Robot Programming and Control for Human Interaction
Alin Albu-Schäffer

TU München
Lehrstuhl für Steuerungs- und Regelungstechnik

Deutsches Zentrum für Luft- und Raumfahrt
Institut für Robotik und Mechatronik

Optional lecture, summer semester 2014
time: Thursdays, 12:00 – 13:30
Room: 02.09.023 NEW ROOM!!
The lecture treats aspects of robotic manipulation and human-robot interaction starting from the theoretical basis over implementation in simulation up to hands-on validation on the KUKA-DLR light-weight robot.
At the end of the lecture, the students should have gained not only theoretical knowledge but also first practical experience with compliant, torque controlled robots and their programming. Half of the lecture will be dedicated in block units to implementations in simulation and to hands-on experiments.

Recommended lectures:
„Robotics“, Darius Burschka
„Sensor based robotic manipulation and locomotion“, A. Albu-Schäffer
Contents of the Lecture

- Robot modelling and parameter identification
- Position control
- Torque control
- Cartesian Impedance Control
- Collision detection
- Reactive path generation
- State machines for task programming
Organisation

- 3 ETCS credits

- Examination: project result2, composed of:
  - simulation results
  - experimental results
  - oral presentation of results (10-15 min)

- Material: Slides, hand-out material, simulink models, java scripts

- Web: check website of the TUM chair
Tentative Schedule

10.04 lecture (90 min):
   introduction
   modelling & parameter identification

24.04 lecture (90 min):
   Position control
   Torque control
   Cartesian Impedance Control

08.05. lecture (90 min):
   Collision detection
   Reactive path generation

15.05. lecture (90 min):
   State machines for task programming
   Java API for KUKA light-weight robot

22.05 Simulation block unit (6 hours)
   Matlab/Simulink installation is needed on own notebook (TUM campus license)
   A robot control tutorial is run with implementation of main control concepts
   The students then have 3 weeks for completing the simulation problems

12.06 Experimental block unit (6 hours)
   Implementation of an interactive task on the JAVA API for the KUKA light-weight robot

19.06 Back-up Block for simulation / experiments

26.06 / 03.07 (90 min each):
   final presentations
Basics of Robotics – in a Nutshell - Reload

This material should be known from the „Robotics“ or „Sensor Based Robotics Manipulation and Locomotion“

homogeneous transformations

\[ T = \begin{bmatrix} R & p \\ 0_{1 \times 3} & 1 \end{bmatrix}_{4 \times 4} \]

translation vector \( p \in \mathbb{R}^3 \)

rotation matrix \( R_{3 \times 3} \)

\( R^T R = I \) - orthogonal
\( \det(R) = 1 \) - right-handed coordinate system

forward kinematics

\[ oT^{TCP}(q) = oT^1(q_0)T^2(q_1)\cdots T^j(q_i)\cdots T^{TCP}(q_{n-1}) \]
Cartesian Impedance Control

What is a Cartesian coordinate $x$ and how does it relate to homogeneous transformations?

$$f = M \Delta \ddot{x} + D_k \Delta \dot{x} + K_k \Delta x$$

$d \mathbf{x} \in \mathbf{x}$

$$\mathbf{x} = \begin{bmatrix} p_x \\ p_y \\ p_z \\ \alpha \\ \beta \\ \gamma \end{bmatrix}$$

translation vector

representation of orientation

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{p}_x \\ \dot{p}_y \\ \dot{p}_z \\ \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix}$$

translational velocity

„velocity“
Representation of Orientations

Roll-Pitch-Yaw

Angle axis representation

\[ \theta = \cos^{-1} \left( \frac{r_{11} + r_{22} + r_{33} - 1}{2} \right) \]
\[ k = \frac{1}{2 \sin \theta} \begin{bmatrix} r_{32} - r_{23} \\ \eta_{3} - \eta_{1} \\ r_{21} - \eta_{2} \end{bmatrix} \]

Singular for \( \theta = n \pi \)!

