Ch10: Centralized Control - Motion Control – Part 1

Motion Control

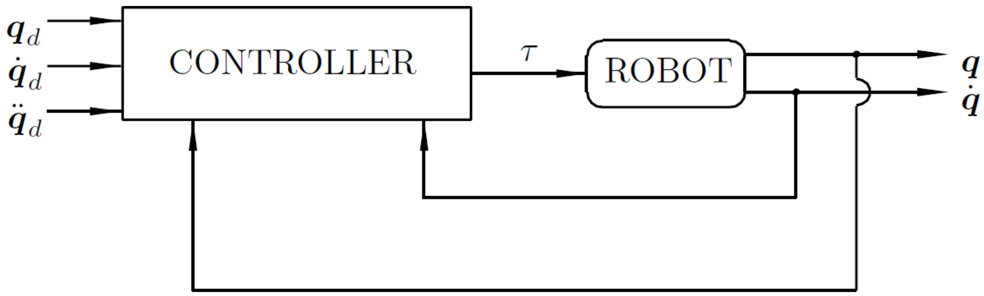
Motion Control Objective

Given desired joint position $\boldsymbol{q}_d(t) \in \mathbb{R}^n$, velocity $\dot{\boldsymbol{q}}_d(t) \in \mathbb{R}^n$, and acceleration $\ddot{\boldsymbol{q}}_d(t) \in \mathbb{R}^n$, we wish to find joint torques/forces $\boldsymbol{\tau} \in \mathbb{R}^n$ such that the joint position $\boldsymbol{q}(t) \in \mathbb{R}^n$ follow (asymptotically) $\boldsymbol{q}_d(t)$ accurately:

$$\lim_{t\to\infty} \boldsymbol{q}(t) = \boldsymbol{q}_d(t) \quad \Rightarrow \quad \lim_{t\to\infty} \boldsymbol{e}(t) = \boldsymbol{0}$$

$$\boldsymbol{e}(t) = \boldsymbol{q}_d(t) - \boldsymbol{q}(t) \in \mathbb{R}^n$$

position error



$$\dot{\boldsymbol{e}}(t) = \dot{\boldsymbol{q}}_d(t) - \dot{\boldsymbol{q}}(t) \in \mathbb{R}^n$$

velocity error

The most common motion controllers:

- PD/PID Control
- PD with Feedforward Control
- PD Control with Gravity Compensation
- PD+ Control
- Inverse Dynamics Control

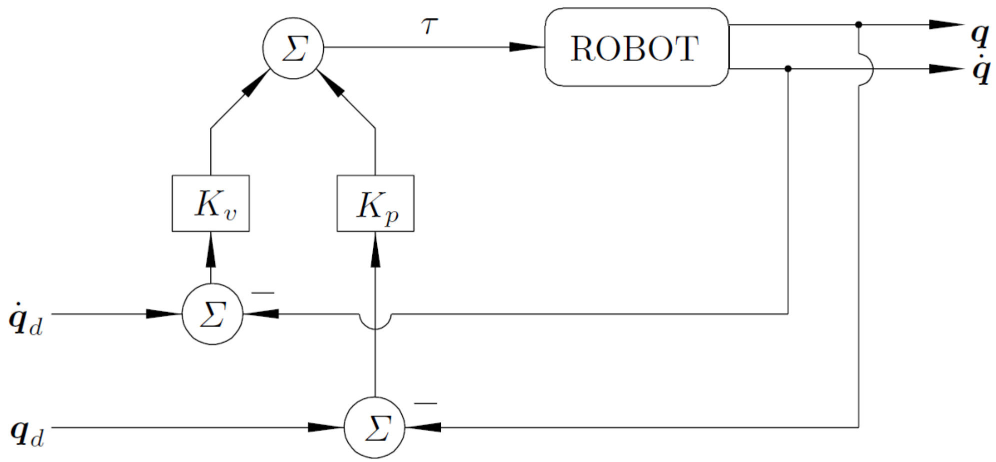
PD/PID Control

PD Control

The PD control law is given by $\boldsymbol{\tau} = \boldsymbol{K}_p \boldsymbol{e} + \boldsymbol{K}_v \dot{\boldsymbol{e}}$ $\boldsymbol{e} = \boldsymbol{q}_d - \boldsymbol{q}$

$\boldsymbol{K}_p, \boldsymbol{K}_v \in \mathbb{R}^{n\times n}$ are symmetric positive definite matrices. If $\boldsymbol{K}_p = \mathrm{diag}\{K_{p,i}\}$, $\boldsymbol{K}_v = \mathrm{diag}\{K_{v,i}\}$, the controller is called PD Independent Joint Control.

This is the simplest (linear) controller that may be used to control robot manipulators.



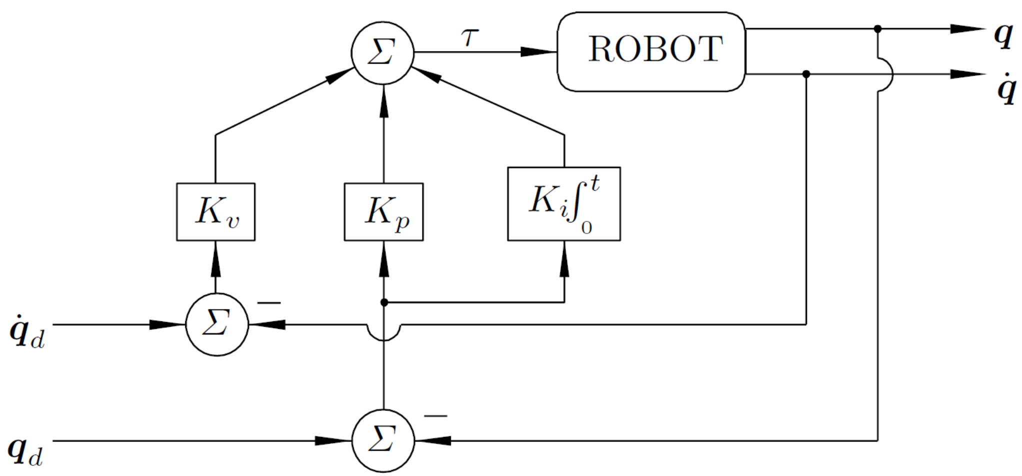
- For <u>any</u> $\boldsymbol{K}_p = \boldsymbol{K}_p^T > 0$, $\boldsymbol{K}_v = \boldsymbol{K}_v^T > 0$, it is guaranteed that $\boldsymbol{e}(t)$ and $\dot{\boldsymbol{e}}(t)$ are <u>bounded</u> for all initial conditions.
- The error bound decreases, as $K_{v,i}$ become larger (in case $\boldsymbol{K}_v = \mathrm{diag}\{K_{v,i}\}$), however, large $K_{v,i}$ can saturate the robot actuators.

PID Control

The residual error at steady state due to gravity with PD control can be removed to some extend using the PID control law which is given by

$$\boldsymbol{\tau} = \boldsymbol{K}_p\boldsymbol{e} + \boldsymbol{K}_v\dot{\boldsymbol{e}} + \boldsymbol{K}_i\int_0^t \boldsymbol{e}(\tau)d\tau \qquad\qquad \boldsymbol{e} = \boldsymbol{q}_d - \boldsymbol{q}$$

$\boldsymbol{K}_p, \boldsymbol{K}_v, \boldsymbol{K}_i \in \mathbb{R}^{n\times n}$ (position, velocity, and integral gains) are symmetric positive definite matrices. If $\boldsymbol{K}_p = \mathrm{diag}\{K_{p,i}\}$, $\boldsymbol{K}_v = \mathrm{diag}\{K_{v,i}\}$, $\boldsymbol{K}_i = \mathrm{diag}\{K_{i,i}\}$, the controller is called PID Independent Joint Control.



