Open Questions

- How do we choose muscle activations so as to obtain a given joint torque?
- Which is the minimum number of elementary force fields that we need to perform a ‘complete’ set of movements?
- Is there a way of choosing the primitives to accommodate different kinematic structures?
- How can we predict the trajectory followed by the system when it is driven by a given force field $F$? (Dynamic model of the limb)
- Is there a way of choosing the ‘complete’ set of elementary force fields $F_k$? (A trivial solution to the spinal field synthesis problem)
- How should we choose joint torques so as to obtain a given basis force field $F_k$? (The map $-\mapsto F_k$)

A trivial solution to the synthesis problem (1/2)

**HINT:**

\[ \tau_k = -K_k \tau_k - K_{q,k} q - q_{d,k} \]

And impose the following for all admissible $q_0$:

\[ \sum_{k=1}^{n} \lambda_k \tau_k(q_k) = -K_k \dot{q} - K_{q,k} (q - q_d) \]

Which can be rewritten:

\[ \sum_{k=1}^{n} \lambda_k [K_k \dot{q} + K_{q,k} (q - q_{d,k})] = K_k \dot{q} + K_{q,k} (q - q_{d,k}) \]

Which is verified if:

\[ \sum_{k=1}^{n} \lambda_k = 1 \]

and \[ \sum_{k=1}^{n} \lambda_k q_{d,k} = q_d \]

A trivial solution to the synthesis problem (2/2)

Rewriting the previous expression:

\[ A \lambda = \begin{bmatrix} q_{d,1} & \cdots & q_{d,n} \end{bmatrix} \]

Which has a solution for any $q_{d}$ if and only if the matrix on the left has full row rank. This observation gives a criteria to choose the equilibrium points realized by the elementary controls.

Moreover, we can have full row rank only if:

\[ K \geq n + 1 \]

This can be proven to be the minimum number of primitives necessary to control an $n$-DOF kinematic chain!

Back to the end-effector space

In the redundant case we can go back and forth:

\[ \tau(q, \dot{q}) = \sum_{k=1}^{n} \lambda_k \tau_k(q_k, \dot{q}_k) \quad \text{and} \quad \tilde{F}(\bar{z}, \tilde{z}) = \sum_{k=1}^{n} \lambda_k \tilde{F}_k(\bar{z}, \tilde{z}) \]

Using the following equations:

\[ \bar{z} = \Lambda(q) \quad \text{and} \quad \dot{q} = \Lambda_{\omega}(\bar{z}) \]

\[ \tilde{z} = J(q) \dot{q} \quad \text{and} \quad \dot{q} = J^{-1}(\tilde{z}) \tilde{q} \]

\[ \tau(q, \dot{q}) = J^T(q) \tilde{F}(\tilde{z}, \tilde{z}) \\quad \text{and} \quad \tilde{F}(\bar{z}, \tilde{z}) = J^{-1}(\Lambda_{\omega}(\bar{z})) \tilde{F}(\bar{z}, \tilde{z}) \]

Graphical representation of the fields (2DOF chain)

Can be graphically represented (null velocities): \[ \tau_k(q, \dot{q}) \]

Looks quite different in the Cartesian space: \[ \tilde{F}(\tilde{z}, \tilde{z}) \]

**EXAMPLE**

Identity \[ \tau_k(q, \dot{q}) = (K_k \dot{q} - q_{d,k}) \]

Equilibrium Point

\[ x_F = \begin{bmatrix} x_e \\ y_e \end{bmatrix} \]

Ending
Trajectory tracking

We have seen how to choose the control action so as to drive the system to a predefined (global) equilibrium. Sometimes we are interested in tracking a given trajectory.

Specifically, we are interested in finding a control action such that a given trajectory is asymptotically tracked i.e. the tracking error:

\[ e = q - q_d \]

asymptotically tends to zero.

