Ch11: Force and Impedance 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.

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

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

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

Wrench applied by manipulator to environment

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

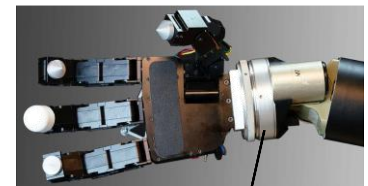
$\boldsymbol{\mathcal{F}}_d$: desired wrench

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

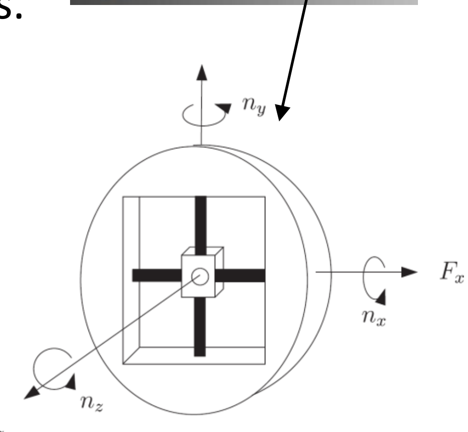
Pure Force Control

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

$$\boldsymbol{\tau} = \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{J}^T(\boldsymbol{q})\left(\boldsymbol{\mathcal{F}}_d + \boldsymbol{K}_p\boldsymbol{\mathcal{F}}_e + \boldsymbol{K}_i\int \boldsymbol{\mathcal{F}}_e(t)\mathrm{d}t\right)$$



where $\boldsymbol{\mathcal{F}}_e = \boldsymbol{\mathcal{F}}_d - \boldsymbol{\mathcal{F}}_{\text{tip}}$ and $\boldsymbol{K}_p$ and $\boldsymbol{K}_i$ are positive-definite matrices.



Closed-loop dynamics:

$$\boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{\mathcal{F}}_{\text{tip}} = \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{J}^T(\boldsymbol{q})\left(\boldsymbol{\mathcal{F}}_d + \boldsymbol{K}_p\boldsymbol{\mathcal{F}}_e + \boldsymbol{K}_i\int \boldsymbol{\mathcal{F}}_e(t)\mathrm{d}t\right)$$

$$\left(\boldsymbol{K}_p + \boldsymbol{I}\right)\boldsymbol{\mathcal{F}}_e + \boldsymbol{K}_i\int \boldsymbol{\mathcal{F}}_e(t)\mathrm{d}t = \boldsymbol{0}$$

$$\left(\boldsymbol{K}_p + \boldsymbol{I}\right)\dot{\boldsymbol{\mathcal{F}}}_e + \boldsymbol{K}_i\boldsymbol{\mathcal{F}}_e = \boldsymbol{0}$$

$\Rightarrow$ $\boldsymbol{\mathcal{F}}_e$ 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

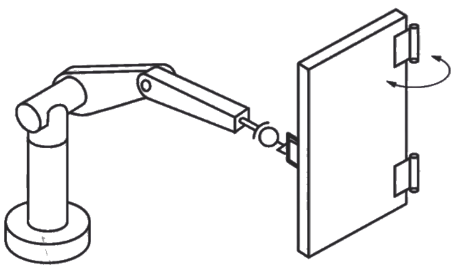
$$\boldsymbol{\tau} = \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{J}^T(\boldsymbol{q})\left(\boldsymbol{\mathcal{F}}_d + \boldsymbol{K}_p\boldsymbol{\mathcal{F}}_e + \boldsymbol{K}_i\int \boldsymbol{\mathcal{F}}_e(t)\mathrm{d}t - \boldsymbol{K}_{\mathrm{damp}}\boldsymbol{\mathcal{V}}\right)$$

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

Hybrid Motion-Force Control

Natural and Artificial Constraints

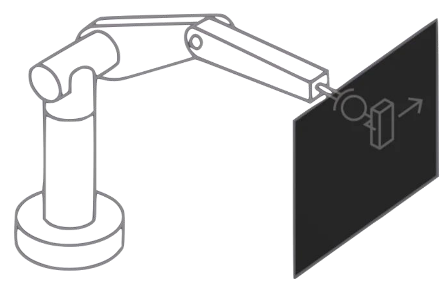
Assume that the task space is r-dimensional, and the environment is infinitely stiff (rigid constraints) in k directions and unconstrained in $r-k$ directions. Thus, there are k directions in which the robot can freely apply forces and the $r-k$ directions of free motion.



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.