Quaternions (Euler parameters) not minimal, singularity-free, global

\[ \lambda = [\lambda_0, \lambda_1, \lambda_2, \lambda_3] = \left[ \cos \frac{\theta}{2}, k \sin \frac{\theta}{2} \right] \]
for velocities, there is a singularity-free representation of rotational velocity, leading to the favorite representation of TCP velocity

\[
\dot{x}_\omega = \begin{bmatrix}
\dot{p}_x \\
\dot{p}_y \\
\dot{p}_z \\
\omega_x \\
\omega_y \\
\omega_z
\end{bmatrix}
\]

translation vector \( p \in \mathbb{R}^3 \)

angular velocity vector \( \omega \in \mathbb{R}^3 \)

the Jacobian matrix defines the relation between joint and Cartesian velocity

\[
\dot{x} = J(q)\dot{q}
\]

\[
J(q) = \frac{\partial x(q)}{\partial q}
\]

Be aware that, depending on the representation of \( x \) and the coordinate frame in which this is represented, there exist a multitude of Jacobian matrix versions!!
Basic Jacobian

If \( T_{i,j} = \begin{bmatrix} R_{i,j} & p_{i,j} \\ 0 & 1 \end{bmatrix} \)

for serial manipulators with rotatory joints, the basic Jacobian is:

\[
J_j(q) = \begin{bmatrix}
R_{j,1}z_1 \times p_{1,n+1}, & \cdots & R_{j,n}z_n \times p_{n,n+1} \\
R_{j,1}z_1, & \cdots & R_{j,n}z_n
\end{bmatrix}
\]

\( z_i \) : axis of joint \( i \)

\( z_i, p_{i,n+1} \) expressed in coordinate system of joint \( i \)

for serial manipulators with translational joints, the basic Jacobian is:

\[
J_j(q) = \begin{bmatrix}
R_{j,1}z_1, & \cdots & R_{j,n}z_n \\
0, & \cdots & 0
\end{bmatrix}
\]

(whiteboard sketch)

for derivation, see [Spong, Khalil, Murray]
Joint dynamics

\[ f = \begin{bmatrix} f_x \\ f_y \\ f_z \\ m_x \\ m_y \\ m_z \end{bmatrix} \]

force vector \( f \in \mathbb{R}^3 \)

torque vector \( m \in \mathbb{R}^3 \)

the transposed Jacobian matrix defines the relation between Cartesian and joint torques

\[ \tau = J^T(q) f \]

\[ J(q) = \frac{\partial x(q)}{\partial q} \]

\[ M(q)\ddot{q} + c(q, \dot{q}) + g(q) = \tau \]

- \( M(q) \) - mass matrix
- \( c(q, \dot{q}) \) - vector of centripetal and Coriolis torques
- \( g(q) \) - vector of gravity torques
- Robot Parameters Identification
Problem Statement

\[ \tau = M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) \]

The model depends on following groups of parameters:

- joint variables \((q, \dot{q}, \ddot{q})\) - measured
- kinematics parameters (such as the constant DH-Parameter \(p_{DH}\)) – from CAD
- dynamics parameters \(p\) (masses, centers of mass, momentum of inertia) – often unprecisely known from CAD or variable (variable loads on the tip or along the structure).

\(\tau\) is known, because it is the controller output. For the light-weight robot, it can be directly measured

We will present an off-line procedure for identification of the dynamics parameters

(for unknown kinematics parameters, similar identification methods exist.)
the dynamics model can be rewritten as follows

\[ \tau = Y(q, \dot{q}, \ddot{q}, p_{DH}) p \]

Regressor unknown vector of dynamics parameters

Basic algorithm for identification:
perform several measurements in different configuration:

\[ \tau_1 = Y_1 p \]
\[ \tau_2 = Y_2 p \]
\[ \vdots \]
\[ \tau_l = Y_l p \]

If \( \text{Rank}(Y) = k \), then the system can be solved for \( p \).

\[ p = Y_{tot}^{-1} \tau_{tot} \quad \text{for } k = l \cdot n \]

or

\[ p = Y_{tot}^\# \tau_{tot} \quad \text{for } k \leq l \cdot n \]

(right pseudoinverse: \( Y_{tot}^\# Y_{tot} = I_{k \times k} \))
The dynamics equations of the robot can be obtained by the dynamics formalism:

\[
\frac{d}{dt}\left(\frac{dL(q, \dot{q})}{d\dot{q}}\right) - \frac{dL(q, \dot{q})}{dq} = \tau \quad \text{with} \quad L(q, \dot{q}) = T(q, \dot{q}) - U(q)
\]

Remark: the Lagrangian-Formalism is used here for didactical purposes. For a practical implementation of dynamics computation, the Newton-Euler Algorithm is much more efficient (O(n) vrs. O(n^3)). If M, C, and g are required independently, the Featherstone algorithm is currently the most efficient method for dynamics computation (O(n)).

