rule $\dfrac{df}{dt} = \dfrac{df}{ds}\dfrac{ds}{dt}$ to reduce the number of states

- Dynamics

$$\frac{ds}{dt} = \dot{s}, \quad \frac{d\dot{s}}{dt} = \ddot{s}$$

- Constraints

$$\tau_i^{min} \leq \tau_i \leq \tau_i^{max}$$

$$K_i\tau_i = (f_i''(s)\dot{s}^2 + f_i'(s)\ddot{s})T_i + f_i'(s)\dot{s}, \qquad i = 1, 2, \ldots, 6$$

- Boundary conditions

$$s(0) = s_0, \quad \dot{s}(0) = 0, \qquad s(t_f) = s_f, \quad \dot{s}(t_f) = 0$$

- Cost function

$$\min_{\ddot{s}(t)} t_f = \min_{\ddot{s}(t)} \int_0^{t_f} 1 \mathrm{d}t$$

Reformulating the Original Problem

Using the chain rule $\dfrac{df}{dt} = \dfrac{df}{ds}\dfrac{ds}{dt}$ to reduce the number of states

- Dynamics

$$\frac{ds}{dt} = \dot{s}, \quad \frac{d\dot{s}}{dt} = \ddot{s}$$

- Constraints

$$\tau_i^{min} \leq \tau_i \leq \tau_i^{max}$$

$$K_i\tau_i = (f_i''(s)\dot{s}^2 + f_i'(s)\ddot{s})T_i + f_i'(s)\dot{s}, \qquad i = 1, 2, \ldots, 6$$

- Boundary conditions

$$s(0) = s_0, \quad \dot{s}(0) = 0, \qquad s(t_f) = s_f, \quad \dot{s}(t_f) = 0$$

- Cost function

$$\min_{\ddot{s}(t)} t_f = \min_{\ddot{s}(t)} \int_0^{t_f} 1\mathrm{d}t$$

Reformulation to a Fix-Time Problem

Introduce $x_1(s) = \dfrac{\dot{s}^2}{2}$

- Dynamics

$$\frac{dx_1}{ds} = \ddot{s}(s) = u$$

- Constraints

$$\tau_i^{min} \leq \tau_i \leq \tau_i^{max}$$

$$K_i \tau_i = (f_i''(s) 2x_1 + f_i'(s) u) T_i + f_i'(s)\sqrt{2x_1}, \qquad i = 1, 2, \ldots, 6$$

- Boundary conditions

$$x_1(s_0) = 0, \quad x_1(s_f) = 0$$

- Cost function

$$\min_{\tau(t)} \int_0^{t_f} 1 \mathrm{d}t = \min_{\ddot{s}} \int_{s_0}^{s_f} \frac{1}{\dot{s}} \mathrm{d}s = \min_{u(s)} \int_{s_0}^{s_f} \frac{1}{\sqrt{2x_1}} \mathrm{d}s$$

Reformulation to a Fix-Time Problem

Introduce $x_1(s) = \frac{\dot{s}^2}{2}$

- Dynamics

$$\frac{dx_1}{ds} = \ddot{s}(s) = u$$

- Constraints

$$\tau_i^{min} \leq \tau_i \leq \tau_i^{max}$$

$$K_i \tau_i = (f_i''(s) 2x_1 + f_i'(s) u) T_i + f_i'(s)\sqrt{2x_1}, \qquad i = 1, 2, \ldots, 6$$

- Boundary conditions

$$x_1(s_0) = 0, \quad x_1(s_f) = 0$$

- Cost function

$$\min_{\tau(t)} \int_0^{t_f} 1 \mathrm{d}t = \min_{\ddot{s}} \int_{s_0}^{s_f} \frac{1}{\dot{s}} \mathrm{d}s = \min_{u(s)} \int_{s_0}^{s_f} \frac{1}{\sqrt{2x_1}} \mathrm{d}s$$

Optimization formulation

- General optimization problem [Dahl, 1992]

$$\text{minimize} \quad \int_{s_0}^{s_f} \frac{1}{\sqrt{\beta(s)}} + \eta \sum_{i=1}^{n} |\tau_i'(s)| \, \mathrm{d}s$$

$$\text{such that} \quad \tau(s) = \Gamma_1(s)\ddot{s}(s) + \Gamma_2(s, \dot{s})$$

$$\dot{s}(s_0) = \dot{s}(s_f) = 0 \quad , \quad \beta'(s) = 2\ddot{s}(s) \quad , \quad \dot{s}(s) \geq 0$$

$$\tau_{min} \leq \tau(s) \leq \tau_{max} \quad , \quad \beta(s) = \dot{s}(s)^2$$

  with fixed final time and reduced number of states

- Convex optimization reformulation [Verscheure *et al.*, 2009]
  - Global optimum is ensured
  - Cartesian speed and acceleration constraints
  - Requires certain form of robot dynamics — no viscous friction — for affine constraints
  - Relaxed velocity cond. by [Ardeshiri *et al.*, 2011]