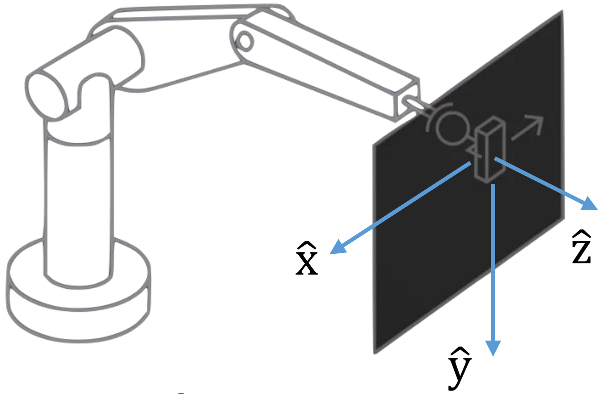
- If there is friction at the contact, $k=3$.
- If the robot move away from the board, $k=0$. Thun, velocity constraint is inequality.



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

$$\mathcal{V}_b = (\omega_x, \omega_y, \omega_z, v_x, v_y, v_z) \qquad \mathcal{F}_b = (m_x, m_y, m_z, f_x, f_y, f_z)$$



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:

$\omega_x = 0$ $f_x = 0$
$\omega_y = 0$ $f_y = 0$
$v_z = 0$ $m_z = 0$

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

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

$$\downarrow$$

$$\boldsymbol{A}(\boldsymbol{q})\dot{\boldsymbol{\mathcal{V}}} + \dot{\boldsymbol{A}}(\boldsymbol{q})\boldsymbol{\mathcal{V}} = \boldsymbol{0} \qquad (2)$$

Task-space Dynamics:

$$\boldsymbol{\mathcal{F}} = \boldsymbol{M}_C(\boldsymbol{q})\dot{\boldsymbol{\mathcal{V}}} + \boldsymbol{c}_C(\boldsymbol{q},\boldsymbol{\mathcal{V}}) + \boldsymbol{g}_C(\boldsymbol{q}) = \boldsymbol{M}_C(\boldsymbol{q})\dot{\boldsymbol{\mathcal{V}}} + \boldsymbol{h}_C(\boldsymbol{q},\boldsymbol{\mathcal{V}}) \qquad (3)$$

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

Constrained Dynamics:

$$\boldsymbol{\mathcal{F}} = \boldsymbol{M}_C(\boldsymbol{q})\dot{\boldsymbol{\mathcal{V}}} + \boldsymbol{h}_C(\boldsymbol{q},\boldsymbol{\mathcal{V}}) + \underbrace{\boldsymbol{A}^T(\boldsymbol{q})\boldsymbol{\lambda}}_{\boldsymbol{\mathcal{F}}_{\text{tip}}} \qquad (4)$$

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

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

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

Hybrid Motion-Force Control

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

$$\boldsymbol{\lambda} = \left(\boldsymbol{A}\boldsymbol{M}_C{}^{-1}\boldsymbol{A}^{\mathrm{T}}\right)^{-1}\left(\boldsymbol{A}\boldsymbol{M}_C{}^{-1}(\boldsymbol{\mathcal{F}} - \boldsymbol{h}_C) - \boldsymbol{A}\dot{\boldsymbol{\mathcal{V}}}\right) \quad (5)$$

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

$$\boldsymbol{P}(\boldsymbol{q})\boldsymbol{\mathcal{F}} = \boldsymbol{P}(\boldsymbol{q})\left(\boldsymbol{M}_C\dot{\boldsymbol{\mathcal{V}}} + \boldsymbol{h}_C\right)$$

$$\boldsymbol{P}(\boldsymbol{q}) = \boldsymbol{I} - \boldsymbol{A}^{\mathrm{T}}\left(\boldsymbol{A}\boldsymbol{M}_C{}^{-1}\boldsymbol{A}^{\mathrm{T}}\right)^{-1}\boldsymbol{A}\boldsymbol{M}_C{}^{-1} \in \mathbb{R}^n$$

$$\mathrm{rank}(\boldsymbol{P}) = 6-k$$
$$\boldsymbol{I} = \mathrm{diag}(1) \in \mathbb{R}^n$$

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

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

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

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

$$\boldsymbol{y} = \frac{d}{dt}\left(\left[\mathrm{Ad}_{\boldsymbol{T}^{-1}\boldsymbol{T}_d}\right]\boldsymbol{\mathcal{V}}_d\right) + \boldsymbol{K}_p\boldsymbol{T}_e + \boldsymbol{K}_d\boldsymbol{\mathcal{V}}_e$$

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

Configuration Error: $[\boldsymbol{T}_e] = \log(\boldsymbol{T}^{-1}\boldsymbol{T}_d)$