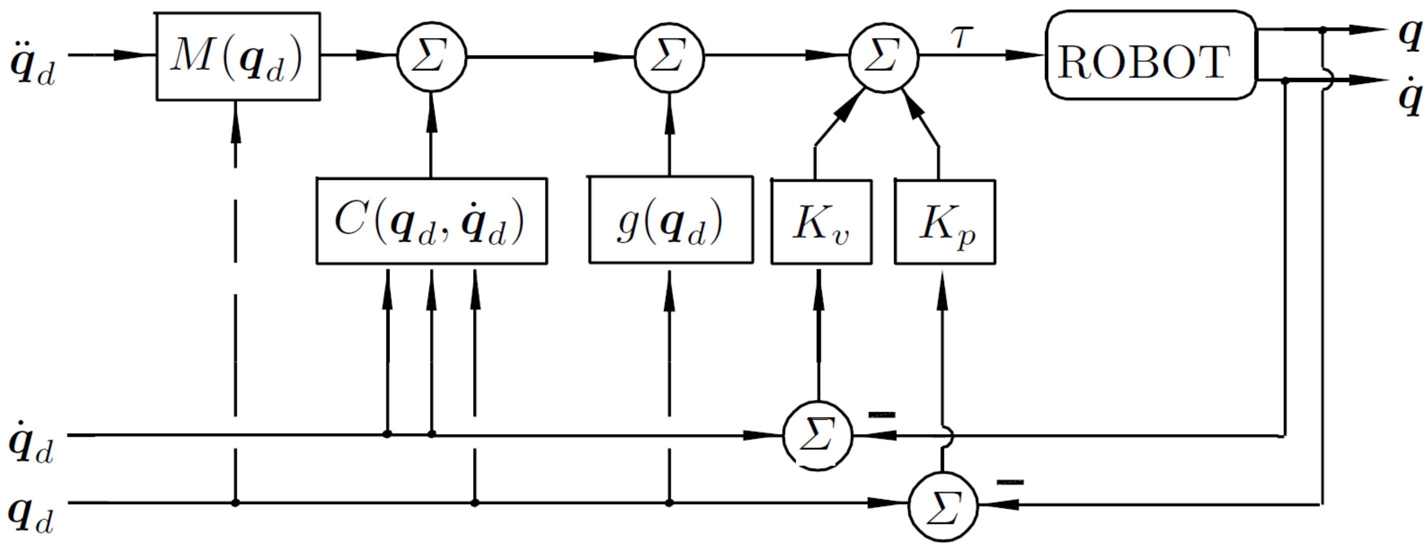
PD with Feedforward Control

PD with Feedforward Control

The PD with Feedforward control law is given by

$$\boldsymbol{\tau} = \boldsymbol{K}_p \boldsymbol{e} + \boldsymbol{K}_v \dot{\boldsymbol{e}} + \underbrace{\boldsymbol{M}(\boldsymbol{q}_d)\ddot{\boldsymbol{q}}_d + \boldsymbol{C}(\boldsymbol{q}_d, \dot{\boldsymbol{q}}_d)\dot{\boldsymbol{q}}_d + \boldsymbol{g}(\boldsymbol{q}_d)}_{\text{For reducing the number of computations in real time implementation.}} \qquad \boldsymbol{e} = \boldsymbol{q}_d - \boldsymbol{q}$$

$\boldsymbol{K}_p, \boldsymbol{K}_v \in \mathbb{R}^{n\times n}$ are symmetric positive definite matrices.



PD with Feedforward Control (cont.)

The closed-loop dynamic equation is derived as

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{K}_p\boldsymbol{e} + \boldsymbol{K}_v\dot{\boldsymbol{e}} + \boldsymbol{M}(\boldsymbol{q}_d)\ddot{\boldsymbol{q}}_d + \boldsymbol{C}(\boldsymbol{q}_d,\dot{\boldsymbol{q}}_d)\dot{\boldsymbol{q}}_d + \boldsymbol{g}(\boldsymbol{q}_d)$$

$$\boldsymbol{q} = \boldsymbol{q}_d - \boldsymbol{e}$$

$$\frac{d}{dt}\begin{bmatrix}\boldsymbol{e}\\ \dot{\boldsymbol{e}}\end{bmatrix} = \begin{bmatrix}\dot{\boldsymbol{e}}\\ \boldsymbol{M}(\boldsymbol{q})^{-1}[-\boldsymbol{K}_p\boldsymbol{e} - \boldsymbol{K}_v\dot{\boldsymbol{e}} - \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{e}} - \boldsymbol{h}(t,\boldsymbol{e},\dot{\boldsymbol{e}})]\end{bmatrix}$$

$$\boldsymbol{h}(t,\boldsymbol{e},\dot{\boldsymbol{e}}) = [\boldsymbol{M}(\boldsymbol{q}_d) - \boldsymbol{M}(\boldsymbol{q})]\ddot{\boldsymbol{q}}_d + [\boldsymbol{C}(\boldsymbol{q}_d,\dot{\boldsymbol{q}}_d) - \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})]\dot{\boldsymbol{q}}_d + [\boldsymbol{g}(\boldsymbol{q}_d) - \boldsymbol{g}(\boldsymbol{q})]$$

We can show that

- Origin $(\boldsymbol{e},\dot{\boldsymbol{e}}) = \boldsymbol{0} \in \mathbb{R}^{2n}$ is an equilibrium, independently of the gain matrices $\boldsymbol{K}_p, \boldsymbol{K}_v$.
- The number of equilibria of the system depends on the proportional gain $\boldsymbol{K}_p$.
- By choosing $\boldsymbol{K}_p$ sufficiently large, then the system has a unique equilibrium at origin.
- By choosing $\boldsymbol{K}_p, \boldsymbol{K}_v$ sufficiently large, the origin is globally uniformly asymptoticly stable.

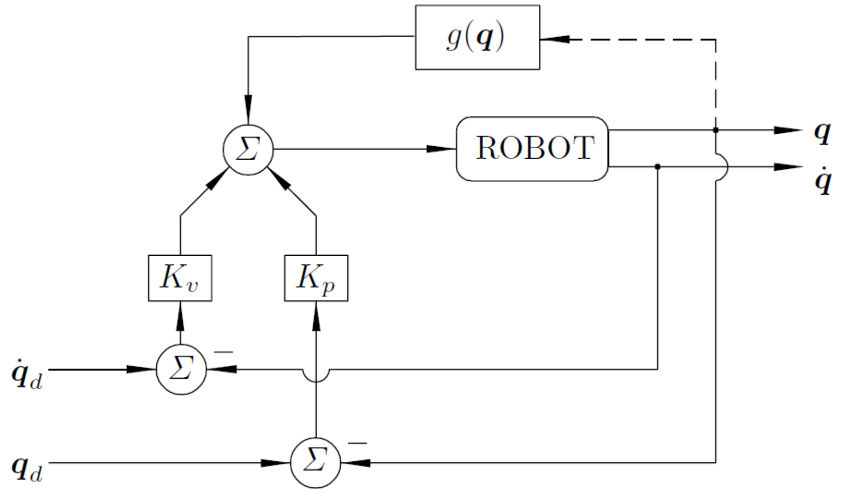
PD Control with Gravity Compensation

PD Control with Gravity Compensation

The PD control law with gravity compensation is given by $\boldsymbol{\tau} = \boldsymbol{K}_p \boldsymbol{e} + \boldsymbol{K}_v \dot{\boldsymbol{e}} + \boldsymbol{g}(\boldsymbol{q})$

$\boldsymbol{e} = \boldsymbol{q}_d - \boldsymbol{q}$

$\boldsymbol{K}_p, \boldsymbol{K}_v \in \mathbb{R}^{n\times n}$ are symmetric positive definite matrices.