Optimization formulation

- General optimization problem [Dahl, 1992]

$$\text{minimize} \quad \int_{s_0}^{s_f} \frac{1}{\sqrt{\beta(s)}} + \eta \sum_{i=1}^{n} |\tau_i'(s)| \, \mathrm{d}s$$

$$\text{such that} \quad \tau(s) = \Gamma_1(s)\ddot{s}(s) + \Gamma_2(s,\dot{s})$$

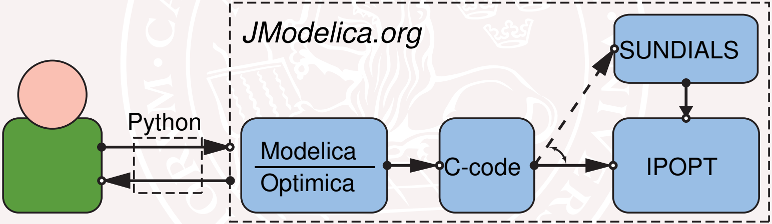
$$\dot{s}(s_0) = \dot{s}(s_f) = 0 \quad , \quad \beta'(s) = 2\ddot{s}(s) \quad , \quad \dot{s}(s) \geq 0$$

$$\tau_{min} \leq \tau(s) \leq \tau_{max} \quad , \quad \beta(s) = \dot{s}(s)^2$$

  with fixed final time and reduced number of states
- Convex optimization reformulation [Verscheure *et al.*, 2009]
  - Global optimum is ensured
  - Cartesian speed and acceleration constraints
  - Requires certain form of robot dynamics — no viscous friction — for affine constraints
  - Relaxed velocity cond. by [Ardeshiri *et al.*, 2011]

Solution of optimal path-tracking problem

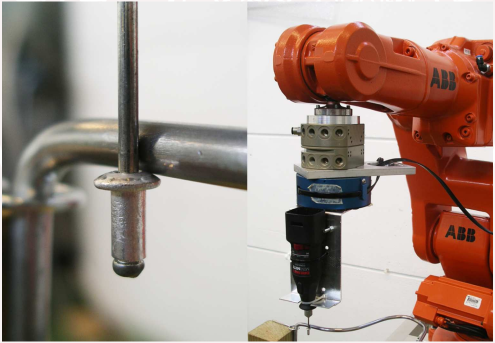
- JModelica.org [Åkesson *et al.*, 2010]
  - Modeling language Modelica and extension Optimica
  - System dynamics described by DAEs
  - Automatic transcription
  - IPOPT is used as internal solver of NLP



- YALMIP and SDP–solver in MATLAB for solution of convex optimization problem

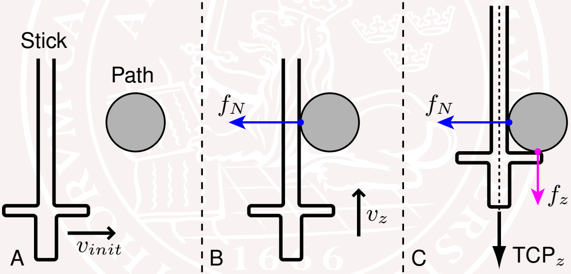
Path identification

- Contact-force approach in order to determine geometric motion of robot along the path
- Force and torque sensor attached to robot flange
- Path identification and reorientation of tool



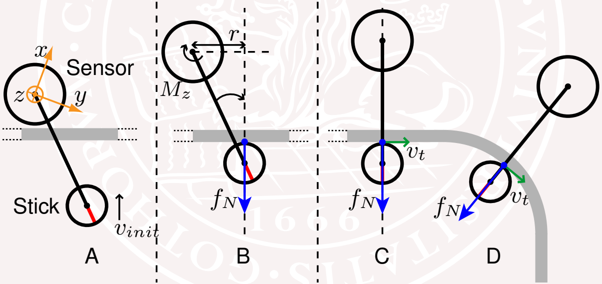
Force control

- PI–controllers keep the contact-forces constant in two directions
- One force controller for normal force and one for height control



Reorientation of tool

- Reorientation by controlling the torque around sensor z-axis to be constant
- Moving along the path according to tangential direction of the path perpendicular to normal force vector