Velocity Error: $\boldsymbol{\mathcal{V}}_e = \left[\mathrm{Ad}_{\boldsymbol{T}^{-1}\boldsymbol{T}_d}\right]\boldsymbol{\mathcal{V}}_d - \boldsymbol{\mathcal{V}}$

Impedance Control

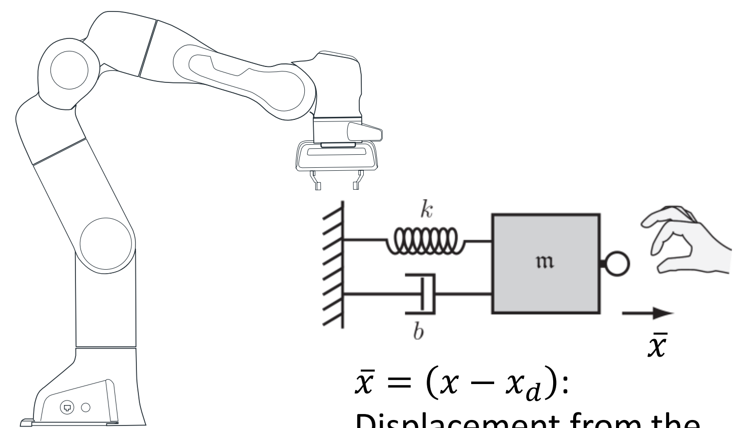
Impedance Control

(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.

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



$\bar{x} = (x - x_d)$: Displacement from the current (desired) pose.

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

Taking the Laplace transform: $(\mathfrak{m}s^2 + bs + k)X(s) = F(s)$

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 :

$$\boldsymbol{M}\ddot{\overline{\boldsymbol{x}}} + \boldsymbol{B}\dot{\overline{\boldsymbol{x}}} + \boldsymbol{K}\overline{\boldsymbol{x}} = \boldsymbol{f}_{\text{ext}}$$

$\overline{\boldsymbol{x}}, \boldsymbol{f}_{\text{ext}} \in \mathbb{R}^m$

$\boldsymbol{M}, \boldsymbol{B}, \boldsymbol{K} \in \mathbb{R}^{m\times m}$, PD

$\overline{\boldsymbol{x}} = (\boldsymbol{x} - \boldsymbol{x}_d)$

$\ddot{\boldsymbol{x}}_d = \dot{\boldsymbol{x}}_d = \boldsymbol{0}$

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

A control law is

$$\boldsymbol{\tau} = \boldsymbol{J}^T(\boldsymbol{q})\Big(\underbrace{\boldsymbol{M}_C(\boldsymbol{q})\ddot{\boldsymbol{x}} + \boldsymbol{h}_C(\boldsymbol{q}, \dot{\boldsymbol{x}})}_{\text{robot dynamics compensation}} - \underbrace{\big(\boldsymbol{M}\ddot{\overline{\boldsymbol{x}}} + \boldsymbol{B}\dot{\overline{\boldsymbol{x}}} + \boldsymbol{K}\overline{\boldsymbol{x}}\big)}_{\boldsymbol{f}_{\text{ext}}}\Big) \qquad \overline{\boldsymbol{x}} = (\boldsymbol{x} - \boldsymbol{x}_d)$$

Note: Since measurement of the acceleration $\ddot{\boldsymbol{x}}$ is likely to be noisy, it is not uncommon to eliminate the mass compensation term $\boldsymbol{M}_C(\boldsymbol{q})\ddot{\boldsymbol{x}}$ and to set $\boldsymbol{M} = \boldsymbol{0}$. 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., $\boldsymbol{h}_C(\boldsymbol{q}, \dot{\boldsymbol{x}}) = \boldsymbol{g}_C(\boldsymbol{q})$.

$$\boldsymbol{\tau} = \boldsymbol{J}^T(\boldsymbol{q})\Big(\boldsymbol{g}_C(\boldsymbol{q}) - \big(\boldsymbol{B}\dot{\overline{\boldsymbol{x}}} + \boldsymbol{K}\overline{\boldsymbol{x}}\big)\Big) = \boldsymbol{J}^T(\boldsymbol{q})(\boldsymbol{K}(\boldsymbol{x}_d - \boldsymbol{x}) - \boldsymbol{B}\dot{\boldsymbol{x}}) + \boldsymbol{g}(\boldsymbol{q})$$

Admittance Controller

An admittance controller measures end-effector forces $\boldsymbol{f}_{\text{ext}}$ using a wrist force-torque sensor and creates end-effector motions $\boldsymbol{x}(t)$ 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 $\ddot{\boldsymbol{x}}$ by having the current state $(\boldsymbol{x}, \dot{\boldsymbol{x}})$:

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

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

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