

Ch10: Force Control

Pure Force Control

Pure Force Control

(Direct Force Control)

- When the task is not to create motions at the end-effector but to apply forces and torques to the environment, **force control** is needed.
- Pure force control is only possible if the environment provides resistance forces in every direction.

$$\tau = M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) + J^T(q)\mathcal{F}_{\text{tip}}$$

During a force control task: $\dot{q} = \ddot{q} \approx 0$ ↓

$$\tau = g(q) + J^T(q)\mathcal{F}_{\text{tip}}$$

Wrench applied by manipulator to environment

(1) If only joint position feedback is available, the control law is: $\tau = g(q) + J^T(q)\mathcal{F}_d$

\mathcal{F}_d : desired wrench

- This requires a good model for gravity compensation and precise control of the torques at joints.

Pure Force Control

(2) If a six-axis force-torque sensor between the arm and the end-effector is available to directly measure \mathcal{F}_{tip} , a PI force controller is

$$\tau = g(q) + J^T(q)(\mathcal{F}_d + K_p \mathcal{F}_e + K_i \int \mathcal{F}_e(t) dt)$$

where $\mathcal{F}_e = \mathcal{F}_d - \mathcal{F}_{\text{tip}}$ and K_p and K_i are positive-definite matrices.

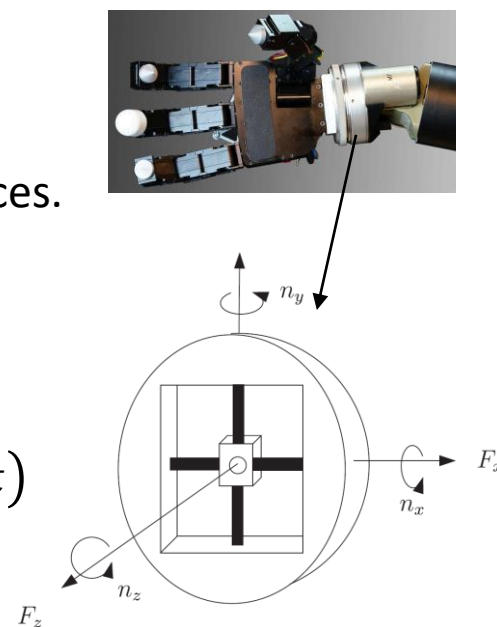
Closed-loop dynamics:

$$g(q) + J^T(q)\mathcal{F}_{\text{tip}} = g(q) + J^T(q)(\mathcal{F}_d + K_p \mathcal{F}_e + K_i \int \mathcal{F}_e(t) dt)$$

$$(K_p + I)\mathcal{F}_e + K_i \int \mathcal{F}_e(t) dt = 0$$

$$(K_p + I)\dot{\mathcal{F}}_e + K_i \mathcal{F}_e = 0$$

$\Rightarrow \mathcal{F}_e$ converges to zero for positive-definite matrices.



Pure Force Control

If there is nothing for the robot to push against, it will accelerate in a failing attempt to create end-effector forces. Since a typical force-control task requires little motion, we can limit this acceleration by adding velocity damping as

$$\boldsymbol{\tau} = \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{J}^T(\boldsymbol{q})(\boldsymbol{\mathcal{F}}_d + \boldsymbol{K}_p\boldsymbol{\mathcal{F}}_e + \boldsymbol{K}_i\int \boldsymbol{\mathcal{F}}_e(t)dt - \boldsymbol{K}_{\text{damp}}\boldsymbol{v})$$

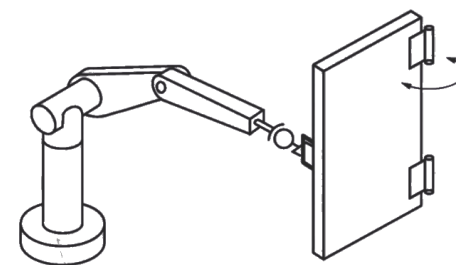
where $\boldsymbol{K}_{\text{damp}}$ is positive definite.

Hybrid Motion-Force Control

Natural and Artificial Constraints

Assume that the task space is r -dimensional, and the environment is infinitely stiff (rigid constraints) in k directions and unconstrained in $r - k$ directions. Thus, there are k directions in which the robot can freely apply forces and the $r - k$ directions of free motion.

