Task Space Control
Decoupling in the Task Space
Robotic Assembly of Complex Planar Parts: an Experimental Evaluation

P. Robuffo Giordano, A. Stemmer, K. Arbter and A. Albu-Schäffer

German Aerospace Center (DLR)
Robot Equations in Task Coordinates

Robot Model in joint coordinates:

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

- \( q \) - coordinates of \( Q \) (\( \mathbb{R}^6 \) or 6-Torus)
- \( \dot{q} \) - vector
- \( c(q, \dot{q}), g(q), \tau \) - covector
- \( M(q) \) - covariant tensor of second order (2,0)

Forward kinematics: local mapping between manifolds (\( Q \) and \( SE3 \))

\[ f : Q \rightarrow SE3, \quad x = f(q) \]

\( f \) is a local 1:1 mapping between manifolds
- for non-redundant manipulators (\( \dim(Q)=6 \)) and
- in non-singular configurations

Then we know how all quantities transform!
Robot Equations in Task Coordinates

Robot model in $SE3$ coordinates:

$$M_K(q)\ddot{x} + c_K(q, \dot{q}) + g_K(q) = F$$

$$x = f(q); \quad q = f^{-1}(x)$$

$$\dot{x} = J(q)\dot{q}; \quad \dot{q} = J^{-1}(q)\dot{x}$$

And the acceleration?

$$\ddot{x} = J(q)\ddot{q} + \dot{J}(q)\dot{q}$$

$$F = J^{-T}(q)\tau; \quad \tau = J^T(q)F$$

$$c_K(q, \dot{q}) = J^{-T}(q)c(q, \dot{q}) - M_K(q)\dot{J}(q)\dot{q}$$

$$M_K(x) = J^{-T}(q)M(q)J^{-1}(q); \quad M(q) = J^T(q)M_K(x)J(q)$$

Alternative formulation:

$$M_x(x)\ddot{x} + c_x(x, \dot{x}) + g_x(x) = F$$

Restricted to configurations in which $x = f(q)$ is a 1:1 mapping.

Extra info: (a vector on a $2n$ manifold with coordinates $(q, \dot{q})$: Tangent bundle)

(the same with $g_K(q)$ )
Robot model in $\textit{SE3}$ coordinates $x$:

$$M_K(q) \ddot{x} + c_K(q, \dot{q}) + g_K(q) = F$$

Is here $F$ a Wrench $F = (f, m)^T$?

$x = f(q); \quad q = f^{-1}(x)$

$$x = \begin{bmatrix} p_x \\ p_y \\ p_z \\ \alpha \\ \beta \\ \gamma \end{bmatrix}, \quad \dot{x} = \begin{bmatrix} \dot{p}_x \\ \dot{p}_y \\ \dot{p}_z \\ \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix}$$

F is a generalized force dual to such $\dot{x}$ that $F^T \dot{x} = P$

Jacobian matrices general:

$$\dot{x} = \frac{df(q)}{dt} = \frac{\partial f(q)}{\partial q} \dot{q} = J(q) \dot{q}$$

The dual vector to $m$ is $\omega$.

$$\dot{x}_j = J_j(q) \dot{q}, \quad \tau = J_j^T(q) F_j \quad \text{mit} \quad \dot{x}_j = \begin{bmatrix} v_j \\ \omega_j \end{bmatrix}, \quad F_j = \begin{bmatrix} f_j \\ m_j \end{bmatrix}$$

(basic Jacobian, geometric Jacobian)

expressed in the coordinate system $j$

for examples expressed in TCP- coordinates - body Jacobian

for the derivation of $J_j$ see [Spong, Khalil, Murray]
The angular velocity is not integrable to local coordinates for SO3, however there is following relation to the derivative of $R$:

$$\hat{\omega}_{i,j} = R_{i,j}^T \dot{R}_{i,j}$$

Matrix representation of a vector

$$\hat{\omega} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$$

such that $\omega \times r = \hat{\omega} r$

Vector product as matrix operation

Since the local coordinates $x_R$ for SO3 are expressed in terms of R as well

$$x_R = f_R(R_v) \Rightarrow \dot{x}_R = J_f R_v \dot{R}_v$$

since $R_v$ contains the elements of $R$ stacked as a vector

one can derive the transformation between the two vectors:

$$\dot{x}_R = J_R \omega$$

The Jacobian can therefore be expressed as

$$\dot{x} = \begin{bmatrix} \nu \\ \dot{x}_R \end{bmatrix} = \begin{bmatrix} I_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & J_R \end{bmatrix} \begin{bmatrix} \nu \\ \omega \end{bmatrix} = J_{TR} J_b \dot{q} = J \dot{q}$$
Roll-Pitch-Yaw-Angle:

\[
\begin{align*}
\gamma &= \text{atan2}(r_{32}, r_{33}) \\
\beta &= \text{atan2}\left(-r_{31}, \sqrt{r_{11}^2 + r_{21}^2}\right) \\
\alpha &= \text{atan2}(r_{21}, r_{11})
\end{align*}
\]