The kinetic energy of robot segment \( j \) is (without proof):

\[
T = \frac{1}{2} \left[ v^T \omega^T \begin{bmatrix} ml_{3 \times 3} & -m\hat{l}_c \\ m\hat{l}_c & J_I \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \right] \quad m \text{ - mass}
\]

\( l_c \) – center of mass (3 parameters)

\( J_I \) – momentum of inertia (6 parameters))

all quantities w.r.t. reference frame of joint \( j \) (whiteboard)

\[
J_I = I_c - m\hat{l}_c\hat{l}_c
\]

\( I_c \) – inertia w.r.t. center of mass

The kinetic energy is linear in the parameters \( \{m, ml_c, J_I\} \)
One can analogously show that the potential energy is linear in $\{m, ml_c\}$.

One therefore has

$$p_0^T = \begin{bmatrix} m_1, ml_{c1}, J_1, \ldots, m_n, ml_{cn}, J_n \end{bmatrix} \quad 10n \text{ parameters!}$$

however, in general not all parameters can be independently identified:

- if the column of $Y$ corresponding to one parameter is zero, then the parameter can be removed.
- if two columns in $Y$ are linearly dependent, then only a linear combination of the two parameters can be identified.

By eliminating all linear dependencies, one can reach a minimal parameter set, for which $Y$ can have full rank (if the measurements are independent). This minimal parameter set can be completely identified.

In [Khalil02] one can find detailed rules for the formulation of the minimal parameter set.
a) Write the regressor for the identification of the dynamics parameters for this system.

b) Which conditions must be fulfilled in this case in order to identify all parameters? What is the minimal number of robot configurations in which measurements have to be performed for the identification? Why should one choose in practice substantially more points?
find the minimal parameter set and the regressor for the following system:

\[
\begin{align*}
\tau_1 &= M_{11} \ddot{q}_1 + M_{12} \ddot{q}_2 + C_{11} \dot{q}_1 + C_{12} \dot{q}_2 + g_1 \\
\tau_2 &= M_{21} \ddot{q}_1 + M_{22} \ddot{q}_2 + C_{21} \dot{q}_1 + C_{22} \dot{q}_2 + g_2
\end{align*}
\]

\[
M_{11} = m_2 (l_1^2 + 2l_1 l_c \cos(q_2)) + I_1 + I_2 \\
M_{12} = M_{21} = 2m_2 l_c l_1 \cos(q_2) + I_2 \\
M_{22} = I_2 \\
h = -m_2 l_1 l_c \sin(q_2)
\]

\[
C_{11} = h \dot{q}_2 \\
C_{12} = h (\dot{q}_1 + \dot{q}_2) \\
C_{21} = -h \dot{q}_1 \\
C_{22} = 0
\]

\[
g_1 = (m_1 l_c + m_2 l_1) g \cos(q_1) + m_2 l_c g \cos(q_1 + q_2) \\
g_2 = m_2 l_c g \cos(q_1 + q_2)
\]

Hint: The columns of the regressor will be linearly dependent – reduce the dimension of the parameter vector
Practical Aspects

• In practice, one will choose $l >> k$.
• The identification can be also formulated as a minimization problem:
  \[
  \min_{p} \{ \tau - Yp \}
  \]
  If $Y$ has full rank, then the pseudoinverse provides the solution of the optimization problem.
• It is very important to choose measurement points (trajectories), which excite all parameters well - each parameter should have a strong influence on the measurements. A measure therefore is the conditioning index of $Y$

  \[
  \text{cond}(Y) = \frac{\lambda_{\text{max}}}{\lambda_{\text{min}}} \geq 1
  \]
  \[\lambda\]-singular values
  which should be as close as possible to 1.
• It is useful to group the parameters in small sets (wrist joints/ shoulder joints, static parameters / constant velocities, …) and to design special purpose trajectories for excitation of each set separately.
• signals must mostly be filtered for good results.