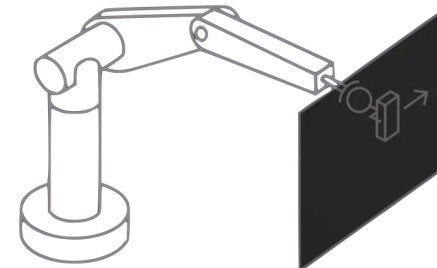
Example: A robot firmly grasping a door handle has $6 - k = 1$ motion freedom of its end-effector (rotation about the hinge), and $k = 5$ force freedoms.



Example: A robot writing on a chalkboard has $6 - k = 5$ motion freedom of its end-effector, and $k = 1$ force freedoms.

- If there is friction at the contact, $k = 3$.
- If the robot move away from the board, $k = 0$. Thun, velocity constraint is inequality.

Example: A robot erasing a frictionless chalkboard using an eraser modeled as a rigid block has $6 - k = 3$ motion freedom of its end-effector, and $k = 3$ force freedoms.



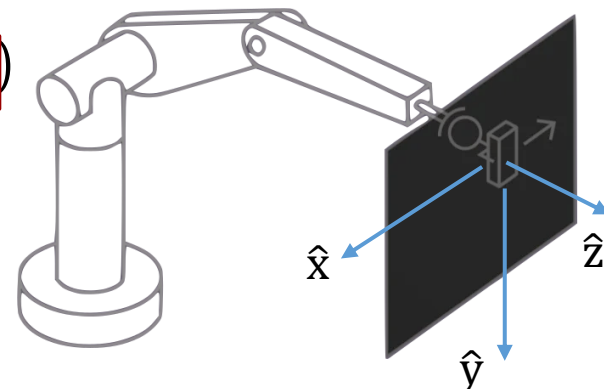
Natural and Artificial Constraints

$$\mathbf{v}_b = (\omega_x, \omega_y, \omega_z, v_x, v_y, v_z) \quad \mathbf{F}_b = (m_x, m_y, m_z, f_x, f_y, f_z)$$

The parameters that the robot can control. These motion and force specifications are called **artificial constraints**.

These constraints are called **natural constraints**, specified by the environment:

$$\begin{aligned} \omega_x &= 0 & f_x &= 0 \\ \omega_y &= 0 & f_y &= 0 \\ v_z &= 0 & m_z &= 0 \end{aligned}$$



natural constraint	artificial constraint
$\omega_x = 0$	$m_x = 0$
$\omega_y = 0$	$m_y = 0$
$m_z = 0$	$\omega_z = 0$
$f_x = 0$	$v_x = k_1$
$f_y = 0$	$v_y = 0$
$v_z = 0$	$f_z = k_2 < 0$

(An example set of artificial constraints)

Hybrid Motion-Force Control

Assume that an n -DOF open-chain manipulator is in contact with a rigid environment with k **natural constraints** on the velocity in 6-dimensional task space. Thus, forces can be applied in k constraint directions. These Pfaffian (holonomic and/or nonholonomic) constraints can be written w.r.t twist as

$$A(q)\mathcal{V} = \mathbf{0} \quad (1) \quad A(q) \in \mathbb{R}^{k \times 6}, \quad \mathcal{V} \in \mathbb{R}^6 \text{ twist}$$

↓

$$A(q)\dot{\mathcal{V}} + \dot{A}(q)\mathcal{V} = \mathbf{0} \quad (2)$$

Task-space Dynamics: $\mathcal{F} = M_c(q)\dot{\mathcal{V}} + c_c(q, \mathcal{V}) + g_c(q) = M_c(q)\dot{\mathcal{V}} + h_c(q, \mathcal{V}) \quad (3)$

$$\tau = J^T(q)\mathcal{F}$$

Constrained Dynamics: $\mathcal{F} = M_c(q)\dot{\mathcal{V}} + h_c(q, \mathcal{V}) + \underbrace{A^T(q)\lambda}_{\mathcal{F}_{\text{tip}}} \quad (4)$

$\lambda \in \mathbb{R}^k$: Lagrange Multipliers
 $\mathcal{F}_{\text{tip}} \in \mathbb{R}^6$: Wrench that robot applies against the constraints

Note: Desired wrench \mathcal{F}_d must lie in the column space of $A^T(q)\lambda$.

Note: All velocity constraints are equality constraints, and the contact is frictionless.

