Ch10: Centralized Control - Motion Control – Part 2

Robust Inverse Dynamics Control

Introduction

All the control technique discussed are based on the assumption of **accurate** (partial or complete) knowledge of dynamic terms ($\boldsymbol{M}(\boldsymbol{q})$, $\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})$ and/or $\boldsymbol{g}(\boldsymbol{q})$) for cancellation of nonlinearities and also real time (or on-line) computation of these terms.

However, in practice, the dynamic model of the robot is usually known with a certain degree of uncertainty (model uncertainty) due to

- imperfect knowledge of parameters (bounded parametric uncertainties), e.g., inertial parameters,
- unknown end-effector payloads,
- existence of unmodeled/neglected dynamic effects.

Moreover, we may simplify/approximate the dynamic terms to make them less demanding for real time computation in milliseconds, e.g., computing only the dominant terms.

The **robust control** and **adaptive control** techniques can be used to cope with these situations.

Robust Inverse Dynamics Control

Consider the dynamic model of an n-DOF open-chain manipulator with no friction at the joints and no external force at the end-effector.

$$\boldsymbol{\tau} = \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) \qquad (1)$$

Let $\widehat{\boldsymbol{M}}(\boldsymbol{q})$, $\widehat{\boldsymbol{C}}(\boldsymbol{q},\dot{\boldsymbol{q}})$, and $\widehat{\boldsymbol{g}}(\boldsymbol{q})$ be the estimated, computed, or nominal values of actual $\boldsymbol{M}(\boldsymbol{q})$, $\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})$, and $\boldsymbol{g}(\boldsymbol{q})$ terms. Thus, the inverse dynamics (or computed torque) control is

$$\boldsymbol{\tau} = \widehat{\boldsymbol{M}}(\boldsymbol{q})\boldsymbol{y} + \widehat{\boldsymbol{C}}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \widehat{\boldsymbol{g}}(\boldsymbol{q}) \qquad (2)$$

The error on the estimates due to imperfect model compensation as well as to intentional simplification in inverse dynamics computation:

$$\widetilde{\boldsymbol{M}}(\boldsymbol{q}) = \widehat{\boldsymbol{M}}(\boldsymbol{q}) - \boldsymbol{M}(\boldsymbol{q})$$
$$\widetilde{\boldsymbol{C}}(\boldsymbol{q}) = \widehat{\boldsymbol{C}}(\boldsymbol{q}) - \boldsymbol{C}(\boldsymbol{q})$$
$$\widetilde{\boldsymbol{g}}(\boldsymbol{q}) = \widehat{\boldsymbol{g}}(\boldsymbol{q}) - \boldsymbol{g}(\boldsymbol{q})$$

Closed-loop Dynamic Equation is $\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \widehat{\boldsymbol{M}}(\boldsymbol{q})\boldsymbol{y} + \widehat{\boldsymbol{C}}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \widehat{\boldsymbol{g}}(\boldsymbol{q})$
(1), (2)

Robust Inverse Dynamics Control

$$\boldsymbol{M}\ddot{\boldsymbol{q}} = \widehat{\boldsymbol{M}}\boldsymbol{y} + \widetilde{\boldsymbol{C}}\dot{\boldsymbol{q}} + \widetilde{\boldsymbol{g}}$$
$$= \widehat{\boldsymbol{M}}\boldsymbol{y} - \boldsymbol{M}\boldsymbol{y} + \boldsymbol{M}\boldsymbol{y} + \widetilde{\boldsymbol{C}}\dot{\boldsymbol{q}} + \widetilde{\boldsymbol{g}}$$
$$= \boldsymbol{M}\boldsymbol{y} + \widetilde{\boldsymbol{M}}\boldsymbol{y} + \widetilde{\boldsymbol{C}}\dot{\boldsymbol{q}} + \widetilde{\boldsymbol{g}}$$

$\Rightarrow$

$$\ddot{\boldsymbol{q}} = \boldsymbol{y} + \boldsymbol{\eta}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{y}) \quad (3)$$

$$\boldsymbol{\eta}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{y}) = \boldsymbol{M}^{-1}\left(\widetilde{\boldsymbol{M}}\boldsymbol{y} + \widetilde{\boldsymbol{C}}\dot{\boldsymbol{q}} + \widetilde{\boldsymbol{g}}\right)$$
$$= \left(\boldsymbol{M}^{-1}\widehat{\boldsymbol{M}} - \boldsymbol{I}_n\right)\boldsymbol{y} + \boldsymbol{M}^{-1}\left(\widetilde{\boldsymbol{C}}\dot{\boldsymbol{q}} + \widetilde{\boldsymbol{g}}\right)$$

$\boldsymbol{\eta} \in \mathbb{R}^n$ is called the **uncertainty**, and in general, a nonlinear function of $\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{y}$.

Note: The system (3) is coupled due to the nonlinear uncertainty $\boldsymbol{\eta}$ and, therefore, there is no guarantee that the outer-loop control

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

will satisfy desired tracking performance specifications: $\ddot{\boldsymbol{e}} + \boldsymbol{K}_v\dot{\boldsymbol{e}} + \boldsymbol{K}_p\boldsymbol{e} = \boldsymbol{\eta}$

By modifying this outer feedback loop using Lyapunov direct method (by introduction of an additional term), we can have a controller which is robust to the uncertainty $\boldsymbol{\eta}$ and can guarantee global convergence of the tracking error $\boldsymbol{e}$:

$$\boldsymbol{y} = \ddot{\boldsymbol{q}}_d(t) + \boldsymbol{K}_v\dot{\boldsymbol{e}} + \boldsymbol{K}_p\boldsymbol{e} + \boldsymbol{\delta y} \quad (4)$$

Robust Inverse Dynamics Control

(3), (4) $\Rightarrow$ $$\ddot{\boldsymbol{e}} + \boldsymbol{K}_v\dot{\boldsymbol{e}} + \boldsymbol{K}_p\boldsymbol{e} + \boldsymbol{\delta y} + \boldsymbol{\eta}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{y}) = \boldsymbol{0}$$

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

$\boldsymbol{\zeta} = \begin{bmatrix} \boldsymbol{e} \\ \dot{\boldsymbol{e}} \end{bmatrix}$

In state space

$$\boxed{\dot{\boldsymbol{\zeta}} = \boldsymbol{A\zeta} + \boldsymbol{B}(-\boldsymbol{\delta y} - \boldsymbol{\eta})}$$

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{0}_n & \boldsymbol{I}_n \\ -\boldsymbol{K}_p & -\boldsymbol{K}_v \end{bmatrix} \in \mathbb{R}^{2n\times 2n}, \boldsymbol{B} = \begin{bmatrix} \boldsymbol{0}_n \\ \boldsymbol{I}_n \end{bmatrix} \in \mathbb{R}^{2n\times n}$$

Note: $\boldsymbol{K}_p$ and $\boldsymbol{K}_v$ are chosen to be positive definite (e.g., $\boldsymbol{K}_p = \mathrm{diag}\{\omega_{n1}^2, \dots, \omega_{nn}^2\}$ and $\boldsymbol{K}_v = \mathrm{diag}\{2\zeta_1\omega_{n1}, \dots, 2\zeta_n\omega_{nn}\}$). Therefore, the matrix $\boldsymbol{A}$ is Hurwitz (i.e., its eigenvalues all have negative real parts).