(Singularity at \( \beta = \pm 90^\circ \))

Quaternions (Euler Parameter):

\[
\lambda = [\lambda_0, \lambda_1, \lambda_2, \lambda_3] = \left[ \cos \frac{\theta}{2}, \vec{k} \sin \frac{\theta}{2} \right]
\]

\[
\lambda_0 = \frac{1}{2} \sqrt{r_{11} + r_{22} + r_{33} + 1}
\]

\[
\lambda_1 = \frac{1}{2} \text{sgn}(r_{32} - r_{23}) \sqrt{r_{11} - r_{22} - r_{33} + 1}
\]

\[
\lambda_2 = \frac{1}{2} \text{sgn}(r_{13} - r_{31}) \sqrt{-r_{11} + r_{22} - r_{33} + 1}
\]

\[
\lambda_3 = \frac{1}{2} \text{sgn}(r_{21} - r_{12}) \sqrt{-r_{11} - r_{22} + r_{33} + 1}
\]

\[
\dot{\beta} = J_R \omega = \begin{bmatrix}
\cos \alpha & \sin \alpha & 0 \\
\cos \beta & \cos \beta & 0 \\
-\sin \alpha & \cos \alpha & 0
\end{bmatrix} \omega
\]

\[
\begin{bmatrix}
\dot{\alpha} \\
\dot{\beta} \\
\dot{\gamma}
\end{bmatrix} = J_R \omega
\]

\[
J_R = \frac{1}{2} \begin{bmatrix}
-\lambda_1 & -\lambda_2 & -\lambda_3 \\
\lambda_0 & \lambda_3 & -\lambda_2 \\
-\lambda_3 & \lambda_0 & \lambda_1 \\
\lambda_2 & -\lambda_1 & \lambda_0
\end{bmatrix}
\]
Decoupled Position Control in Task Coordinates

(Feedback Linearization or I/O Linearization or Computed Torque Controller)

Robot model in $SE3$ coordinates $x$:

$$M_K(q)\ddot{x} + c_K(q, \dot{q}) + g_K(q) = F$$

Controller

$$F = M_K(q)[\ddot{x}_d + K(x_d - x) + D(\dot{x}_d - \dot{x})] + c_K(q, \dot{q}) + g_K(q)$$

$$\tau = J^T(q)F$$

Commanding torques requires:
- Direct drives (no gearbox)
- Torque control

It follows a linear, decoupled error dynamics:

$$\ddot{e} + D\dot{e} + Ke = 0$$

$$e = x_d - x$$
Redundant Robots

Task Space: $m$ dimensional  \( m \leq 6 \)
Joint Space $Q$: $n$ dimensional

Redundant manipulator: $n > m$

$n-m$: degree of redundancy

$n=7, \ m=6$

$n=10, \ m=6$
Nullspace Motion for Justin
For non-redundant robots we have:

\[
M_K(q)\ddot{x} + c_K(q, \dot{q}) + g_K(q) = F
\]

\[
\ddot{x} = J(q)\dot{q}; \quad \dot{q} = J^{-1}(q)\ddot{x}
\]

\[
F = J^{-T}(q)\tau; \quad \tau = J^T(q)F
\]

\[
M_K(x) = J^{-T}(q)M(q)J^{-1}(q);
\]

For redundant robots, \(J_{m\times n}\) is not quadratic, thus not invertible!

Solution: pseudo-inverse Matrix:

\[
J_{m\times n}J^\#_{n\times m} = I_{m\times m}
\]

note that \(J^\#_{n\times m}J_{m\times n} \neq I_{n\times n}\)

with

\[
J^\# = A J^T (JAJ^T)^{-1}
\]

\(A\)-p.d. matrix

\(A=I\) special case:

Moore-Penrose

Pseudo-inverse

\[
J^\# = J^T_{n\times m}(JJ^T)^{-1}
\]
then we have: \[ \dot{q} = J^\#(q)\dot{x} \quad F = J^#T(q)\tau \]
\[ M_R(x) = J^#T(q)M(q)J^\#(q) \]

and again: 
\[ M_R(q)\ddot{x} + c_R(q,\dot{q}) + g_R(q) = F \]

But: How to chose the matrix $A$ correctly?
(different $A$ lead to different results for $M_R$!?!)

Everything you can do with a pseudo inverse, you can do also without !!!