Hybrid Motion-Force Control

Solving (4) for $\dot{\mathcal{V}}$, substituting into (2), solving for λ , and using $-A(q)\dot{\mathcal{V}} = \dot{A}(q)\mathcal{V}$:

$$\lambda = (AM_C^{-1}A^T)^{-1}(AM_C^{-1}(\mathcal{F} - h_C) - A\dot{\mathcal{V}}) \quad (5)$$

Substituting (5) into (4) and manipulating, the 6 equations of the constrained dynamics (4) can be expressed as the $6 - k$ independent motion equations:

$$P(q)\mathcal{F} = P(q)(M_C\dot{\mathcal{V}} + h_C)$$

$$P(q) = I - A^T(AM_C^{-1}A^T)^{-1}AM_C^{-1} \in \mathbb{R}^n \quad \text{rank}(P) = 6 - k$$

$$I = \text{diag}(1) \in \mathbb{R}^n$$

- P projects an arbitrary wrench \mathcal{F} onto the subspace of wrenches that move the end-effector tangent to the constraints ($\text{rank}(P) = 6 - k$).
- $I - P$ projects an arbitrary wrench \mathcal{F} onto the subspace of wrenches that act against the constraints ($\text{rank}(I - P) = k$).

$$\mathcal{F} = \underbrace{P(q)\mathcal{F}}_{\mathcal{F}_{\text{motion}}} + \underbrace{(I - P(q))\mathcal{F}}_{\mathcal{F}_{\text{tip}}}$$

Hybrid Motion-Force Control

Hybrid motion-force controller is the sum of a task-space motion controller (e.g., a computed torque control law), and a task-space force controller, each projected to generate forces in its appropriate subspace as

$$\boldsymbol{\tau} = \mathbf{J}^T(\mathbf{q}) \left(\underbrace{(\mathbf{P})(\mathbf{M}_c(\mathbf{q})\mathbf{y})}_{\text{motion control}} + \underbrace{(\mathbf{I} - \mathbf{P})(\mathcal{F}_d + \mathbf{K}_p\mathcal{F}_e + \mathbf{K}_i \int \mathcal{F}_e(t)dt)}_{\text{force control}} + \mathbf{h}_c(\mathbf{q}, \mathbf{v}) \right)$$

$$\mathbf{y} = \frac{d}{dt} \left([\text{Ad}_{\mathbf{T}^{-1}\mathbf{T}_d}] \mathbf{v}_d \right) + \mathbf{K}_p \mathbf{T}_e + \mathbf{K}_d \mathbf{v}_e$$

Feedforward acceleration expressed in the actual end-effector frame at \mathbf{T} .

$$[\mathbf{T}_e] = \log(\mathbf{T}^{-1}\mathbf{T}_d)$$

Configuration Error

Velocity Error

$$\mathbf{v}_e = [\text{Ad}_{\mathbf{T}^{-1}\mathbf{T}_d}] \mathbf{v}_d - \mathbf{v}$$

Impedance Control

Impedance Control

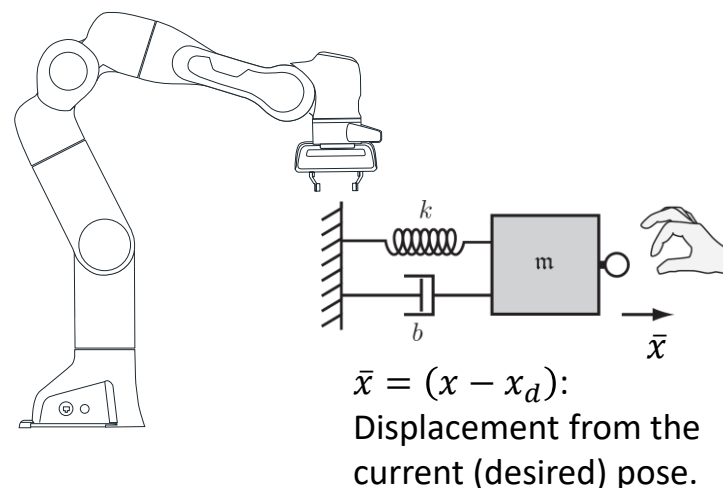
(Indirect Force Control)

In **impedance control**, the robot end-effector is asked to render particular mass, spring, and damper properties (example: haptic surgical simulator).

Assume that a robot creating a 1-DOF mass-spring-damper virtual environment at the end-effector and a user applies a force f .

$$m\ddot{\bar{x}} + b\dot{\bar{x}} + k\bar{x} = f$$

We can say that the robot **impedance is high** if one or more of the $\{m, b, k\}$ parameters, usually including b or k , is **large**, and the **impedance is low** if all these parameters are **small**.



Taking the Laplace transform: $(ms^2 + bs + k)X(s) = F(s)$

Impedance: $Z(s) = F(s)/X(s)$

Admittance: $Y(s) = Z^{-1}(s) = X(s)/F(s)$

Impedance and Admittance

- An ideal **motion controller** is characterized by high impedance or low admittance (since $\Delta X = Y\Delta F$, if Y is small, force disturbances ΔF produce only small change in motion ΔX).
- An ideal **force controller** is characterized by low impedance or high admittance (since $\Delta F = Z\Delta X$, if Z is small, motion disturbances ΔX produce only small change in force).