- In general, if $\boldsymbol{q}_d$ is not constant (i.e., motion control), then the controller guarantees <u>bounded</u> tracking errors $\boldsymbol{e}(t)$ about zero, but <u>the error never goes exactly to zero</u>.
- The error bound decreases, as the PD gains become larger (in case $\boldsymbol{K}_p = \mathrm{diag}\{K_{p,i}\}$, $\boldsymbol{K}_v = \mathrm{diag}\{K_{v,i}\}$).
- PD Controller with Gravity Compensation may behave batter than a PID controller for the same PD gains. However, the integral term can reject other terms besides gravity (e.g., friction of some sorts).

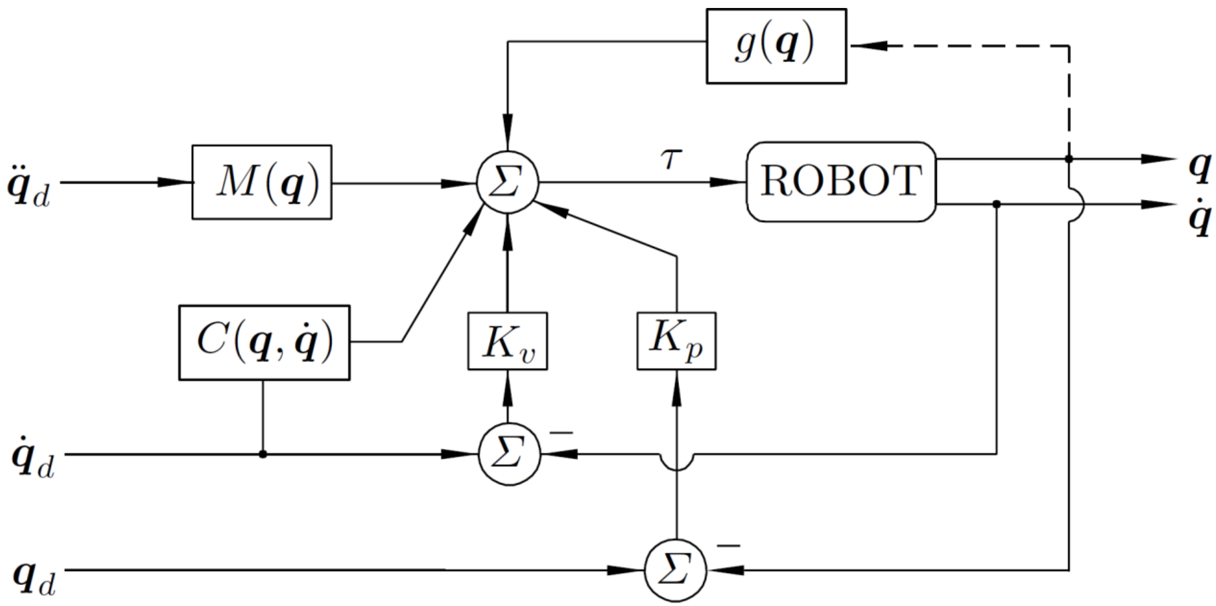
PD+ Control

PD+ Control

The PD+ control law is given by

$$\boldsymbol{\tau} = \boldsymbol{K}_p \boldsymbol{e} + \boldsymbol{K}_v \dot{\boldsymbol{e}} + \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}}_d + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}}_d + \boldsymbol{g}(\boldsymbol{q}) \qquad \boldsymbol{e} = \boldsymbol{q}_d - \boldsymbol{q}$$

$\boldsymbol{K}_p, \boldsymbol{K}_v \in \mathbb{R}^{n\times n}$ are symmetric positive definite matrices.



PD+ Control (cont.)

The closed-loop dynamic equation is derived as $\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{e}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{e}} = -\boldsymbol{K}_p\boldsymbol{e} - \boldsymbol{K}_v\dot{\boldsymbol{e}}$

$$\frac{d}{dt}\begin{bmatrix}\boldsymbol{e}\\ \dot{\boldsymbol{e}}\end{bmatrix} = \begin{bmatrix}\dot{\boldsymbol{e}}\\ \boldsymbol{M}(\boldsymbol{q})^{-1}\left[-\boldsymbol{K}_p\boldsymbol{e} - \boldsymbol{K}_v\dot{\boldsymbol{e}} - \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{e}}\right]\end{bmatrix} \qquad \boldsymbol{q} = \boldsymbol{q}_d - \boldsymbol{e}$$

The system is **nonautonomous**, and origin $(\boldsymbol{e},\dot{\boldsymbol{e}}) = \boldsymbol{0} \in \mathbb{R}^{2n}$ is the only equilibrium point.

Consider a Lyapunov function candidate as $V(t,\boldsymbol{e},\dot{\boldsymbol{e}}) = \frac{1}{2}\dot{\boldsymbol{e}}^T\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{e}} + \frac{1}{2}\boldsymbol{e}^T\boldsymbol{K}_p\boldsymbol{e} > 0$.

$$\dot{V}(t,\boldsymbol{e},\dot{\boldsymbol{e}}) = \dot{\boldsymbol{e}}^T\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{e}} + \frac{1}{2}\dot{\boldsymbol{e}}^T\dot{\boldsymbol{M}}(\boldsymbol{q})\dot{\boldsymbol{e}} + \boldsymbol{e}^T\boldsymbol{K}_p\dot{\boldsymbol{e}}$$

Using closed-loop dynamic equation, and

$$\dot{\boldsymbol{e}}^T\left[\frac{1}{2}\dot{\boldsymbol{M}}(\boldsymbol{q}) - \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\right]\dot{\boldsymbol{e}} = \boldsymbol{0}$$

$$\dot{V}(t,\boldsymbol{e},\dot{\boldsymbol{e}}) = -\dot{\boldsymbol{e}}^T\boldsymbol{K}_v\dot{\boldsymbol{e}} \le 0$$

Thus, the origin $(\boldsymbol{e},\dot{\boldsymbol{e}}) = \boldsymbol{0}$ is <u>stable</u>.

- Using more advance theorems (e.g., Matrosov's theorem) or a different Lyapunov function, we can show that the origin is <u>globally uniformly asymptoticly stable</u>.