- The basic idea to design $\boldsymbol{\delta y} \in \mathbb{R}^n$ and guarantee global convergence of $\boldsymbol{\zeta}$ (asymptotic stability of the system) is to assume that, even though the uncertainty $\boldsymbol{\eta}$ is unknown, we can estimate a scalar bound $\rho(\boldsymbol{\zeta}, t) \geq 0$ on its range of variation as

$$\boxed{\|\boldsymbol{\eta}\| \leq \rho(\boldsymbol{\zeta}, t)}$$

(the greater the uncertainty, the greater the ρ)

$$\|\boldsymbol{\eta}\| = \left\|\left(\boldsymbol{M}^{-1}\widehat{\boldsymbol{M}} - \boldsymbol{I}_n\right)\left(\ddot{\boldsymbol{q}}_d(t) + \boldsymbol{K}_v\dot{\boldsymbol{e}} + \boldsymbol{K}_p\boldsymbol{e} + \boldsymbol{\delta y}\right) + \boldsymbol{M}^{-1}\left(\widetilde{\boldsymbol{C}}\dot{\boldsymbol{q}} + \widetilde{\boldsymbol{g}}\right)\right\|$$

$$\leq \left\|\boldsymbol{M}^{-1}\widehat{\boldsymbol{M}} - \boldsymbol{I}_n\right\|\left(\|\ddot{\boldsymbol{q}}_d\| + \left\|\boldsymbol{K}_v\dot{\boldsymbol{e}} + \boldsymbol{K}_p\boldsymbol{e}\right\| + \|\boldsymbol{\delta y}\|\right) + \|\boldsymbol{M}^{-1}\|\left\|\widetilde{\boldsymbol{C}}\dot{\boldsymbol{q}} + \widetilde{\boldsymbol{g}}\right\| \leq \rho(\boldsymbol{\zeta}, t)$$

Assumptions

To simplify this inequality, we make the following assumptions as well:

(1) $\|\boldsymbol{M}^{-1}(\boldsymbol{q})\widehat{\boldsymbol{M}}(\boldsymbol{q}) - \boldsymbol{I}_n\| \leq \alpha < 1 \quad \forall \boldsymbol{q}$

Since $0 < M_m \leq \|\boldsymbol{M}^{-1}(\boldsymbol{q})\| \leq M_M < \infty$ $(\forall \boldsymbol{q})$ a choice for $\widehat{\boldsymbol{M}}(\boldsymbol{q})$ always exists which satisfies (1). For example, by setting

$$\widehat{\boldsymbol{M}} = \frac{2}{M_M + M_m}\boldsymbol{I}_n$$

We can show that $\|\boldsymbol{M}^{-1}\widehat{\boldsymbol{M}} - \boldsymbol{I}\| \leq \dfrac{M_M - M_m}{M_M + M_m} = \alpha < 1.$ (If $\widehat{\boldsymbol{M}} = \boldsymbol{M}$, then $\alpha = 0$)

(2) $\sup_{t \geq 0} \|\ddot{\boldsymbol{q}}_d\| < Q_M < \infty \quad \forall \ddot{\boldsymbol{q}}_d$

(3) $\left\|\widetilde{\boldsymbol{C}}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \widetilde{\boldsymbol{g}}(\boldsymbol{q})\right\| \leq \Phi < \infty \quad \forall \boldsymbol{q}, \dot{\boldsymbol{q}}$

(4) $\|\boldsymbol{\delta y}\| = \rho(\boldsymbol{\zeta}, t)$ (This is required to make $\dot{V}(\boldsymbol{\zeta}) < \boldsymbol{0}$)

Robust Inverse Dynamics Control

$$\|\boldsymbol{\eta}\| \leq \left\|\boldsymbol{M}^{-1}\widehat{\boldsymbol{M}} - \boldsymbol{I}_n\right\|\left(\|\ddot{\boldsymbol{q}}_d\| + \left\|\boldsymbol{K}_v\dot{\boldsymbol{e}} + \boldsymbol{K}_p\boldsymbol{e}\right\| + \|\boldsymbol{\delta y}\|\right) + \|\boldsymbol{M}^{-1}\|\left\|\widetilde{\boldsymbol{C}}\dot{\boldsymbol{q}} + \widetilde{\boldsymbol{g}}\right\| \leq \rho(\boldsymbol{\zeta}, t)$$

$$\leq \alpha Q_M + \alpha\left\|\boldsymbol{K}_v\dot{\boldsymbol{e}} + \boldsymbol{K}_p\boldsymbol{e}\right\| + \alpha\rho(\boldsymbol{\zeta}, t) + M_M\Phi \leq \rho(\boldsymbol{\zeta}, t)$$

$$\Rightarrow \quad \rho(\boldsymbol{\zeta}, t) \geq \frac{1}{1-\alpha}\left(\alpha Q_M + \alpha\left\|\boldsymbol{K}_v\dot{\boldsymbol{e}} + \boldsymbol{K}_p\boldsymbol{e}\right\| + M_M\Phi\right)$$

Using the Lyapunov function candidate $V(\boldsymbol{\zeta}) = \boldsymbol{\zeta}^T\boldsymbol{P}\boldsymbol{\zeta} > 0 \quad \forall\boldsymbol{\zeta} \neq \boldsymbol{0}$ (where $\boldsymbol{P} \in \mathbb{R}^{2n\times 2n}$ is a PD matrix), we can determine $\boldsymbol{\delta y}$ as:

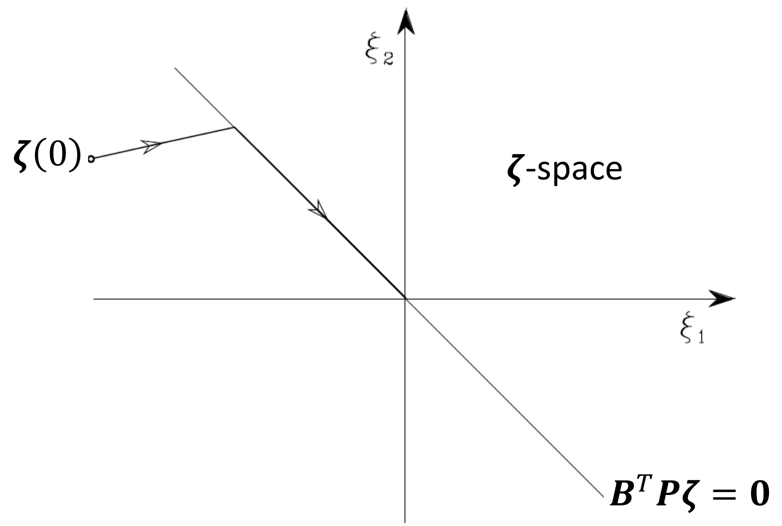
$$\boldsymbol{\delta y} = \begin{cases} \rho\dfrac{\boldsymbol{B}^T\boldsymbol{P}\boldsymbol{\zeta}}{\|\boldsymbol{B}^T\boldsymbol{P}\boldsymbol{\zeta}\|} & \|\boldsymbol{B}^T\boldsymbol{P}\boldsymbol{\zeta}\| \neq 0 \\ \boldsymbol{0}_{n\times 1} & \|\boldsymbol{B}^T\boldsymbol{P}\boldsymbol{\zeta}\| = 0 \end{cases} \qquad \boldsymbol{B}^T\boldsymbol{P}\boldsymbol{\zeta} \in \mathbb{R}^n$$

Since matrix $\boldsymbol{A}$ is Hurwitz, we may choose any $\boldsymbol{Q} > 0$ (e.g., $\boldsymbol{Q} = \boldsymbol{I}_{2n}$) and let $\boldsymbol{P} > 0$ be a unique symmetric PD matrix satisfying the Lyapunov equation:

$$\boldsymbol{A}^{\boldsymbol{T}}\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} = -\boldsymbol{Q}$$

Chattering Phenomenon

The error trajectory $\boldsymbol{\zeta}(t)$ under this robust control reaches (attracts) the *sliding hyperplane* $\boldsymbol{B}^T\boldsymbol{P}\boldsymbol{\zeta} = \mathbf{0} \in \mathbb{R}^n$ from $\boldsymbol{\zeta}(0)$ (where $\boldsymbol{\zeta}(0) \neq 0$ and $\boldsymbol{\zeta} \notin \mathcal{N}(\boldsymbol{B}^T\boldsymbol{P})$) and then, tend (slide) to the origin (zero) with a transient depending on $\boldsymbol{K}_p$, $\boldsymbol{K}_v$, and $\boldsymbol{P}$.



Since this control term $\boldsymbol{\delta y}$ is <u>discontinuous</u> on the subspace defined by $\boldsymbol{B}^T\boldsymbol{P}\boldsymbol{\zeta} = \mathbf{0}$, in practice, this discontinuity results in the **phenomenon of chattering**, where the control switches at a finite frequency <u>between the control values</u> in $\boldsymbol{\delta y}$ and the trajectories oscillate around the sliding subspace with a magnitude as low as the frequency is high.