Solution: to compute $M_R$ compute first the inverse $M_{R}^{-1}$

$M_{R}^{-1}$ is a tensor of type $(0,2)$ – twice contravariant

It therefore transforms as:
\[ M_{R}^{-1} = JM^{-1}J^T \]

because $p^T M_R^{-1} p = E_{kin}$
\[ p = M_R \dot{x} \quad \text{impulse, covariant} \]

It follows: 
\[ M_{R} = (JM^{-1}J^T)^{-1} \]
Redundant Robots: Dynamics

\[ J^\# = A J^T (JAJ^T)^{-1} \]

\[ M_R = (JM^{-1}J^T)^{-1} \]

One can directly verify that for \( A = M^{-1}(q) \)

\[ M_R(x) = J^\#^T(q)M(q)J^\#(q) \]

Therefore, one should choose the pseudoinverse:

\[ J^\# = M^{-1} J^T (JM^{-1}J^T)^{-1} \]

Exercise: check that with this pseudo inverse, the two expressions for \( M_R \) are equivalent
Useful Formulas

\[(AB)^T = B^T A^T \quad A, B – Matrices\]
\[(AB)^{-1} = B^{-1} A^{-1} \quad A, B – Invertible Matrices\]

Properties of mass and stiffness matrices:
\[A \in \{M, K\}\]

\[\text{A is positive definite (p.d.):}\]
\[v^T Av > 0, \quad \forall v, \|v\| \neq 0\]
\[\angle(v, Av) < 90^\circ \quad \forall v\]

All eigenvalues of \(A\) are positive

\[A\] is symmetrical
\[A^T = A\]

Since \(A\) is symmetric and p.d., the same is valid for
\[J^T AJ \quad \text{if } J – \text{non-singular}\]

Orthogonal decomposition:
\[A = U\Lambda U^T \quad \Lambda - \text{Diagonal} \quad \lambda_i - \text{Eigenvalues of } A\]
\[U - \text{Orthogonal matrix}\]
Nullspace Motion

Nullspace projection:

\[ \dot{q}_N = (I - J^\#(q)J(q))\dot{q}_0 \]

Verification:

\[ J(q)\dot{q}_N = 0 \]

Invers kinematics:

\[ \dot{q} = J^\#(q)\dot{x} + (I - J^\#(q)J(q))\dot{q}_0 \]

\[ \delta q = J^\#(q)\delta x + (I - J^\#(q)J(q))\delta q_0 \]
Interpretation of the Pseudoinverse

(What do the different scalings of the pseudoinverse mean?)

Inverse kinematics: \[ \dot{q} = J^\#(q)\dot{x} \]

\[
J^\# = A \quad J^T (JAJ^T)^{-1}
\]

For \( A=I \), leading to
\[ J^\# = J^T (JJ^T)^{-1} \]
the inverse kinematics will minimize \( \|\dot{q}\| \) under the constraint
\[ \dot{x} - J(q)\dot{q} = 0 \]

Generally, for a weighting matrix \( A \), the norm \( \|\dot{q}\|_{A^{-1}} \) will be minimized

For example, for \( A = \frac{1}{2} M^{-1}(q) \) one minimizes the kinetic energy:
\[
\|\dot{q}\|^2_M = \frac{1}{2} \dot{q}^T M(q)\dot{q}
\]
Nullspace Torque

Nullspace projection:

\[ \tau_N = (I - J^T(q)J\#^T(q))\tau_0 \]

Verification

\[ \tau_N = 0 \text{ if } \tau_0 = J^T\tau_N \]

Commanded joint torque:

\[ \tau = J^T(q)F + (I - J^T(q)J\#^T(q))\tau_0 \]

nullspace component (secondary task)

Cartesian force (primary task)
When transforming the Cartesian mass matrix back to joint coordinates in the redundant case,

\[ M_J(q) = J^T(q)M_R(q)J(q) \]

one does not obtain the complete joint space matrix, but only the part which has an effect on the TCP. The null-space component is filtered out, similarly to the transformation of joint torques to TCP coordinates and back.

\[ M_J(q) \neq M(q) \]
The computed torque approach leads to a linear, decoupled dynamics in task space:

\[ \ddot{e} + D\dot{e} + Ke = 0 \]

\[ e = x_d - x \]

The nullspace component should fulfill at least the following requirements:

- provide a damping term
  \[ \tau_0 = -D\dot{q} \]

- provide a term for avoiding end stops
  \[ \tau_0 = K_N (q_0 - q) \]

- possibly contain a term for avoiding singularities and possibly other optimization criteria
Handling Singularities

\[ \dot{x}_m = J_{m \times n}(q) \dot{q}_n \]

The robot is singular, if

\[ \text{Rank} \{ J_{m \times n}(q) \} = m - r < m \]

for non-redundant robots, this means:

\[ d(q) = \det(J_{m \times m}(q)) = 0 \]

for redundant robots we can restate:

\[ d(q) = \det(J(q)J^T(q)) = 0 \]

\[ m \times m \text{ Matrix} \]

\[ d(q) = d_1(q) \cdot d_2(q) \cdot \ldots \cdot d_i(q) \quad i \text{ different types of singularities} \]

In the vicinity of singularities, one can control \((m-r)\) coordinates and regard the manipulator as a redundant system.

The control in the singular direction (locale coordinate \(d_i\)) is performed in the null-space of the primary task.