Inverse Dynamics Control

Inverse Dynamics Control
(or Computed Torque Control)

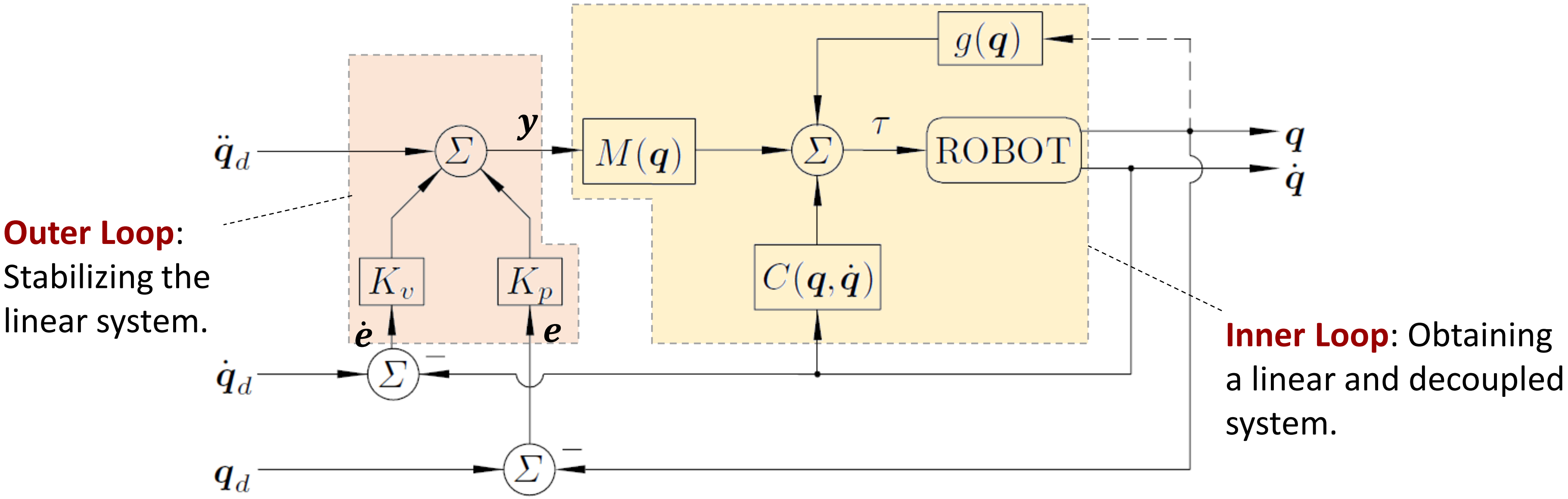
The idea of Inverse Dynamics control is to seek a nonlinear feedback control law which results in a linear closed-loop system. The control law is given by

$$\boldsymbol{\tau} = \boldsymbol{M}(\boldsymbol{q})\boldsymbol{y} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) \qquad \boldsymbol{e} = \boldsymbol{q}_d - \boldsymbol{q}$$

$$\boldsymbol{y} = \ddot{\boldsymbol{q}}_d + \boldsymbol{K}_v\dot{\boldsymbol{e}} + \boldsymbol{K}_p\boldsymbol{e}$$ ($\rightarrow$ PD control with feedforward acceleration)

$\boldsymbol{K}_p, \boldsymbol{K}_v \in \mathbb{R}^{n \times n}$ are symmetric positive definite matrices.

This is a model-based motion control approach.



Inverse Dynamics Control

The closed-loop dynamic equation is derived as

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{M}(\boldsymbol{q})\boldsymbol{y} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) \longrightarrow \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} = \boldsymbol{M}(\boldsymbol{q})\boldsymbol{y}$$

(n uncoupled linear double integrators) $$\ddot{\boldsymbol{q}} = \boldsymbol{y} \longrightarrow \ddot{\boldsymbol{q}} = \ddot{\boldsymbol{q}}_d + \boldsymbol{K}_v\dot{\boldsymbol{e}} + \boldsymbol{K}_p\boldsymbol{e}$$

$$\ddot{\boldsymbol{e}} + \boldsymbol{K}_v\dot{\boldsymbol{e}} + \boldsymbol{K}_p\boldsymbol{e} = \boldsymbol{0} \longrightarrow \frac{d}{dt}\begin{bmatrix}\boldsymbol{e}\\ \dot{\boldsymbol{e}}\end{bmatrix} = \begin{bmatrix}\dot{\boldsymbol{e}}\\ -\boldsymbol{K}_p\boldsymbol{e} - \boldsymbol{K}_v\dot{\boldsymbol{e}}\end{bmatrix} = \begin{bmatrix}\boldsymbol{0} & \boldsymbol{I}_n\\ -\boldsymbol{K}_p & -\boldsymbol{K}_v\end{bmatrix}\begin{bmatrix}\boldsymbol{e}\\ \dot{\boldsymbol{e}}\end{bmatrix}$$

The system is **linear** and **autonomous**, and the origin $(\boldsymbol{e},\dot{\boldsymbol{e}}) = \boldsymbol{0} \in \mathbb{R}^{2n}$ is the unique equilibrium point.

Let's introduce the constant $\varepsilon \in \mathbb{R}_{++}$ satisfying $\lambda_{\min}(\boldsymbol{K}_v) > \varepsilon > 0$, thus, $\lambda_{\min}(\boldsymbol{K}_v)\boldsymbol{x}^T\boldsymbol{x} > \varepsilon\boldsymbol{x}^T\boldsymbol{x}$, and since $\lambda_{\min}(\boldsymbol{K}_v)\boldsymbol{x}^T\boldsymbol{x} \le \boldsymbol{x}^T\boldsymbol{K}_v\boldsymbol{x}$, then $\boldsymbol{x}^T(\boldsymbol{K}_v - \varepsilon\boldsymbol{I}_n)\boldsymbol{x} > 0\ \forall\boldsymbol{x} \neq \boldsymbol{0} \in \mathbb{R}^n$. This means $\boldsymbol{K}_v - \varepsilon\boldsymbol{I}_n > 0$ and $\boldsymbol{K}_p + \varepsilon\boldsymbol{K}_v - \varepsilon^2\boldsymbol{I}_n > 0$. Now, a Lyapunov function candidate is

$$V(\boldsymbol{e},\dot{\boldsymbol{e}}) = \frac{1}{2}\begin{bmatrix}\boldsymbol{e}\\ \dot{\boldsymbol{e}}\end{bmatrix}^T\begin{bmatrix}\boldsymbol{K}_p + \varepsilon\boldsymbol{K}_v & \varepsilon\boldsymbol{I}_n\\ \varepsilon\boldsymbol{I}_n & \boldsymbol{I}_n\end{bmatrix}\begin{bmatrix}\boldsymbol{e}\\ \dot{\boldsymbol{e}}\end{bmatrix} = \frac{1}{2}[\dot{\boldsymbol{e}} + \varepsilon\boldsymbol{e}]^T[\dot{\boldsymbol{e}} + \varepsilon\boldsymbol{e}] + \frac{1}{2}\boldsymbol{e}^T[\boldsymbol{K}_p + \varepsilon\boldsymbol{K}_v - \varepsilon^2\boldsymbol{I}_n]\boldsymbol{e} > 0$$

$$\Rightarrow \quad V(\boldsymbol{e},\dot{\boldsymbol{e}}) = \frac{1}{2}\dot{\boldsymbol{e}}^T\dot{\boldsymbol{e}} + \frac{1}{2}\boldsymbol{e}^T[\boldsymbol{K}_p + \varepsilon\boldsymbol{K}_v]\boldsymbol{e} + \varepsilon\boldsymbol{e}^T\dot{\boldsymbol{e}} > 0$$