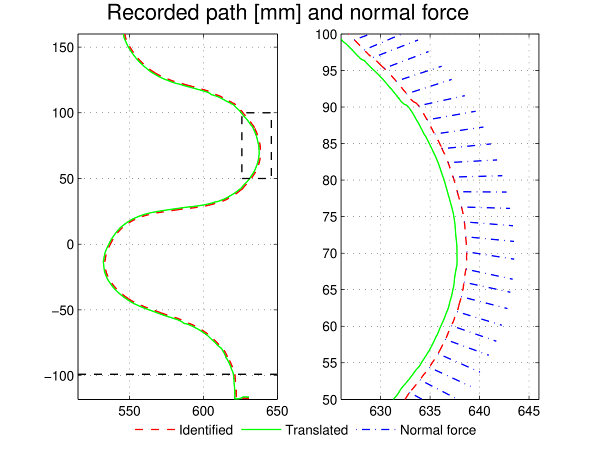


Movie

Movie showing the path identification procedure with contact-force control

Identification results

- Identified path and detail showing the normal force



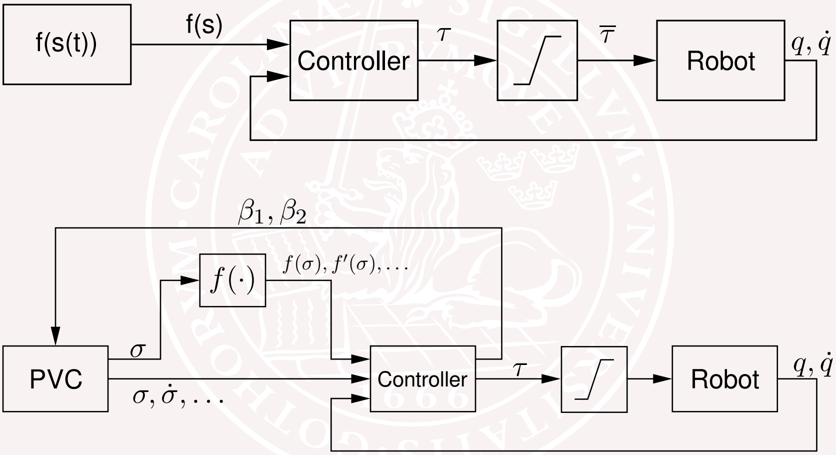
Online path tracking

- One input signal saturated in every time instance — low robustness to
  - Modeling errors
  - Disturbances
- Path velocity controller (PVC) by [Dahl, 1992]
- Modifies optimization result online based on feedback
- Path traverse described by σ corresponding to s
- Ordinary tracking controller parametrized as

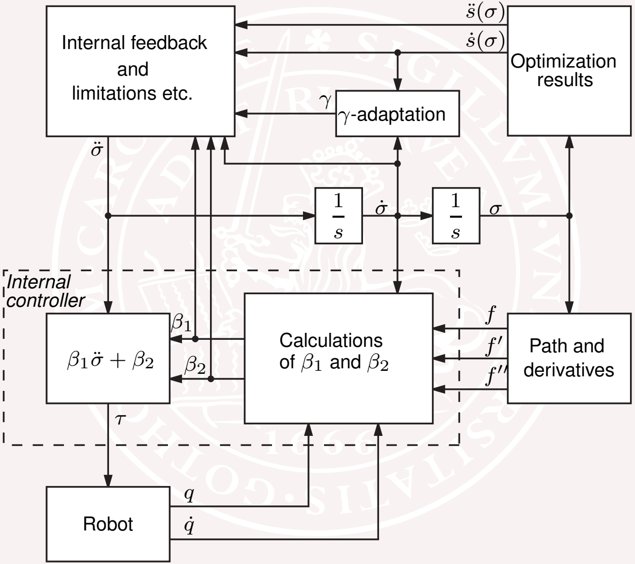
$$\tau = \beta_1 \ddot{\sigma} + \beta_2$$

- Limit optimized path acceleration $\ddot{s}(s)$ online to obtain $\ddot{\sigma}(\sigma)$