### COMPUTED TORQUE control of a chain (1/2)

Suppose that you know the matrices \( M, C, G \) of the dynamical model:

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

Control Action

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

System Dynamics

\[ M(q) \ddot{q} = M(q) \ddot{q} + \dddot{\tau} \]

\[ \dddot{\tau} = -K \dddot{q} \]

Control method: \( \dddot{\tau} = -K \dddot{q} \)

We now prove that: \( \lim_{t \to \infty} e = 0 \)

### COMPUTED TORQUE control of a chain (2/2)

- FACT: If the matrices \( K_1 \) and \( K_2 \) are symmetric and positive definite then the 'computed torque' control law results in exponential trajectory tracking i.e. \( e \to 0 \)
- PROOF: First observe that the dynamics of \( e \) are linear and can be rewritten as follows:

\[
\begin{bmatrix}
\dot{e} \\
\ddot{e}
\end{bmatrix} =
\begin{bmatrix}
0 & 1 \\
-K & K
\end{bmatrix}
\begin{bmatrix}
e \\
\dot{e}
\end{bmatrix}
\]

We want to prove that all the eigenvalues of \( A \) have negative real part. Let \( \lambda_1, \lambda_2 \) be the corresponding eigenvectors. Then we have:

\[
\begin{bmatrix}
\lambda_1 \\
\lambda_2
\end{bmatrix}
= \begin{bmatrix} v_1 & v_2 \end{bmatrix}
\]

Without loss of generality choose \( v_1 \) with unitary norm so that we have:

\[
\begin{bmatrix} v_1^T \\
\dot{v}_1^T \end{bmatrix} A \begin{bmatrix} v_1^T \\
\dot{v}_1^T \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 \\
0 & \lambda_2 \end{bmatrix}
\]

where \( c>0 \) and \( \beta > 0 \) because \( K_1 \) and \( K_2 \) are positive definite. We therefore have:

\[
\dot{v}_1 = H v_1 \]

where \( H \) is symmetric if and only if:

\[
\dot{v}_1 \dot{v}_1^T \text{ symmetric}
\]

and

\[
H = \begin{bmatrix} 0 & I \\
-K & K \end{bmatrix}
\]

Then:

\[
0 = 1 + \lambda_1 \lambda_2, \quad 0 \leq \lambda_1, \lambda_2 < \infty
\]

\[ e \to 0 \text{ as } t \to \infty \]

### Control action in the task space (1/2)

Consider the following 'non-redundant' manipulator:

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

And let the Jacobian be:

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

Commutations lead to:

\[
M(q) \ddot{q} + C(q, \dot{q}) \dot{q} + G(q) = F
\]

where:

\[
\dot{M}(q) = J^T(q) M(q) J(q) + M(q) \dot{J}(q)
\]

\[
\dot{C}(q, \dot{q}) = J^T(q) \left[ C(q, \dot{q}) \dot{J}(q) + M(q) \frac{dJ(q)}{dt} J^T(q) \right]
\]

\[
\dot{G}(q) = J^T(q) \dot{G}(q)
\]

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

### Control action in the task space (2/2)

- LEMMA: The matrices of the dynamic model of a kinematic chain satisfy the same properties that hold for \( M \) and \( C \):

\[
M(q) = \tilde{M}(q) > 0
\]

Follows from the fact that the kinetic energy is positive and equals zero only at rest

\[
\tilde{M}(q) = 2 \tilde{C}(q, \dot{q})
\]

is skew symmetric

A given matrix \( A \) is skew symmetric if and only if:

\[
A = A^T
\]
Control in the task space

PD Control in the task space:

Stabilizing controller

\[ F = \ddot{G}(q) = K_{\dot{q}} \dot{x} = K_x (x - x_d) \]

Tracking Control in the task space:

Tracking error \[ e = x - x_d \]

Tracking controller

\[ F = \ddot{M}(q)(\ddot{x} - K_{\dot{x}} \dot{x} - K_x e) + \dddot{C}(q, \dot{q}) \dot{q} + \dddot{G}(q) \]