Inverse Dynamics Control

$$\dot{V}(\boldsymbol{e},\dot{\boldsymbol{e}}) = \ddot{\boldsymbol{e}}^T\dot{\boldsymbol{e}} + \boldsymbol{e}^T\left[\boldsymbol{K}_p + \varepsilon\boldsymbol{K}_v\right]\dot{\boldsymbol{e}} + \varepsilon\dot{\boldsymbol{e}}^T\dot{\boldsymbol{e}} + \varepsilon\boldsymbol{e}^T\ddot{\boldsymbol{e}} \xrightarrow{\ddot{\boldsymbol{e}} + \boldsymbol{K}_v\dot{\boldsymbol{e}} + \boldsymbol{K}_p\boldsymbol{e} = \boldsymbol{0}}$$

$$\Rightarrow \quad \dot{V}(\boldsymbol{e},\dot{\boldsymbol{e}}) = -\dot{\boldsymbol{e}}^T[\boldsymbol{K}_v - \varepsilon\boldsymbol{I}_n]\dot{\boldsymbol{e}} - \varepsilon\boldsymbol{e}^T\boldsymbol{K}_p\boldsymbol{e} < 0$$

Thus, the origin $(\boldsymbol{e},\dot{\boldsymbol{e}}) = \boldsymbol{0}$ is <u>globally asymptotically stable</u> for any initial condition $\boldsymbol{q}(0), \dot{\boldsymbol{q}}(0) \in \mathbb{R}^n$:

$$\lim_{t\to\infty} \boldsymbol{e}(t) = \boldsymbol{0} \qquad \lim_{t\to\infty} \dot{\boldsymbol{e}}(t) = \boldsymbol{0}$$

$\Rightarrow$ Thus, the motion control objective is achieved.

Note: Since the closed-loop equation is linear and autonomous, the origin is <u>globally exponentially stable</u>.

Note: Friction at the joints may also affect the position error. Moreover, in the presence of bounded disturbance $\boldsymbol{\tau}_{\text{dist}}(t)$, error $\boldsymbol{e}(t)$ remains bounded.

Note: This controller is a special case of the method of <u>**feedback linearization**</u> for nonlinear systems.

Inverse Dynamics Control: Parameter Selection

$\boldsymbol{K}_p$ and $\boldsymbol{K}_v$ may be chosen diagonal as:

$$\boldsymbol{K}_p = \text{diag}\{K_{p,i}\} = \text{diag}\{\omega_{n,i}^2\}$$
$$\boldsymbol{K}_v = \text{diag}\{K_{v,i}\} = \text{diag}\{2\zeta_i\omega_{n,i}\}$$

With this choice, the closed-loop equation is n **decoupled** 2nd-order linear ODEs. The natural frequency $\omega_{n,i} \in \mathbb{R}$ determines the speed of response (the larger, the faster) and the damping ratio $\zeta_i \in \mathbb{R}$ determines the existence of overshoot in joint error $\boldsymbol{e}(t)$.

Note 1: It may be useful to select the desired responses at the end of the arm faster than near the base, where the masses that must be moved are heavier.

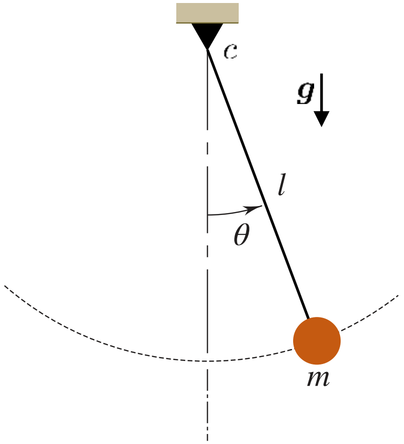
Note 2: It is undesirable for the robot to exhibit overshoot (e.g., since this could cause impact for paths near the workpiece surface). Therefore, the damping ratios are usually selected $\zeta_i = 1$ to have a critically damped responses.

Note 3: If the gains $K_{p,i}, K_{v,i}$ are too large, the control torque may reach its upper limits and saturate some or all of the actuators.

Note 4: The implementation of this control scheme requires the real-time computation of $\boldsymbol{M}(\boldsymbol{q})$ and $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$, and $\boldsymbol{g}(\boldsymbol{q})$, which may be computationally expensive.

Inverse Dynamics Control: Example

Consider the equation of a pendulum of length l and mass m concentrated at its tip, subject to the action of gravity g and to which is applied a torque τ at the axis of rotation. Drive the inverse dynamics control law.



$$ml^2\ddot{\theta} + mgl\sin\theta = \tau$$

Approximate Inverse Dynamics Control

In some cases, $\boldsymbol{M}(\boldsymbol{q})$ is not known exactly (e.g., unknown payload mass), or $\boldsymbol{h}(\boldsymbol{q},\dot{\boldsymbol{q}}) = \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q})$ is not known exactly (e.g., unknown friction terms). Then $\widehat{\boldsymbol{M}}(\boldsymbol{q})$ and $\widehat{\boldsymbol{h}}(\boldsymbol{q},\dot{\boldsymbol{q}})$ could be the best estimate we have for these terms. On the other hand, we might simply wish to avoid computing $\boldsymbol{M}(\boldsymbol{q})$ and $\boldsymbol{h}(\boldsymbol{q},\dot{\boldsymbol{q}})$ at each short sample time and instead compute more simpler $\widehat{\boldsymbol{M}}(\boldsymbol{q})$ and $\widehat{\boldsymbol{h}}(\boldsymbol{q},\dot{\boldsymbol{q}})$. The approximate inverse dynamics control law is given by

$$\boldsymbol{\tau} = \widehat{\boldsymbol{M}}(\boldsymbol{q})\boldsymbol{y} + \widehat{\boldsymbol{h}}(\boldsymbol{q},\dot{\boldsymbol{q}})$$

$$\boldsymbol{y} = \ddot{\boldsymbol{q}}_d + \boldsymbol{K}_v\dot{\boldsymbol{e}} + \boldsymbol{K}_p\boldsymbol{e} \qquad \boldsymbol{e} = \boldsymbol{q}_d - \boldsymbol{q}$$

It can be shown that even if $\widehat{\boldsymbol{M}} \neq \boldsymbol{M}$ and $\widehat{\boldsymbol{h}} \neq \boldsymbol{h}$, the performance of the controller can be <u>acceptable</u> if the symmetric positive definite matrices $\boldsymbol{K}_p, \boldsymbol{K}_v \in \mathbb{R}^{n\times n}$ are selected large enough.

PID Inverse Dynamics Control

In the presence of unknown constant disturbances ($\boldsymbol{\tau}_{\text{dist}}$), PD control gives a nonzero steady-state error. Thus, we by including an integrator (I) in the outer loop, we can achieve a PID inverse dynamics controller as

$$\boldsymbol{\tau} = \boldsymbol{M}(\boldsymbol{q})\boldsymbol{y} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q})$$

$$\boldsymbol{e} = \boldsymbol{q}_d - \boldsymbol{q}$$

$$\boldsymbol{y} = \ddot{\boldsymbol{q}}_d + \boldsymbol{K}_v\dot{\boldsymbol{e}} + \boldsymbol{K}_p\boldsymbol{e} + \boldsymbol{K}_i\int_0^t \boldsymbol{e}(\tau)d\tau$$

$\boldsymbol{K}_p, \boldsymbol{K}_v, \boldsymbol{K}_i \in \mathbb{R}^{n\times n}$ are symmetric positive definite matrices.