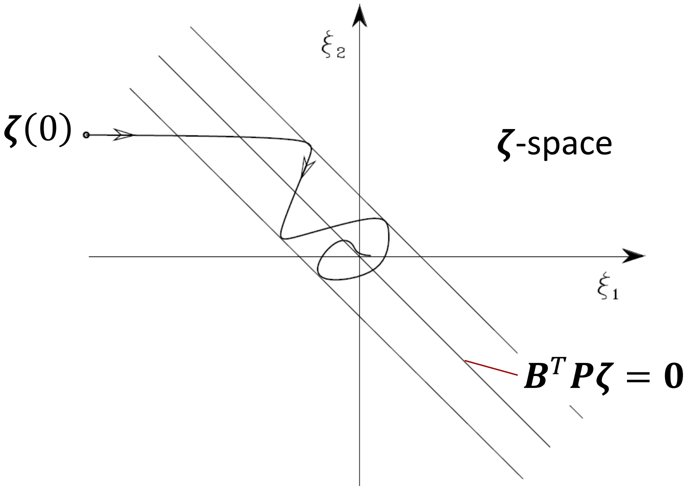
Elimination of Chattering Phenomenon

Elimination of the high-frequency chattering can be achieved by adopting a continuous approximation of the robust control law which, even if it does not guarantee error convergence to zero, ensures bounded-norm errors. By defining a constant $\epsilon \in \mathbb{R}$:

$$\boldsymbol{\delta y} = \begin{cases} \rho \dfrac{\boldsymbol{B}^T\boldsymbol{P\zeta}}{\|\boldsymbol{B}^T\boldsymbol{P\zeta}\|} & \|\boldsymbol{B}^T\boldsymbol{P\zeta}\| > \epsilon \\ \dfrac{\rho}{\epsilon}\boldsymbol{B}^T\boldsymbol{P\zeta} & \|\boldsymbol{B}^T\boldsymbol{P\zeta}\| \le \epsilon \end{cases}$$

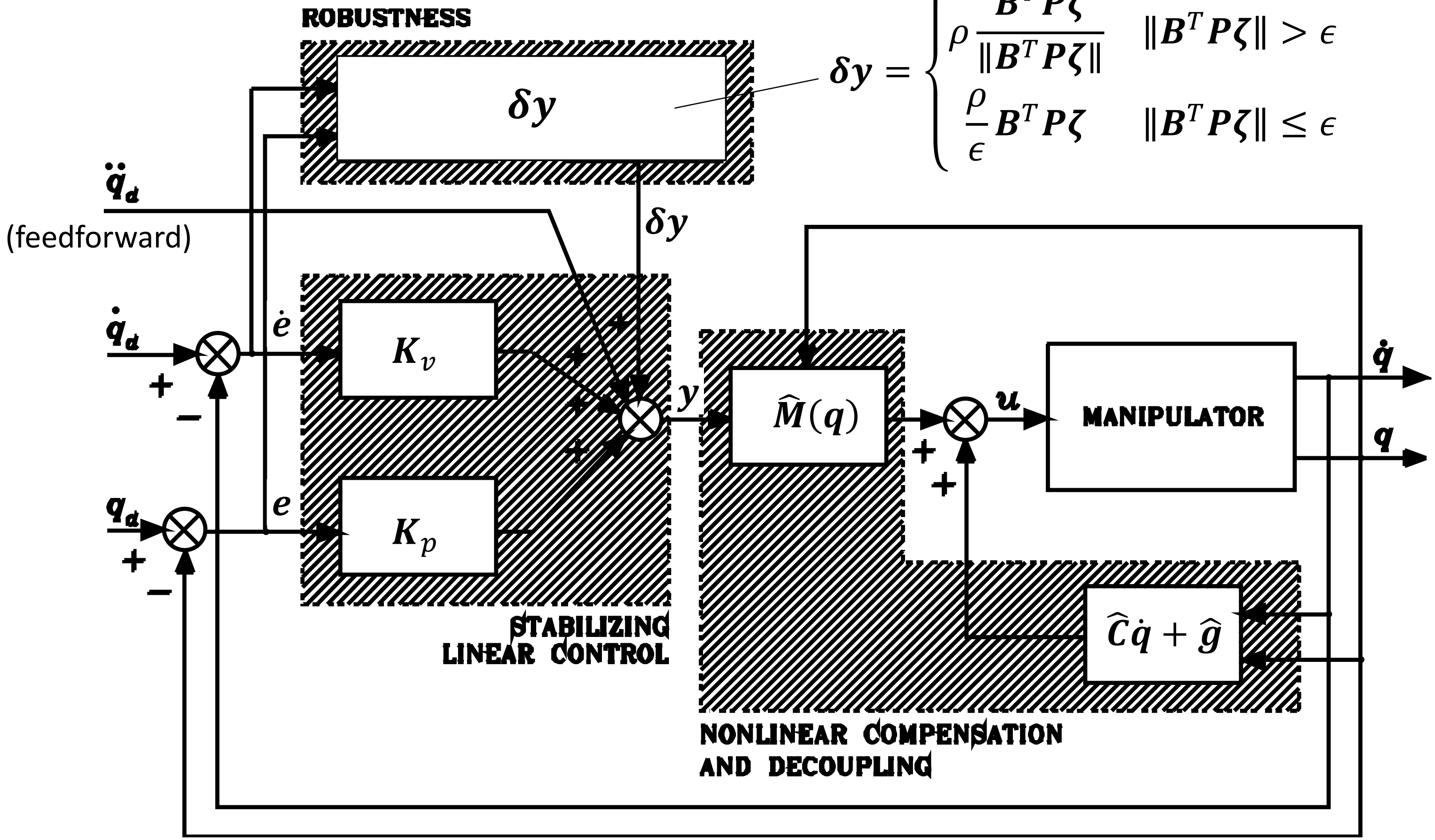
Note: This control signal is now continuous and differentiable (except at $\|\boldsymbol{B}^T\boldsymbol{P\zeta}\|$).

Note: The hyperplane $\boldsymbol{B}^T\boldsymbol{P\zeta} = \boldsymbol{0}$ is no longer attractive, and the error is allowed to vary within a boundary layer whose thickness depends on ϵ.



Block Scheme of Robust Inverse Dynamics Control

$$\delta y = \begin{cases} \rho \dfrac{B^T P \zeta}{\|B^T P \zeta\|} & \|B^T P \zeta\| > \epsilon \\ \dfrac{\rho}{\epsilon} B^T P \zeta & \|B^T P \zeta\| \leq \epsilon \end{cases}$$



Adaptive Inverse Dynamics Control

Adaptive Control

The dynamic models of robot manipulators possess parameters which depend on physical quantities such as the mass of the objects possibly held by the end-effector. This mass is typically unknown, which means that the values of these parameters are unknown. The problem of controlling systems with unknown parameters is the main objective of the adaptive controllers.

These owe their name to the addition of an on-line adaptation law (or estimation scheme) which updates (during execution) an estimate of the unknown parameters to be used in the control law, to those of the true parameters. Therefore, the control law is updated in real time.

Note: For designing the adaptive controllers, we assume that structure of the dynamic equations of the system is known, but there are some unknown constant or slowly-varying parameters. Moreover, we assume there is no external disturbances.

Adaptive Inverse Dynamics Control

Consider the dynamic model of an n-DOF open-chain manipulator with no external force at the end-effector.

$$\boldsymbol{\tau} = \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) \qquad (1)$$

It is always possible to express the nonlinear equations of motion of the robot in a linear form with respect to a suitable set of constant parameters $\boldsymbol{\pi} \in \mathbb{R}^p$ as

$$\boldsymbol{\tau} = \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{Y}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})\boldsymbol{\pi} \qquad \boldsymbol{Y}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}) \in \mathbb{R}^{n\times p}$$

The property of linearity in the parameters is fundamental for deriving **Adaptive Control** laws.

Suppose that the computational robot dynamic model has the <u>same structure as that of the manipulator actual dynamic model (1)</u>, but (some) of its parameters $\boldsymbol{\pi}$ are not known exactly. Therefore,

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

where $\widehat{\boldsymbol{\pi}}$ is the available estimate on the parameters $\boldsymbol{\pi}$, and $\widehat{\boldsymbol{M}}(\boldsymbol{q})$, $\widehat{\boldsymbol{C}}(\boldsymbol{q}, \dot{\boldsymbol{q}})$, and $\widehat{\boldsymbol{g}}(\boldsymbol{q})$ are estimated terms based on $\widehat{\boldsymbol{\pi}}$.