Structure of the PVC



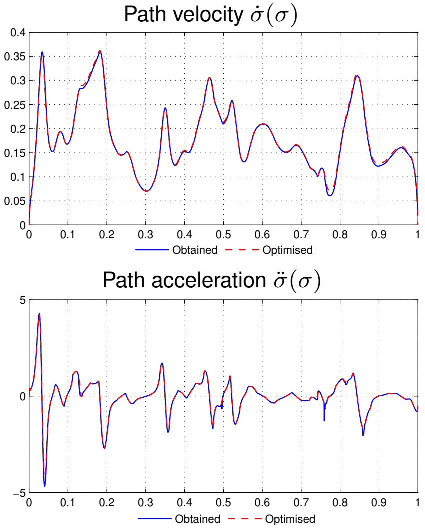
Structure of the PVC



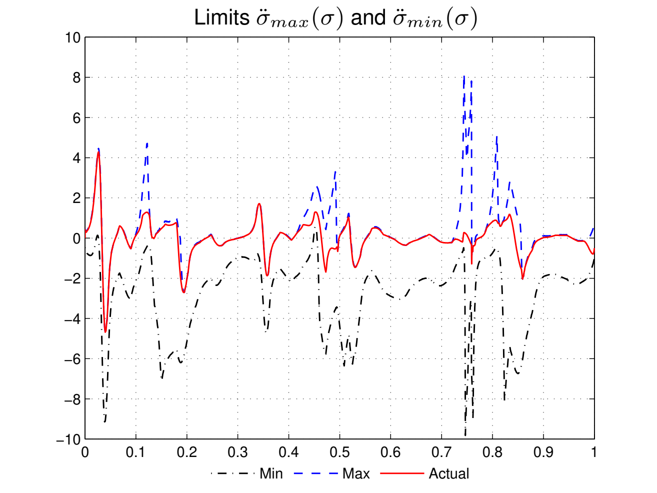
Movie

Movie showing the time-optimal tracking of the identified path

Experimental results

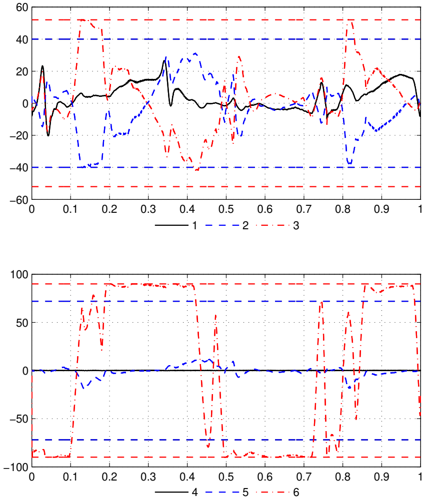


Experimental results

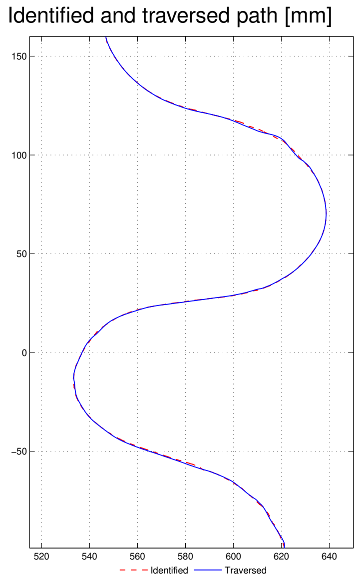


Experimental results

Joint velocity references $\dot{q}^r$ [deg/s] as function of σ



Experimental results



Conclusions and future work

- Part of this work
  - Path identification with contact-force control
  - Off-line optimization with JModelica.org
  - Online path tracking with feedback according to the PVC-structure
- Future work
  - External sensing integrated with PVC structure feedback
  - Exploit convex formulation (relaxations to velocity constraints) also for experiments on robot system

Bibliography

- Åkesson, J., K.-E. Årzén, M. Gäfvert, T. Bergdahl, and H. Tummescheit (2010): "Modeling and optimization with Optimica and JModelica.org—Languages and tools for solving large-scale dynamic optimization problem." *Computers and Chemical Engineering*, January.
- Bobrow, J. E., S. Dubowsky, and J. S. Gibson (1985): "Time-optimal control of robotic manipulator along specified paths." *The International Journal of Robotics Research*, **4:3**, pp. 3–17.
- Dahl, O. (1992): *Path Constrained Robot Control*. PhD thesis ISRN LUTFD2/TFRT--1038--SE, Department of Automatic Control, Lund University, Sweden.
- Shin, K. G. and N. D. McKay (1985): "Minimum-time control of robot manipulators with geometric path constraints." *IEEE Transactions on Automatic Control*, **AC-30:6**, pp. 531–541.

Bibliography cont'd

Verscheure, D., B. Demeulenaere, J. Swevers, J. De Schutter, and M. Diehl (2009): "Time-optimal path tracking for robots: A convex optimization approach." *IEEE Transactions on Automatic Control*, **54:10**, pp. 2318–2327.

T. Ardeshiri, M. Norrlöf, J. Löfberg, and A. Hansson (2011): "Convex Optimization approach for Time-Optimal Path Tracking of Robots with Speed Dependent Constraints" , *Proceedings of IFAC 2011*, Milan, Italy