The closed-loop dynamic equation is derived as $\dddot{\boldsymbol{e}} + \boldsymbol{K}_v\ddot{\boldsymbol{e}} + \boldsymbol{K}_p\dot{\boldsymbol{e}} + \boldsymbol{K}_i\boldsymbol{e} = \boldsymbol{0}$

Note: If control gains are diagonal ($\boldsymbol{K}_p = \text{diag}\{K_{p,i}\}$, $\boldsymbol{K}_v = \text{diag}\{K_{v,i}\}$, $\boldsymbol{K}_i = \text{diag}\{K_{i,i}\}$), for closed-loop stability, based on Routh-Hurwitz criterion, we require that

$$K_{i,i} < K_{p,i}K_{v,i}$$

Task Space Control

Task Space Control

Since the robot interacts with the external environment and objects in it, it may be more convenient to express the motion as a trajectory of the end-effector in task space. If the end-effector trajectory be specified by $\boldsymbol{x}_d(t) \in \mathbb{R}^m$ or $\boldsymbol{T}_d(t) \in SE(3), \boldsymbol{\mathcal{V}}_d(t) \in \mathbb{R}^6$:

Method 1: Converting a desired trajectory in task space to joint-space and proceed with joint-space control.

$$\begin{cases} \boldsymbol{q}_d(t) = \boldsymbol{f}^{-1}(\boldsymbol{x}_d(t)) \\ \dot{\boldsymbol{q}}_d(t) = \bar{\boldsymbol{f}}^{-1}(\dot{\boldsymbol{x}}_d(t)) \\ \ddot{\boldsymbol{q}}_d(t) = \bar{\bar{\boldsymbol{f}}}^{-1}(\ddot{\boldsymbol{x}}_d(t)) \end{cases} \quad \text{or} \quad \begin{cases} \boldsymbol{q}_d(t) = \boldsymbol{T}^{-1}(\boldsymbol{T}_d(t)) \\ \dot{\boldsymbol{q}}_d(t) = \boldsymbol{J}^{\dagger}(\boldsymbol{q}_d(t))\boldsymbol{\mathcal{V}}_d(t) \\ \ddot{\boldsymbol{q}}_d(t) = \boldsymbol{J}^{\dagger}(\boldsymbol{q}_d(t))\left(\dot{\boldsymbol{\mathcal{V}}}_d(t) - \dot{\boldsymbol{J}}(\boldsymbol{q}_d(t))\dot{\boldsymbol{q}}_d(t)\right) \end{cases}$$

Drawback: This requires significant computing power. To reduce the computational load, we can first compute $\boldsymbol{q}_d(t)$, then perform a numerical differentiation to compute $\dot{\boldsymbol{q}}_d(t)$ and $\ddot{\boldsymbol{q}}_d(t)$.

Task Space Control (cont.)

Method 2: Developing a control law in the task space using the robot dynamics expressed either in joint space or task space.

$$\boldsymbol{\tau} = \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q})$$

$$\boldsymbol{\mathcal{F}} = \boldsymbol{M}_C(\boldsymbol{q})\dot{\boldsymbol{\mathcal{V}}} + \boldsymbol{C}_C(\boldsymbol{\theta},\boldsymbol{\mathcal{V}})\boldsymbol{\mathcal{V}} + \boldsymbol{g}_C(\boldsymbol{q})$$
$$\boldsymbol{\tau} = \boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{\mathcal{F}}$$

(or)

$$\boldsymbol{F} = \boldsymbol{M}_C(\boldsymbol{q})\ddot{\boldsymbol{x}} + \boldsymbol{C}_C(\boldsymbol{q},\dot{\boldsymbol{x}})\dot{\boldsymbol{x}} + \boldsymbol{g}_C(\boldsymbol{q})$$
$$\boldsymbol{\tau} = \boldsymbol{J}_a^T(\boldsymbol{q})\boldsymbol{F}$$

Difficulties:

- Task space controllers always require computation of manipulator Jacobian. Thus, the presence of **singularities** and/or **redundancy** influences the Jacobian, and the induced effects are somewhat difficult to handle with a task space controller (e.g., we must use Jacobian pseudoinverse or other redundancy handling techniques).
- Expressing the **joint limits** in easier in joint space than task space.

- Here, let's consider a nonredundant manipulator avoiding singularities to develop the control laws.

Position Control: PD Control with Gravity Compensation

(Based on Robot Dynamics in J-Space & Min. Coord. Rep. of EE Frame)

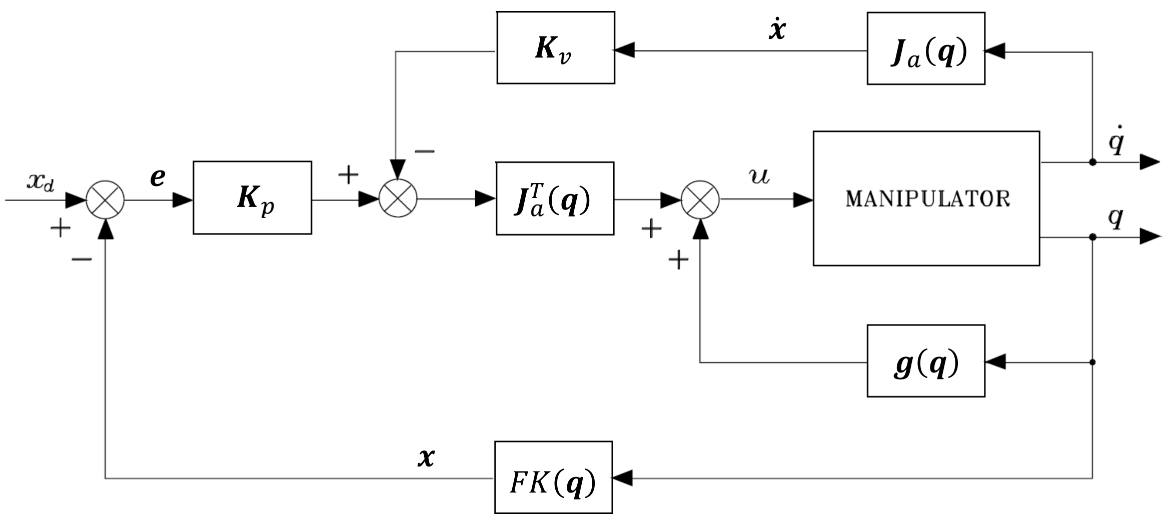
Given a constant end-effector pose $\boldsymbol{x}_d$, the PD control law with gravity compensation is given by

$$\boldsymbol{\tau} = \boldsymbol{J}_a^T(\boldsymbol{q})\left(\boldsymbol{K}_p\boldsymbol{e} - \boldsymbol{K}_v\dot{\boldsymbol{x}}\right) + \boldsymbol{g}(\boldsymbol{q})$$

$$\boldsymbol{e} = \boldsymbol{x}_d - \boldsymbol{x}$$

$$\dot{\boldsymbol{x}} = \boldsymbol{J}_a(\boldsymbol{q})\dot{\boldsymbol{q}}$$

$\boldsymbol{K}_p, \boldsymbol{K}_v \in \mathbb{R}^{n\times n}$ are symmetric positive definite matrices.