Adaptive Inverse Dynamics Control

The inverse dynamics (or computed torque) control is $\boldsymbol{\tau} = \widehat{\boldsymbol{M}}(\boldsymbol{q})\boldsymbol{y} + \widehat{\boldsymbol{C}}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \widehat{\boldsymbol{g}}(\boldsymbol{q})$ (2)

and the outer-loop control is chosen to be $\boldsymbol{y} = \ddot{\boldsymbol{q}}_d(t) + \boldsymbol{K}_v\dot{\boldsymbol{e}} + \boldsymbol{K}_p\boldsymbol{e}$ $\quad$ $\boldsymbol{e} = \boldsymbol{q}_d - \boldsymbol{q}$

Let's define the error on the estimates as:

$$\widetilde{\boldsymbol{M}}(\boldsymbol{q}) = \widehat{\boldsymbol{M}}(\boldsymbol{q}) - \boldsymbol{M}(\boldsymbol{q})$$
$$\widetilde{\boldsymbol{C}}(\boldsymbol{q}) = \widehat{\boldsymbol{C}}(\boldsymbol{q}) - \boldsymbol{C}(\boldsymbol{q})$$
$$\widetilde{\boldsymbol{g}}(\boldsymbol{q}) = \widehat{\boldsymbol{g}}(\boldsymbol{q}) - \boldsymbol{g}(\boldsymbol{q})$$

Closed-loop Dynamic Equation is $\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \widehat{\boldsymbol{M}}(\boldsymbol{q})\boldsymbol{y} + \widehat{\boldsymbol{C}}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \widehat{\boldsymbol{g}}(\boldsymbol{q})$
(1), (2)

$$\boldsymbol{M}\ddot{\boldsymbol{q}} = \widehat{\boldsymbol{M}}\boldsymbol{y} + \widetilde{\boldsymbol{C}}\dot{\boldsymbol{q}} + \widetilde{\boldsymbol{g}}$$
$$\boldsymbol{M}\ddot{\boldsymbol{q}} - \widehat{\boldsymbol{M}}\ddot{\boldsymbol{q}} + \widehat{\boldsymbol{M}}\ddot{\boldsymbol{q}} = \widehat{\boldsymbol{M}}\boldsymbol{y} + \widetilde{\boldsymbol{C}}\dot{\boldsymbol{q}} + \widetilde{\boldsymbol{g}}$$
$$\widehat{\boldsymbol{M}}\ddot{\boldsymbol{q}} - \widehat{\boldsymbol{M}}\boldsymbol{y} = \widetilde{\boldsymbol{M}}(\boldsymbol{q}) + \widetilde{\boldsymbol{C}}\dot{\boldsymbol{q}} + \widetilde{\boldsymbol{g}}$$

$$\Rightarrow$$

$$\widehat{\boldsymbol{M}}(\ddot{\boldsymbol{q}} - \boldsymbol{y}) = \boldsymbol{Y}(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})\widehat{\boldsymbol{\pi}} - \boldsymbol{Y}(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})\boldsymbol{\pi}$$
$$\widehat{\boldsymbol{M}}(\ddot{\boldsymbol{q}} - \boldsymbol{y}) = \boldsymbol{Y}(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})(\widehat{\boldsymbol{\pi}} - \boldsymbol{\pi})$$
$$\ddot{\boldsymbol{e}} + \boldsymbol{K}_v\dot{\boldsymbol{e}} + \boldsymbol{K}_p\boldsymbol{e} = -\widehat{\boldsymbol{M}}^{-1}\boldsymbol{Y}(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})\widetilde{\boldsymbol{\pi}}$$
$$\widetilde{\boldsymbol{\pi}} = \widehat{\boldsymbol{\pi}} - \boldsymbol{\pi}$$

Adaptive Inverse Dynamics Control

In state space:

$$\boldsymbol{\zeta} = \begin{bmatrix} \boldsymbol{e} \\ \dot{\boldsymbol{e}} \end{bmatrix} \longrightarrow \dot{\boldsymbol{\zeta}} = \boldsymbol{A}\boldsymbol{\zeta} - \boldsymbol{B}\boldsymbol{\Phi}\tilde{\boldsymbol{\pi}}$$

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{0}_n & \boldsymbol{I}_n \\ -\boldsymbol{K}_p & -\boldsymbol{K}_v \end{bmatrix} \in \mathbb{R}^{2n\times 2n}, \boldsymbol{B} = \begin{bmatrix} \boldsymbol{0}_n \\ \boldsymbol{I}_n \end{bmatrix} \in \mathbb{R}^{2n\times n}$$

$$\boldsymbol{\Phi} = \widehat{\boldsymbol{M}}^{-1}\boldsymbol{Y}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}) \in \mathbb{R}^{n\times p}$$

Note: $\boldsymbol{K}_p$ and $\boldsymbol{K}_v$ are chosen to be positive definite (e.g., $\boldsymbol{K}_p = \mathrm{diag}\{\omega_{n1}^2, \dots, \omega_{nn}^2\}$ and $\boldsymbol{K}_v = \mathrm{diag}\{2\zeta_1\omega_{n1}, \dots, 2\zeta_n\omega_{nn}\}$). Therefore, the matrix $\boldsymbol{A}$ is Hurwitz (i.e., its eigenvalues all have negative real parts).

Let $\boldsymbol{P} > 0$ be a unique symmetric PD matrix satisfying the Lyapunov equation for any $\boldsymbol{Q} > 0$ (e.g., $\boldsymbol{Q} = \boldsymbol{I}_{2n}$):

$$\boldsymbol{A}^T\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} = -\boldsymbol{Q}$$

Moreover, let choose a gradient type parameter update law (which is time-varying estimates of the true parameters) for updating $\widehat{\boldsymbol{M}}(\boldsymbol{q})$, $\widehat{\boldsymbol{C}}(\boldsymbol{q}, \dot{\boldsymbol{q}})$, and $\widehat{\boldsymbol{g}}(\boldsymbol{q})$ in the controller as

$$\dot{\hat{\boldsymbol{\pi}}} = \boldsymbol{\Gamma}^{-1}\boldsymbol{\Phi}^T\boldsymbol{B}^T\boldsymbol{P}\boldsymbol{\zeta}$$

where $\boldsymbol{\Gamma} \in \mathbb{R}^{p\times p}$ is a constant, symmetric, PD matrix (which determines the convergence rate of parameters to their asymptotic values).