Goal of impedance control is to implement the mass-spring-damper behavior in task-space:

Minimum-Coordinate Representation :

$$\mathbf{M}\ddot{\bar{\mathbf{x}}} + \mathbf{B}\dot{\bar{\mathbf{x}}} + \mathbf{K}\bar{\mathbf{x}} = \mathbf{f}_{\text{ext}}$$
$$\bar{\mathbf{x}}, \mathbf{f}_{\text{ext}} \in \mathbb{R}^m$$
$$\mathbf{M}, \mathbf{B}, \mathbf{K} \in \mathbb{R}^{m \times m}, \text{PD}$$
$$\bar{\mathbf{x}} = (\mathbf{x} - \mathbf{x}_d)$$
$$\ddot{\mathbf{x}}_d = \dot{\mathbf{x}}_d = \mathbf{0}$$

Note: There are two common ways to achieve the impedance behavior; (1) using an impedance controller, (2) using an admittance controller.

Impedance Controller

An impedance controller measures end-effector motions $\mathbf{x}(t)$, $\dot{\mathbf{x}}(t)$, $\ddot{\mathbf{x}}(t)$ using encoders, tachometers, and possibly accelerometers, and commands joint torques/forces to create end-effector forces ($-\mathbf{f}_{\text{ext}}$) to mimic a mass-spring-damper system. Thus, the controller implements a transfer function $Z(s)$ from motions to forces.

A control law is
$$\boldsymbol{\tau} = J^T(\mathbf{q}) \left(\underbrace{\mathbf{M}_C(\mathbf{q})\ddot{\mathbf{x}} + \mathbf{h}_C(\mathbf{q}, \dot{\mathbf{x}})}_{\text{robot dynamics compensation}} - \underbrace{(\mathbf{M}\ddot{\mathbf{x}} + \mathbf{B}\dot{\mathbf{x}} + \mathbf{K}\bar{\mathbf{x}})}_{\mathbf{f}_{\text{ext}}} \right) \quad \bar{\mathbf{x}} = (\mathbf{x} - \mathbf{x}_d)$$

Note: Since measurement of the acceleration $\ddot{\mathbf{x}}$ is likely to be noisy, it is not uncommon to eliminate the mass compensation term $\mathbf{M}_C(\mathbf{q})\ddot{\mathbf{x}}$ and to set $\mathbf{M} = \mathbf{0}$. The mass of the arm will be apparent to the user, but impedance-controlled manipulators are often designed to be lightweight.

Note: It is not uncommon to assume small velocities and replace the nonlinear dynamics compensation with a simpler gravity-compensation model, i.e., $\mathbf{h}_C(\mathbf{q}, \dot{\mathbf{x}}) = \mathbf{g}_C(\mathbf{q})$.

$$\boldsymbol{\tau} = J^T(\mathbf{q}) \left(\mathbf{g}_C(\mathbf{q}) - (\mathbf{B}\dot{\mathbf{x}} + \mathbf{K}\bar{\mathbf{x}}) \right) = J^T(\mathbf{q}) (\mathbf{K}(\mathbf{x}_d - \mathbf{x}) - \mathbf{B}\dot{\mathbf{x}}) + \mathbf{g}(\mathbf{q})$$

Admittance Controller

An admittance controller measures end-effector forces \mathbf{f}_{ext} using a wrist force-torque sensor and creates end-effector motions $\mathbf{x}(t)$ to mimic a mass-spring-damper system. Thus, the controller implements a transfer function $Y(s)$ from forces to motions.

A simple approach is to calculate the desired end-effector acceleration $\ddot{\mathbf{x}}$ by having the current state $(\mathbf{x}, \dot{\mathbf{x}})$:

$$\ddot{\mathbf{x}} = \mathbf{M}^{-1}(\mathbf{f}_{\text{ext}} - \mathbf{B}\dot{\mathbf{x}} - \mathbf{K}\bar{\mathbf{x}})$$

Using definition $\dot{\mathbf{x}} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}$: $\ddot{\mathbf{q}} = \mathbf{J}^{\dagger}(\mathbf{q})(\ddot{\mathbf{x}} - \dot{\mathbf{J}}(\mathbf{q})\dot{\mathbf{q}})$

Then, joint torques/forces are calculated by $\boldsymbol{\tau} = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}})$