Note: If measurements of $\boldsymbol{x}$ and $\dot{\boldsymbol{x}}$ are made directly in the task space, $FK(\boldsymbol{q})$ and $\boldsymbol{J}_a(\boldsymbol{q})$ are not required; however, it is necessary to measure $\boldsymbol{q}$ to update both $\boldsymbol{J}_a^T(\boldsymbol{q})$ and $\boldsymbol{g}(\boldsymbol{q})$ on-line.

Position Control: PD Control with Gravity Compensation

(Based on Robot Dynamics in J-Space & Min. Coord. Rep. of EE Frame) (cont.)

The closed-loop dynamic equation is derived as

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{J}_a^T(\boldsymbol{q})\boldsymbol{K}_p\boldsymbol{e} - \boldsymbol{J}_a^T(\boldsymbol{q})\boldsymbol{K}_v\boldsymbol{J}_a(\boldsymbol{q})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q})$$

$$\frac{d}{dt}\begin{bmatrix}\boldsymbol{e}\\ \dot{\boldsymbol{q}}\end{bmatrix} = \begin{bmatrix}-\dot{\boldsymbol{x}}\\ \boldsymbol{M}(\boldsymbol{q})^{-1}\left(\boldsymbol{J}_a^T(\boldsymbol{q})\boldsymbol{K}_p\boldsymbol{e} - \boldsymbol{J}_a^T(\boldsymbol{q})\boldsymbol{K}_v\boldsymbol{J}_a(\boldsymbol{q})\dot{\boldsymbol{q}} - \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}\right)\end{bmatrix} \qquad \boldsymbol{x} = \boldsymbol{x}_d - \boldsymbol{e}$$

The system is **autonomous** (since $\boldsymbol{x}_d$ is constant), and it has a unique equilibrium point at origin $(\boldsymbol{e},\dot{\boldsymbol{q}}) = \boldsymbol{0} \in \mathbb{R}^{2n}$.

Consider a Lyapunov function candidate as

$$V(\dot{\boldsymbol{q}},\boldsymbol{e}) = \frac{1}{2}\dot{\boldsymbol{q}}^T\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{q}} + \frac{1}{2}\boldsymbol{e}^T\boldsymbol{K}_P\boldsymbol{e} > 0 \quad \text{(PD)}$$

Kinetic energy of the arm

$\forall \boldsymbol{e},\dot{\boldsymbol{q}} \neq \boldsymbol{0}$

$$\dot{V}(\dot{\boldsymbol{q}},\boldsymbol{e}) = \dot{\boldsymbol{q}}^T\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \frac{1}{2}\dot{\boldsymbol{q}}^T\dot{\boldsymbol{M}}(\boldsymbol{q})\dot{\boldsymbol{q}} + \dot{\boldsymbol{e}}^T\boldsymbol{K}_P\boldsymbol{e}$$

$$\dot{\boldsymbol{q}}^T\left[\frac{1}{2}\dot{\boldsymbol{M}} - \boldsymbol{C}\right]\dot{\boldsymbol{q}} = \boldsymbol{0}$$

$$\dot{V}(\dot{\boldsymbol{q}},\boldsymbol{e}) = -\dot{\boldsymbol{x}}^T\boldsymbol{K}_v\dot{\boldsymbol{x}} \leq 0 \qquad \Rightarrow$$

Using LaSalle (invariant set) theorem, the origin $(\boldsymbol{e},\dot{\boldsymbol{q}}) = \boldsymbol{0}$ is globally asymptotically stable.

$$\lim_{t\to\infty}\boldsymbol{e}(t) = \boldsymbol{0}$$

Motion Control: Inverse Dynamics Control

(Based on Robot Dynamics in J-Space & Min. Coord. Rep. of EE Frame)

The inverse dynamics control law is given by

$$\boldsymbol{\tau} = \boldsymbol{M}(\boldsymbol{q})\boldsymbol{y} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q})$$

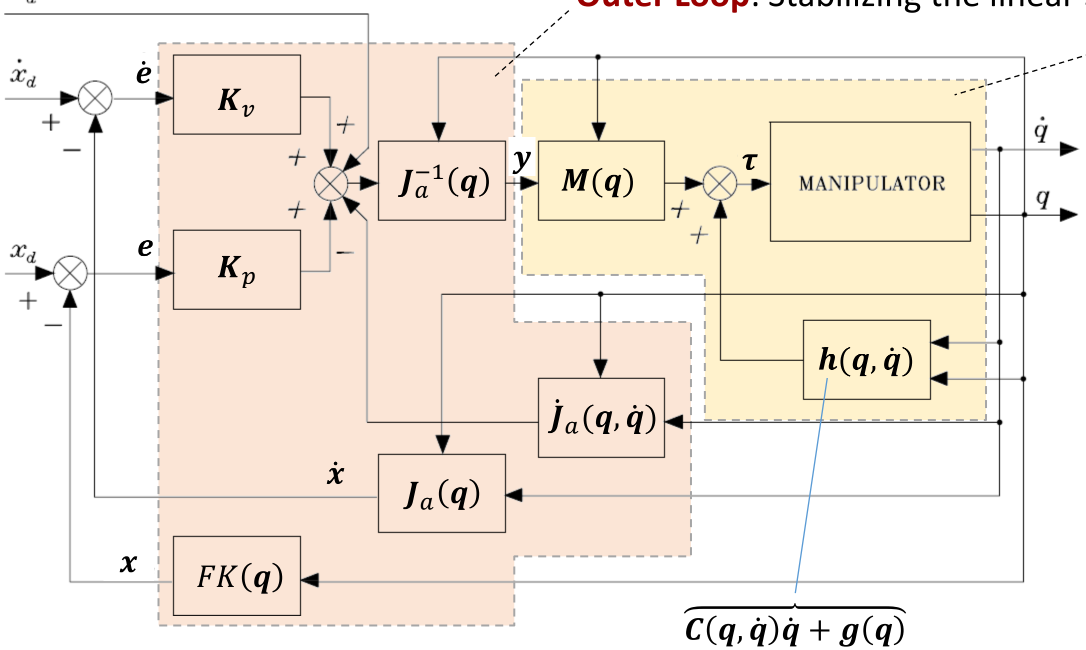
$$\boldsymbol{e} = \boldsymbol{x}_d - \boldsymbol{x}$$

$$\boldsymbol{y} = \boldsymbol{J}_a^{-1}(\boldsymbol{q})\left(\ddot{\boldsymbol{x}}_d + \boldsymbol{K}_v\dot{\boldsymbol{e}} + \boldsymbol{K}_p\boldsymbol{e} - \dot{\boldsymbol{J}}_a(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}\right)$$

$\boldsymbol{K}_p, \boldsymbol{K}_v \in \mathbb{R}^{n\times n}$ are symmetric positive definite matrices.

Outer Loop: Stabilizing the linear system.

Inner Loop: Obtaining a linear and decoupled system.



Note: If measurements of $\boldsymbol{x}$ and $\dot{\boldsymbol{x}}$ are made directly in the task space, $FK(\boldsymbol{q})$ is not required; however, it is necessary to measure $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$ to update $\boldsymbol{J}_a^{-1}$, $\dot{\boldsymbol{J}}_a$, $\boldsymbol{M}$, $\boldsymbol{C}$, and $\boldsymbol{g}$ on-line.