Adaptive Inverse Dynamics Control

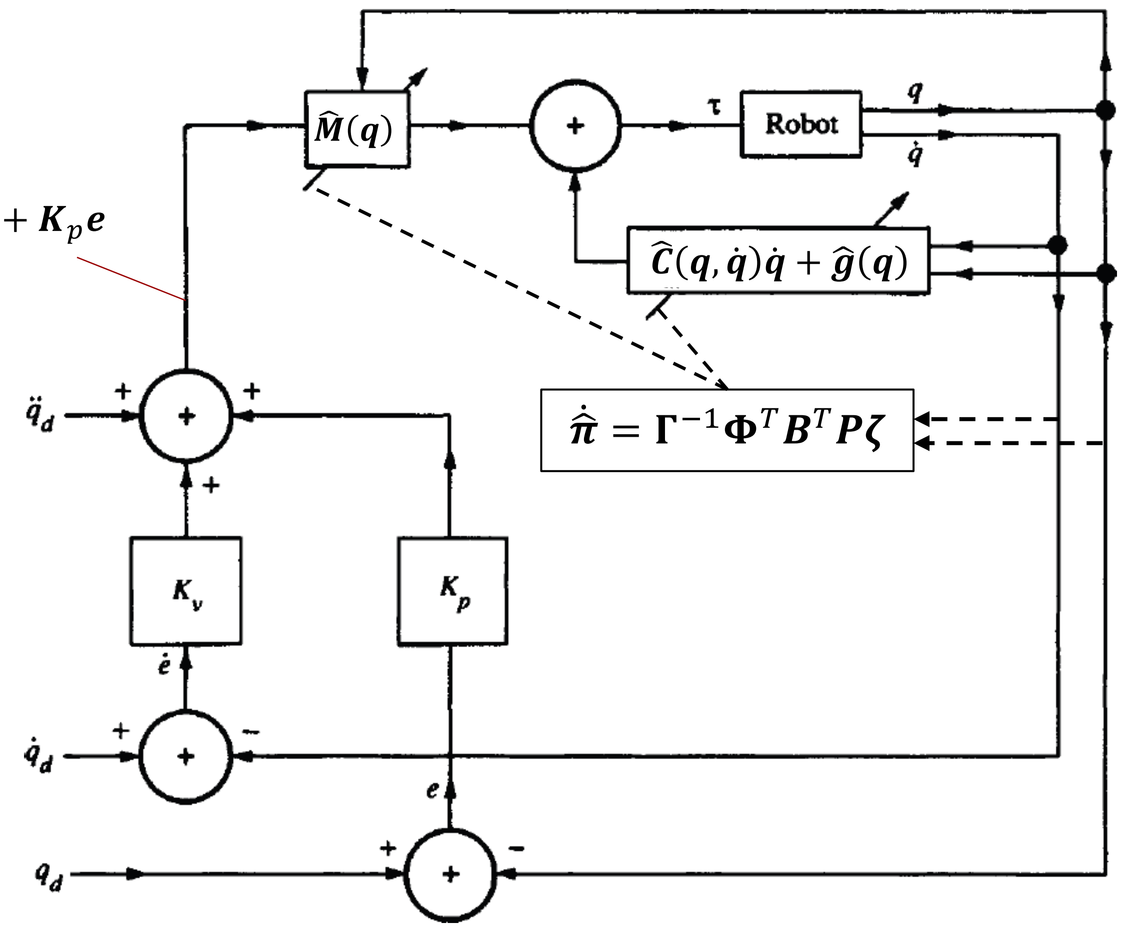
Then, global convergence to zero of the tracking error $\boldsymbol{\zeta}$ with all internal signals remaining bounded can be shown using the Lyapunov function $V(\boldsymbol{\zeta},\tilde{\boldsymbol{\pi}}) = \boldsymbol{\zeta}^T\boldsymbol{P}\boldsymbol{\zeta} + \tilde{\boldsymbol{\pi}}^T\boldsymbol{\Gamma}\tilde{\boldsymbol{\pi}}$:

$$\begin{aligned}
\dot{V} &= \dot{\boldsymbol{\zeta}}^T\boldsymbol{P}\boldsymbol{\zeta} + \boldsymbol{\zeta}^T\boldsymbol{P}\dot{\boldsymbol{\zeta}} + 2\tilde{\boldsymbol{\pi}}^T\Gamma\dot{\tilde{\boldsymbol{\pi}}} \\
&= (\boldsymbol{A}\boldsymbol{\zeta} - \boldsymbol{B}\boldsymbol{\Phi}\tilde{\boldsymbol{\pi}})^T\boldsymbol{P}\boldsymbol{\zeta} + \boldsymbol{\zeta}^T\boldsymbol{P}(\boldsymbol{A}\boldsymbol{\zeta} - \boldsymbol{B}\boldsymbol{\Phi}\tilde{\boldsymbol{\pi}}) + 2\tilde{\boldsymbol{\pi}}^T\boldsymbol{\Gamma}\dot{\tilde{\boldsymbol{\pi}}} \\
&= \boldsymbol{\zeta}^T(\boldsymbol{A}^T\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A})\boldsymbol{\zeta} + 2\tilde{\boldsymbol{\pi}}^T\left(-\boldsymbol{\Phi}^T\boldsymbol{B}^T\boldsymbol{P}\boldsymbol{\zeta} + \boldsymbol{\Gamma}\dot{\tilde{\boldsymbol{\pi}}}\right) \\
&= -\boldsymbol{\zeta}^T\boldsymbol{Q}\boldsymbol{\zeta} + 2\tilde{\boldsymbol{\pi}}^T\left(-\boldsymbol{\Phi}^T\boldsymbol{B}^T\boldsymbol{P}\boldsymbol{\zeta} + \boldsymbol{\Gamma}\dot{\tilde{\boldsymbol{\pi}}}\right)
\end{aligned}
\quad \xrightarrow[\dot{\tilde{\boldsymbol{\pi}}} = \dot{\hat{\boldsymbol{\pi}}}]{\dot{\hat{\boldsymbol{\pi}}} = \boldsymbol{\Gamma}^{-1}\boldsymbol{\Phi}^T\boldsymbol{B}^T\boldsymbol{P}\boldsymbol{\zeta}} \quad \dot{V}(\boldsymbol{\zeta},\tilde{\boldsymbol{\pi}}) = -\boldsymbol{\zeta}^T\boldsymbol{Q}\boldsymbol{\zeta} \leq 0$$

- ❖ Therefore, the tracking error $\boldsymbol{\zeta}$ converge to zero asymptotically and the parameter estimation errors $\tilde{\boldsymbol{\pi}}$ remain bounded. Thus, there is no guarantee that the estimated parameters $\hat{\boldsymbol{\pi}}$ converges to their true values $\boldsymbol{\pi}$ ($\rightarrow$ *parameter drift*).
- ❖ Implementation Issues:
  - The acceleration $\ddot{\boldsymbol{q}}$ is needed in the parameter update law. Acceleration sensors are noisy and introduce additional cost whereas calculating the acceleration by numerical differentiation of position or velocity signals is not feasible in most cases.
  - $\widehat{\boldsymbol{M}}$ must be invertible.
- ❑ The **Passivity-Based** approaches remove both of these issues.

Block Scheme of Adaptive Inverse Dynamics Control

$y = \ddot{q}_d(t) + K_v\dot{e} + K_p e$



Robust Control vs Adaptive Control

- Adaptive controllers perform well in the face of parametric uncertainty. However, a performance degradation is expected whenever unmodelled dynamic effects, e.g., when a reduced computational model is used, or external disturbances occur. On the other hand, robust control techniques provide a natural rejection to external disturbances.

- In a repetitive motion task, the tracking errors produced by a fixed robust controller would tend to be repetitive as well, whereas tracking errors produced by an adaptive controller might be expected to decrease over time as the plant and/or control parameters are updated based on runtime information.

Passivity-Based Control

Passivity-Based Control

The inverse dynamics methods rely, in the ideal case, on exact cancellation of all nonlinearities in the system dynamics. However, the control techniques based on the **passivity** or **skew symmetry property** of the dynamic equations do not rely on complete cancellation of nonlinearities and hence, do not lead to a linear closed-loop system even in the known-parameter case.

Consider the dynamic model of an n-DOF open-chain manipulator with no external force at the end-effector.

$$\boldsymbol{\tau} = \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) \qquad (1)$$

We choose the control input as

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

$\boldsymbol{K}_p, \boldsymbol{K}_v \in \mathbb{R}^{n\times n}$ are symmetric positive definite matrices and $\boldsymbol{\Lambda} = \boldsymbol{K}_v^{-1}\boldsymbol{K}_p$ (which is also a PD Matrix).

Passivity-Based Control

The closed-loop dynamic equation is derived as

$$\boldsymbol{M}(\boldsymbol{q})[\ddot{\boldsymbol{e}} + \boldsymbol{\Lambda}\dot{\boldsymbol{e}}] + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})[\dot{\boldsymbol{e}} + \boldsymbol{\Lambda}\boldsymbol{e}] = -\boldsymbol{K}_p\boldsymbol{e} - \boldsymbol{K}_v\dot{\boldsymbol{e}} = -\boldsymbol{K}_v(\dot{\boldsymbol{e}} + \boldsymbol{\Lambda}\boldsymbol{e})$$

By defining $\boldsymbol{r} = \dot{\boldsymbol{e}} + \boldsymbol{\Lambda}\boldsymbol{e}$:

$$\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{r}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\boldsymbol{r} + \boldsymbol{K}_v\boldsymbol{r} = \boldsymbol{0}$$

Consider the Lyapunov function candidate $V(\boldsymbol{e}, \dot{\boldsymbol{e}}, t) = \frac{1}{2}\boldsymbol{r}^T\boldsymbol{M}(\boldsymbol{q})\boldsymbol{r} + \boldsymbol{e}^T\boldsymbol{K}_p\boldsymbol{e}$

$$\dot{V} = \boldsymbol{r}^T\boldsymbol{M}\dot{\boldsymbol{r}} + \frac{1}{2}\boldsymbol{r}^T\dot{\boldsymbol{M}}\boldsymbol{r} + 2\boldsymbol{e}^T\boldsymbol{K}_p\dot{\boldsymbol{e}} = -\boldsymbol{r}^T\boldsymbol{K}_v\boldsymbol{r} + 2\boldsymbol{e}^T\boldsymbol{K}_p\dot{\boldsymbol{e}} + \frac{1}{2}\boldsymbol{r}^T(\dot{\boldsymbol{M}} - 2\boldsymbol{C})\boldsymbol{r}$$