Motion Control: Inverse Dynamics Control

(Based on Robot Dynamics in J-Space & Min. Coord. Rep. of EE Frame) (cont.)

The closed-loop dynamic equation is derived as

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{M}(\boldsymbol{q})\left[\boldsymbol{J}_a^{-1}(\boldsymbol{q})\left(\ddot{\boldsymbol{x}}_d + \boldsymbol{K}_v\dot{\boldsymbol{e}} + \boldsymbol{K}_p\boldsymbol{e} - \dot{\boldsymbol{J}}_a(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}\right)\right] + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q})$$

$$\downarrow$$

$$\ddot{\boldsymbol{q}} = \boldsymbol{J}_a^{-1}(\boldsymbol{q})\left(\ddot{\boldsymbol{x}}_d + \boldsymbol{K}_v\dot{\boldsymbol{e}} + \boldsymbol{K}_p\boldsymbol{e} - \dot{\boldsymbol{J}}_a(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}\right) = \boldsymbol{y}$$

$$\Big\downarrow \; \ddot{\boldsymbol{x}} = \boldsymbol{J}_a(\boldsymbol{q})\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_a(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}$$

$$\ddot{\boldsymbol{e}} + \boldsymbol{K}_v\dot{\boldsymbol{e}} + \boldsymbol{K}_p\boldsymbol{e} = \boldsymbol{0} \longrightarrow \frac{d}{dt}\begin{bmatrix}\boldsymbol{e}\\ \dot{\boldsymbol{e}}\end{bmatrix} = \begin{bmatrix}\dot{\boldsymbol{e}}\\ -\boldsymbol{K}_p\boldsymbol{e} - \boldsymbol{K}_v\dot{\boldsymbol{e}}\end{bmatrix} = \begin{bmatrix}\boldsymbol{0} & \boldsymbol{I}_n\\ -\boldsymbol{K}_p & -\boldsymbol{K}_v\end{bmatrix}\begin{bmatrix}\boldsymbol{e}\\ \dot{\boldsymbol{e}}\end{bmatrix}$$

The system is **decoupled**, **linear** and **autonomous**, and the origin $(\boldsymbol{e},\dot{\boldsymbol{e}}) = \boldsymbol{0} \in \mathbb{R}^{2n}$ is the unique equilibrium point.

Similar to Inverse Dynamics Control in joint space, the origin $(\boldsymbol{e},\dot{\boldsymbol{e}}) = \boldsymbol{0}$ is <u>globally asymptotically (exponentially) stable</u> for any initial condition $\boldsymbol{q}(0), \dot{\boldsymbol{q}}(0) \in \mathbb{R}^n$:

$$\lim_{t\to\infty} \boldsymbol{e}(t) = \boldsymbol{0}$$

A Remark on Computation of Error

$$\boldsymbol{e} = \begin{bmatrix} \boldsymbol{e}_R \\ \boldsymbol{e}_p \end{bmatrix} = \begin{bmatrix} \boldsymbol{e}_R \\ \boldsymbol{p}_d - \boldsymbol{p} \end{bmatrix}, \qquad \dot{\boldsymbol{e}} = \begin{bmatrix} \dot{\boldsymbol{e}}_R \\ \dot{\boldsymbol{e}}_p \end{bmatrix} = \begin{bmatrix} \dot{\boldsymbol{e}}_R \\ \dot{\boldsymbol{p}}_d - \dot{\boldsymbol{p}} \end{bmatrix}$$

Computation of $\boldsymbol{e}_R$, $\dot{\boldsymbol{e}}_R$ depends on the orientation representation of end-effector frame:

(1) Euler Angles: Method 1: $\boldsymbol{e}_R = \boldsymbol{\phi}_d - \boldsymbol{\phi} \longrightarrow \dot{\boldsymbol{e}}_R = \dot{\boldsymbol{\phi}}_d - \dot{\boldsymbol{\phi}}$

Method 2: $\boldsymbol{e}_R = \text{EulerAngles}(\boldsymbol{R}^T \boldsymbol{R}_d)$

$\boldsymbol{\phi} \in \mathbb{R}^3$
$\boldsymbol{R} \in SO(3)$

Assumption: There is no kinematic or representation singularities.

(2) Angle and Axis (Exponential Coordinates):

Method 1: $\boldsymbol{R}_e = \boldsymbol{R}^T \boldsymbol{R}_d$, $\log(\boldsymbol{R}_e) = [\hat{\boldsymbol{\omega}}]\theta$ (in EE frame) ,

$\boldsymbol{e}_R := \hat{\boldsymbol{\omega}}\theta$ (in EE frame)

$\boldsymbol{e}_R := \boldsymbol{R}\hat{\boldsymbol{\omega}}\theta$ (in base frame)

A Remark on Computation of Error

Method 2: $\boldsymbol{R}_e = \boldsymbol{R}^T \boldsymbol{R}_d$, $\quad \mathrm{UnitQuat}(\boldsymbol{R}_e) = \begin{bmatrix} \cos\theta/2 \\ \sin\theta/2\, \widehat{\boldsymbol{\omega}} \end{bmatrix} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}$

(in EE frame)

$\boldsymbol{e}_R \coloneqq \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix}$ (in EE frame)

$\boldsymbol{e}_R \coloneqq \boldsymbol{R} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix}$ (in base frame)

- If $\boldsymbol{\omega}$ and $\boldsymbol{\omega}_d$ are measured/given in the base frame: $\dot{\boldsymbol{e}}_R \coloneqq \boldsymbol{\omega}_d - \boldsymbol{\omega}$

$$\dot{\boldsymbol{e}} = \begin{bmatrix} \dot{\boldsymbol{e}}_R \\ \dot{\boldsymbol{e}}_p \end{bmatrix} = \begin{bmatrix} \boldsymbol{\omega}_d - \boldsymbol{\omega} \\ \dot{\boldsymbol{p}}_d - \dot{\boldsymbol{p}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\omega}_d \\ \dot{\boldsymbol{p}}_d \end{bmatrix} - \begin{bmatrix} \boldsymbol{\omega} \\ \dot{\boldsymbol{p}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\omega}_d \\ \dot{\boldsymbol{p}}_d \end{bmatrix} - \boldsymbol{J}_g \dot{\boldsymbol{q}}$$

Motion Control: Inverse Dynamics Control

(Based on Robot Dynamics in T-Space)

Inverse dynamics control law when end-effector trajectory is specified by $\boldsymbol{T}_d(t) \in SE(3)$, $\boldsymbol{\mathcal{V}}_d(t) \in \mathbb{R}^6$:

$$\boldsymbol{\tau} = \boldsymbol{J}^{\mathrm{T}}(\boldsymbol{q})\big(\boldsymbol{M}_C(\boldsymbol{q})\boldsymbol{y} + \boldsymbol{C}_C(\boldsymbol{\theta}, \boldsymbol{\mathcal{V}})\boldsymbol{\mathcal{V}} + \boldsymbol{g}_C(\boldsymbol{q})\big)$$

PD:

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

Configuration Error

Twist Error

Feedforward acceleration Expressed in the actual EE frame $\boldsymbol{T}$.

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

Expressed in the actual EE frame $\boldsymbol{T}$.

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

Expressed in the actual EE frame $\boldsymbol{T}$.

PID:

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