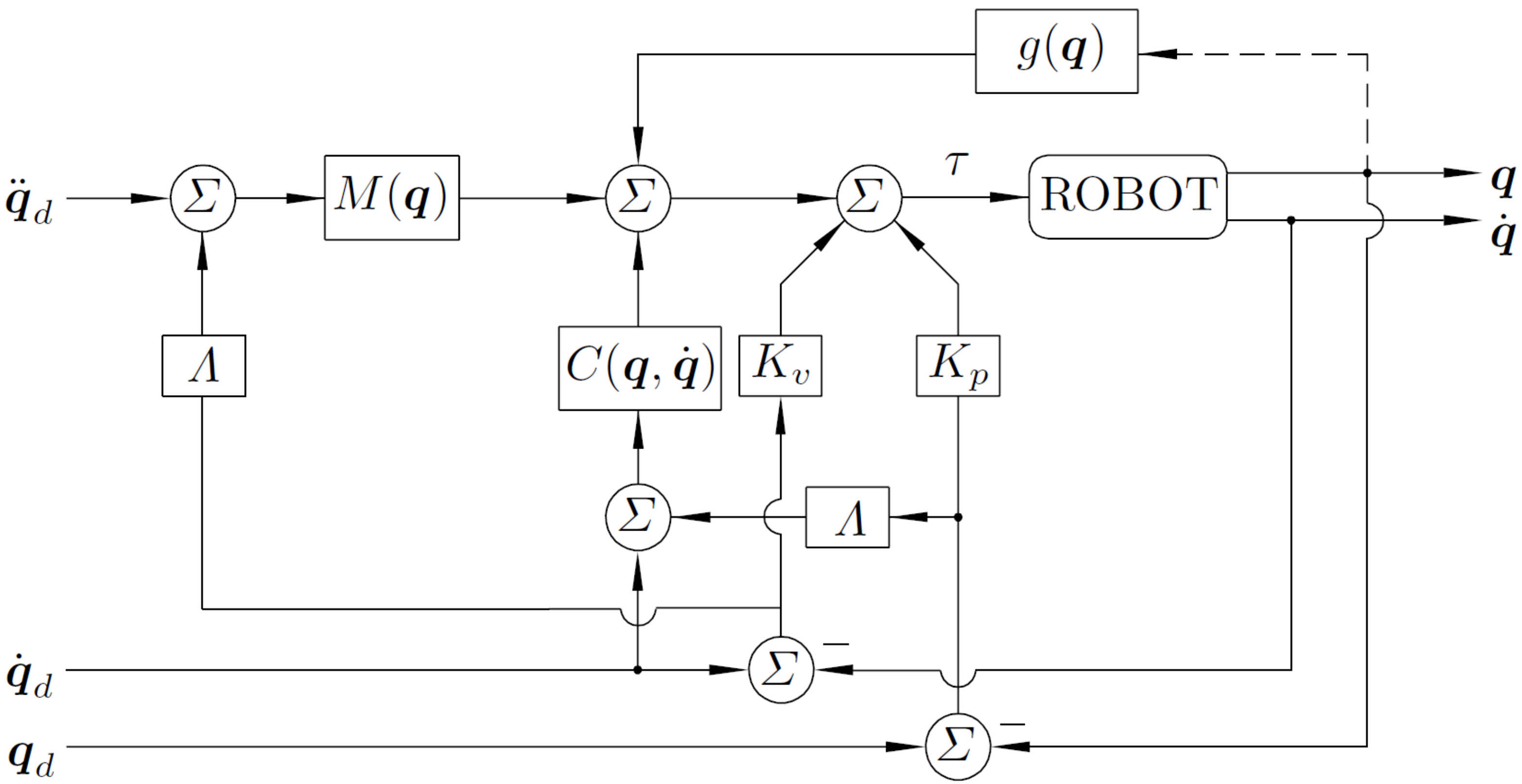
Using the skew symmetry property and the definition of $\boldsymbol{r}$:

$$\begin{aligned}\dot{V} &= -\boldsymbol{e}^T\boldsymbol{\Lambda}^T\boldsymbol{K}_v\boldsymbol{\Lambda}\boldsymbol{e} - \dot{\boldsymbol{e}}^T\boldsymbol{K}_v\dot{\boldsymbol{e}} \\ &= -\boldsymbol{\zeta}^T\boldsymbol{Q}\boldsymbol{\zeta} < 0\end{aligned} \qquad \boldsymbol{Q} = \begin{bmatrix}\boldsymbol{\Lambda}^T\boldsymbol{K}_v\boldsymbol{\Lambda} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{K}_v\end{bmatrix} \quad \boldsymbol{\zeta} = \begin{bmatrix}\boldsymbol{e} \\ \dot{\boldsymbol{e}}\end{bmatrix}$$

Therefore, the tracking error $\boldsymbol{e}$ is globally asymptotically stable.

Note: We concluded the same result for the inverse dynamics control in a much simpler way. However, the real advantage of the passivity-based approach occurs for the robust and adaptive control problems.

Block Scheme of Passivity-Based Control



Passivity-Based Robust Control

Let's use the passivity-based approach to derive an alternative robust control algorithm that exploits both the skew symmetry property and linearity in the parameters and leads to a much easier design in terms of computation of uncertainty bounds.

We choose the control input as

$$\boldsymbol{\tau} = \boldsymbol{K}_p\boldsymbol{e} + \boldsymbol{K}_v\dot{\boldsymbol{e}} + \widehat{\boldsymbol{M}}(\boldsymbol{q})[\ddot{\boldsymbol{q}}_d + \boldsymbol{\Lambda}\dot{\boldsymbol{e}}] + \widehat{\boldsymbol{C}}(\boldsymbol{q},\dot{\boldsymbol{q}})[\dot{\boldsymbol{q}}_d + \boldsymbol{\Lambda}\boldsymbol{e}] + \widehat{\boldsymbol{g}}(\boldsymbol{q}) \quad (1) \qquad \boldsymbol{e} = \boldsymbol{q}_d - \boldsymbol{q}$$

where $\widehat{\boldsymbol{M}}(\boldsymbol{q})$, $\widehat{\boldsymbol{C}}(\boldsymbol{q},\dot{\boldsymbol{q}})$, and $\widehat{\boldsymbol{g}}(\boldsymbol{q})$ are the estimated, computed, or nominal values of actual $\boldsymbol{M}(\boldsymbol{q})$, $\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})$, and $\boldsymbol{g}(\boldsymbol{q})$ terms, $\boldsymbol{K}_p, \boldsymbol{K}_v \in \mathbb{R}^{n\times n}$ are symmetric positive definite matrices, and $\boldsymbol{\Lambda} = \boldsymbol{K}_v^{-1}\boldsymbol{K}_p$. If there is <u>only parametric uncertainty</u>, (1) can be rewritten as

$$\boldsymbol{\tau} = \boldsymbol{K}_p\boldsymbol{e} + \boldsymbol{K}_v\dot{\boldsymbol{e}} + \boldsymbol{Y}\big(\boldsymbol{q},\dot{\boldsymbol{q}},\boldsymbol{q}_d,\dot{\boldsymbol{q}}_d,\ddot{\boldsymbol{q}}_d,\boldsymbol{K}_p,\boldsymbol{K}_v\big)\widehat{\boldsymbol{\pi}} \quad (2)$$

By combining (2) with actual robot dynamic and using (4):

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{K}_p\boldsymbol{e} + \boldsymbol{K}_v\dot{\boldsymbol{e}} + \boldsymbol{Y}\widehat{\boldsymbol{\pi}} \quad (3)$$

$$\boldsymbol{M}(\boldsymbol{q})[\ddot{\boldsymbol{q}}_d + \boldsymbol{\Lambda}\dot{\boldsymbol{e}}] + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})[\dot{\boldsymbol{q}}_d + \boldsymbol{\Lambda}\boldsymbol{e}] + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{Y}\boldsymbol{\pi} \quad (4)$$

(in the absence of uncertainty)

(4)-(3):

$$\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{r}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\boldsymbol{r} + \boldsymbol{K}_v\boldsymbol{r} = \boldsymbol{Y}(\boldsymbol{\pi} - \widehat{\boldsymbol{\pi}}) \quad (5)$$

$$(\boldsymbol{r} = \dot{\boldsymbol{e}} + \boldsymbol{\Lambda}\boldsymbol{e})$$

Passivity-Based Robust Control

Let's choose the term $\hat{\boldsymbol{\pi}}$ as $\hat{\boldsymbol{\pi}} = \boldsymbol{\pi}_0 + \delta\boldsymbol{\pi}$

where $\boldsymbol{\pi}_0$ is a fixed nominal parameter vector and $\delta\boldsymbol{\pi}$ is an additional control term.

Then, (5) can be written as $\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{r}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\boldsymbol{r} + \boldsymbol{K}_v\boldsymbol{r} = \boldsymbol{Y}(\boldsymbol{\pi}' - \delta\boldsymbol{\pi})$

where $\boldsymbol{\pi}' = \boldsymbol{\pi} - \boldsymbol{\pi}_0$ is a <u>constant</u> vector and represents the parametric uncertainty in the system.

If $\boldsymbol{\pi}'$ can be bounded by finding a nonnegative constant $\rho \geq 0$ such that

$$\|\boldsymbol{\pi}'\| = \|\boldsymbol{\pi} - \boldsymbol{\pi}_0\| \leq \rho$$

$\delta\boldsymbol{\pi}$ can be designed according to

$$\delta\boldsymbol{\pi} = \begin{cases} \rho \dfrac{\boldsymbol{Y}^T\boldsymbol{r}}{\|\boldsymbol{Y}^T\boldsymbol{r}\|} & \|\boldsymbol{Y}^T\boldsymbol{r}\| > \epsilon \\ \dfrac{\rho}{\epsilon}\boldsymbol{Y}^T\boldsymbol{r} & \|\boldsymbol{Y}^T\boldsymbol{r}\| \leq \epsilon \end{cases}$$

where $\epsilon \in \mathbb{R}$ is a constant.

Passivity-Based Robust Control

Using the Lyapunov function candidate $V(\boldsymbol{e}, \dot{\boldsymbol{e}}, t) = \dfrac{1}{2}\boldsymbol{r}^T \boldsymbol{M}(\boldsymbol{q})\boldsymbol{r} + \boldsymbol{e}^T \boldsymbol{K}_p \boldsymbol{e}$

we can show uniform ultimate boundedness of the tracking error.

$$\dot{V} = -\boldsymbol{\zeta}^T \boldsymbol{Q}\boldsymbol{\zeta} + \boldsymbol{r}^T \boldsymbol{Y}(\boldsymbol{\pi}' - \delta\boldsymbol{\pi}) \qquad \boldsymbol{Q} = \begin{bmatrix} \boldsymbol{\Lambda}^T \boldsymbol{K}_v \boldsymbol{\Lambda} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{K}_v \end{bmatrix} \quad \boldsymbol{\zeta} = \begin{bmatrix} \boldsymbol{e} \\ \dot{\boldsymbol{e}} \end{bmatrix}$$

(Proof has been skipped.)

Note: Comparing Passivity-Based Robust Control with Robust Inverse Dynamics Control, we see that finding a constant bound ρ for the constant vector $\boldsymbol{\pi}' = \boldsymbol{\pi} - \boldsymbol{\pi}_0$ (which depends only on the inertia parameters of the manipulator) is much simpler than finding a time-varying bound $\rho(\boldsymbol{\zeta}, t)$ for $\boldsymbol{\eta}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{y}) = \boldsymbol{M}^{-1}\left(\widetilde{\boldsymbol{M}}\boldsymbol{y} + \widetilde{\boldsymbol{C}}\dot{\boldsymbol{q}} + \widetilde{\boldsymbol{g}}\right)$ (which depends on the manipulator state vector, the reference trajectory and, in addition, requires some assumptions like $\|\boldsymbol{M}^{-1}(\boldsymbol{q})\widehat{\boldsymbol{M}}(\boldsymbol{q}) - \boldsymbol{I}_n\| < 1$).

Passivity-Based Adaptive Control

In the adaptive approach, the vector $\hat{\boldsymbol{\pi}}$ (the available estimate on the parameters) in the following equation is taken to be a time-varying estimate of the true parameter vector $\boldsymbol{\pi}$.

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

$$\Rightarrow \quad \boldsymbol{\tau} = \boldsymbol{K}_p\boldsymbol{e} + \boldsymbol{K}_v\dot{\boldsymbol{e}} + \boldsymbol{Y}\left(\boldsymbol{q},\dot{\boldsymbol{q}},\boldsymbol{q}_d,\dot{\boldsymbol{q}}_d,\ddot{\boldsymbol{q}}_d,\boldsymbol{K}_p,\boldsymbol{K}_v\right)\hat{\boldsymbol{\pi}} \qquad (1)$$

By combining (1) with actual robot dynamic:

$$\boldsymbol{r} = \dot{\boldsymbol{e}} + \boldsymbol{\Lambda}\boldsymbol{e}$$
$$\tilde{\boldsymbol{\pi}} = \hat{\boldsymbol{\pi}} - \boldsymbol{\pi}$$
$$\boldsymbol{e} = \boldsymbol{q}_d - \boldsymbol{q}$$

$$\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{r}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\boldsymbol{r} + \boldsymbol{K}_v\boldsymbol{r} = -\boldsymbol{Y}\tilde{\boldsymbol{\pi}} \qquad (2)$$

Let's choose the gradient type update law as

$$\dot{\hat{\boldsymbol{\pi}}} = \boldsymbol{\Gamma}^{-1}\boldsymbol{Y}^T(\boldsymbol{q},\dot{\boldsymbol{q}},\boldsymbol{q}_d,\dot{\boldsymbol{q}}_d,\ddot{\boldsymbol{q}}_d,\boldsymbol{K}_p,\boldsymbol{K}_v)\boldsymbol{r}$$

where $\boldsymbol{\Gamma}$ is a constant, symmetric, PD matrix (which determines the convergence rate of parameters to their asymptotic values).

Consider a Lyapunov function as $\quad V(\boldsymbol{e},\dot{\boldsymbol{e}},\tilde{\boldsymbol{\pi}},t) = \frac{1}{2}\boldsymbol{r}^T\boldsymbol{M}(\boldsymbol{q})\boldsymbol{r} + \boldsymbol{e}^T\boldsymbol{K}_p\boldsymbol{e} + \frac{1}{2}\tilde{\boldsymbol{\pi}}^T\boldsymbol{\Gamma}\tilde{\boldsymbol{\pi}}$

Using (2):

$$\begin{aligned}\dot{V} &= -\boldsymbol{e}^T\boldsymbol{\Lambda}^T\boldsymbol{K}\boldsymbol{\Lambda}\boldsymbol{e} - \dot{\boldsymbol{e}}^T\boldsymbol{K}\dot{\boldsymbol{e}} + \tilde{\boldsymbol{\pi}}^T\left\{\boldsymbol{\Gamma}\dot{\hat{\boldsymbol{\pi}}} + \boldsymbol{Y}^T\boldsymbol{r}\right\} \\ &= -\boldsymbol{e}^T\boldsymbol{\Lambda}^T\boldsymbol{K}\boldsymbol{\Lambda}\boldsymbol{e} - \dot{\boldsymbol{e}}^T\boldsymbol{K}\dot{\boldsymbol{e}} = -\boldsymbol{\zeta}^T\boldsymbol{Q}\boldsymbol{\zeta} \leq 0\end{aligned}$$

$$\boldsymbol{Q} = \begin{bmatrix}\boldsymbol{\Lambda}^T\boldsymbol{K}_v\boldsymbol{\Lambda} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{K}_v\end{bmatrix} \quad \boldsymbol{\zeta} = \begin{bmatrix}\boldsymbol{e} \\ \dot{\boldsymbol{e}}\end{bmatrix}$$

Passivity-Based Adaptive Control

This shows that the closed-loop system is stable in the sense of Lyapunov. It follows that the tracking error $\boldsymbol{\zeta}(t)$ converges globally asymptotically to zero and the parameter estimation error $\tilde{\boldsymbol{\pi}}$ is bounded.

Note: This equation

$$\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{r}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\boldsymbol{r} + \boldsymbol{K}_v\boldsymbol{r} = -\boldsymbol{Y}\tilde{\boldsymbol{\pi}} = -\boldsymbol{Y}(\hat{\boldsymbol{\pi}} - \boldsymbol{\pi}) \underset{}{\overset{\text{as } \boldsymbol{\zeta}(t) \to \boldsymbol{0}}{=}} \boldsymbol{0}$$

does not imply that $\hat{\boldsymbol{\pi}}$ tends to $\boldsymbol{\pi}$; indeed, convergence of parameters to their true values depends on the structure of the matrix $\boldsymbol{Y}$, the desired trajectories $\dot{\boldsymbol{q}}_d, \ddot{\boldsymbol{q}}_d$, and actual trajectories $\boldsymbol{q}, \dot{\boldsymbol{q}}$.

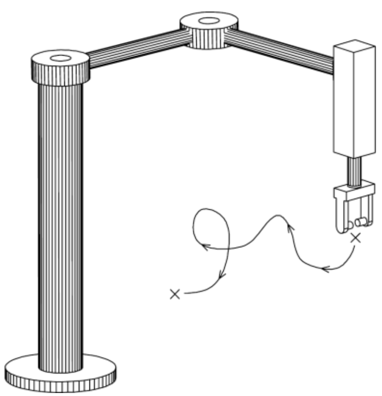
Note: Comparing Passivity-Based Adaptive Control with Adaptive Inverse Dynamics Control, we see that the requirements on acceleration measurement $\ddot{\boldsymbol{q}}$ and boundedness/ invertibility of the estimated inertia matrix $\widehat{\boldsymbol{M}}$ are eliminated.

Torque Optimization

Actuator Saturation

In this section, we consider the problem of **actuator saturation** in the design of nonlinear control laws. The actuator saturation may be considered in two steps:

- As part of the **motion planning problem**, for example, by including bounds on the velocities $\dot{\boldsymbol{q}}_d$ and accelerations $\ddot{\boldsymbol{q}}_d$ when planning desired trajectories and then correlating these bounds with bounds on the actuator torques $\boldsymbol{\tau}$.
- As part of designing a nonlinear control law subject to actuator saturation as a **constrained torque optimization problem**.
  - **Method 1**: Designing a controller using any method discussed before, then, using a torque optimization method on top of that for considering input constraints.
  - **Method 2**: Designing an optimization-based controller like MPC (Model Predictive Controller) for considering input constraints.



Torque Optimization

Given a nominal control law designed as a first step without considering **input constraints**, the goal is to find a control law "closest" to the nominal control with considering input constraints. This can be formulated as a **torque optimization problem**.

Input constraints:

If we assume the dynamic equations for an n-link manipulator as

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

the bounds on the actuator torques $\boldsymbol{\tau}$ can be written as

$$\tau_{i,\min} \leq \tau_i \leq \tau_{i,\max}, \qquad i = 1, \dots, n$$

For simplicity we will assume that these bounds are **constant**.

These inequality constraints can be also written as $\boldsymbol{N\tau} \leq \boldsymbol{c}$

$$\boldsymbol{N} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ -1 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & 1 \\ \vdots & \vdots & & -1 \end{bmatrix} \in \mathbb{R}^{2n \times n} \qquad \boldsymbol{c} = \begin{bmatrix} \tau_{1,\max} \\ -\tau_{1,\min} \\ \vdots \\ \tau_{n,\max} \\ -\tau_{n,\min} \end{bmatrix} \in \mathbb{R}^{2n}$$

Torque Optimization Problem

Let $\boldsymbol{\tau}_0$ be a control law designed by any method (e.g., inverse dynamics control, robust control, adaptive control, etc.). The torque optimization problem is then to choose the actual control input $\boldsymbol{\tau}$ to satisfy

$$\begin{aligned} &\min_{\boldsymbol{\tau}} \frac{1}{2}(\boldsymbol{\tau}-\boldsymbol{\tau}_0)^T \boldsymbol{\Pi}(\boldsymbol{\tau}-\boldsymbol{\tau}_0) \\ &\text{subject to } \boldsymbol{N\tau} \leq \boldsymbol{c} \end{aligned} \qquad (1)$$

Where $\boldsymbol{\Pi} \in \mathbb{R}^{n\times n}$ is a symmetric, positive definite (not necessarily constant) matrix to be chosen by the designer.

This optimal control (minimization) is a **quadratic programming problem** that is solved pointwise, i.e., at each time step.

Solving Optimization Problem Using Primal-Dual Method

The optimization problem (1) is called **Primal Problem** and can be shown to be equivalent to the following optimization problem (2):

$$\max_{\boldsymbol{\lambda}\geq\mathbf{0}} \min_{\boldsymbol{\tau}} \underbrace{\left\{\frac{1}{2}(\boldsymbol{\tau}-\boldsymbol{\tau}_0)^T\boldsymbol{\Pi}(\boldsymbol{\tau}-\boldsymbol{\tau}_0)+\boldsymbol{\lambda}^T(\boldsymbol{N}\boldsymbol{\tau}-\boldsymbol{c})\right\}}_{\boldsymbol{f}(\boldsymbol{\tau})} \quad (2)$$

where $\boldsymbol{\lambda}\in\mathbb{R}^{2n}$ is **Lagrange Multipliers**.

$\min_{\boldsymbol{\tau}} \boldsymbol{f}(\boldsymbol{\tau})$ is an unconstrained minimization and the solution is

$$\frac{d\boldsymbol{f}(\boldsymbol{\tau})}{d\boldsymbol{\tau}}=\mathbf{0} \ \rightarrow (\boldsymbol{\tau}-\boldsymbol{\tau}_0)^T\boldsymbol{\Pi}+\boldsymbol{\lambda}^T\boldsymbol{N}=\mathbf{0} \ \rightarrow \boldsymbol{\tau}^*=\boldsymbol{\tau}_0-\boldsymbol{\Pi}^{-1}\boldsymbol{N}^T\boldsymbol{\lambda}$$

$$\boldsymbol{f}(\boldsymbol{\tau}^*)=-\frac{1}{2}\boldsymbol{\lambda}^T(\boldsymbol{N}\boldsymbol{\Pi}^{-1}\boldsymbol{N}^T)\boldsymbol{\lambda}-\boldsymbol{\lambda}^T(\boldsymbol{c}-\boldsymbol{N}\boldsymbol{\tau}_0)$$

$$\max_{\boldsymbol{\lambda}\geq\mathbf{0}}\left\{-\frac{1}{2}\boldsymbol{\lambda}^T(\boldsymbol{N}\boldsymbol{\Pi}^{-1}\boldsymbol{N}^T)\boldsymbol{\lambda}-\boldsymbol{\lambda}^T(\boldsymbol{c}-\boldsymbol{N}\boldsymbol{\tau}_0)\right\}$$

$$\min_{\boldsymbol{\lambda}\geq\mathbf{0}}\left\{\frac{1}{2}\boldsymbol{\lambda}^T(\boldsymbol{N}\boldsymbol{\Pi}^{-1}\boldsymbol{N}^T)\boldsymbol{\lambda}+\boldsymbol{\lambda}^T(\boldsymbol{c}-\boldsymbol{N}\boldsymbol{\tau}_0)\right\} \quad (3)$$

Dual Problem which is also a quadratic problem.

Solving Optimization Problem Using Primal-Dual Method

After solving the dual problem (3) for $\boldsymbol{\lambda} = \boldsymbol{\lambda}^*$, the optimal control input $\boldsymbol{\tau}$ is given as

$$\boxed{\boldsymbol{\tau}^{**} = \boldsymbol{\tau}_0 - \boldsymbol{\Pi}^{-1}\boldsymbol{N}^T\boldsymbol{\lambda}^*}$$

The advantage of the primal-dual method is that the dual problem has only the constraint $\boldsymbol{\lambda} \geq \mathbf{0}$ and is therefore often easier to solve than the primal problem.

- We may incorporate this torque optimization method into any control design methodology to reduce the tracking errors caused by the input (torque) saturation.