Advanced Robot Control and Learning

Prof. Sami Haddadin

Part 1

Advanced Robot Control

Advanced Robot Control - Outline

Chapter 6: Linear Parametrization and Identification of Robot Dynamics

Motivation – How to identify the Robot's Dynamics Equation?

Manipulator dynamics in joint space:

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau}_{\text{in}}$$

Model depends on:

- Joint torque $\boldsymbol{\tau}_{\text{in}}$ → measurable via sensors[1]
- Joint variables $\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}$ → measurable via sensors[2]
- Kinematic parameters (e.g. constant Denavit-Hartenberg Parameters), → measurable e.g. via CAD model
- Dynamic parameters (e.g. center of mass, inertia tensor) → often not precisely known…

Goal:

A procedure for identification of dynamic parameters

[1]For a perfectly torque-controlled robot: commanded torques = measured torques

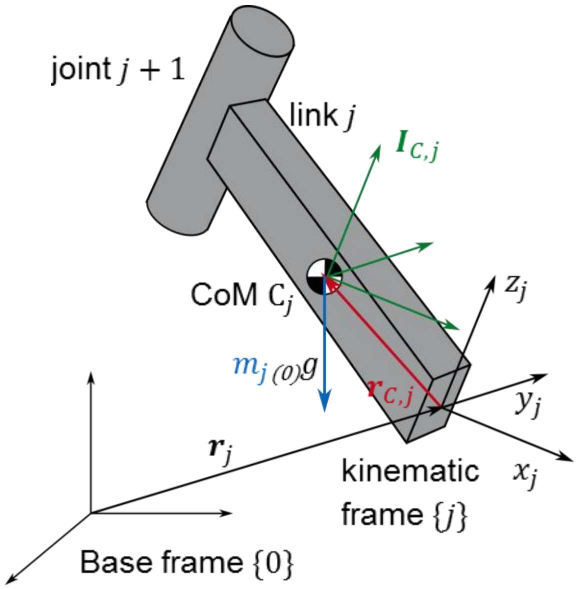
Linear Parametrization of Manipulator‘s Dynamic Model

Dynamic Parameters of a Link

- We consider the link j of a fully rigid robot
- Note: joint j is not shown in the figure for the sake of clarity (using the Modified Denavit-Hartenberg Parameter convention (MDH) leads to alignment of joint axis j with the z_j-axis of the kinematic frame $\{j\}$)
- Each link is characterized by 10 inertial parameters

$$m_j \qquad \boldsymbol{r}_{C,j} = \begin{bmatrix} r_{C,j_x} \\ r_{C,j_y} \\ r_{C,j_z} \end{bmatrix} \qquad \boldsymbol{I}_{C,j} = \begin{bmatrix} I_{C,j_{xx}} & I_{C,j_{xy}} & I_{C,j_{xz}} \\ & I_{C,j_{yy}} & I_{C,j_{yz}} \\ symm. & & I_{C,j_{zz}} \end{bmatrix}$$

- However robot dynamics depend in a nonlinear way on some of the parameters,
  - For example if we want to express the inertia tensor with respect to the kinematic frame $\{j\}$ we make use of the parallel axis theorem
  - ${}_{(j)}I_{C,j_{zz}} = {}_{(C)}I_{C,j_{zz}} + m_j \, {}_{(j)}r_{C,j_x}^2$



Kinetic and Gravitational Energy of a Link

- Goal: write the kinetic and gravitational energy such that the dynamic parameters only appear in a linear way
- The linear velocity of link j at its center of mass $\boldsymbol{v}_{C,j}$, with linear velocity $\boldsymbol{v}_j$ at the origin of frame $\{j\}$, angular velocity $\boldsymbol{\omega}_j$ and $\boldsymbol{r}_{C,j}$ describing the position of the center of mass relative to frame j, is given by

$$\boldsymbol{v}_{C,j} = \boldsymbol{v}_j + \boldsymbol{\omega}_j \times \boldsymbol{r}_{C,j}$$

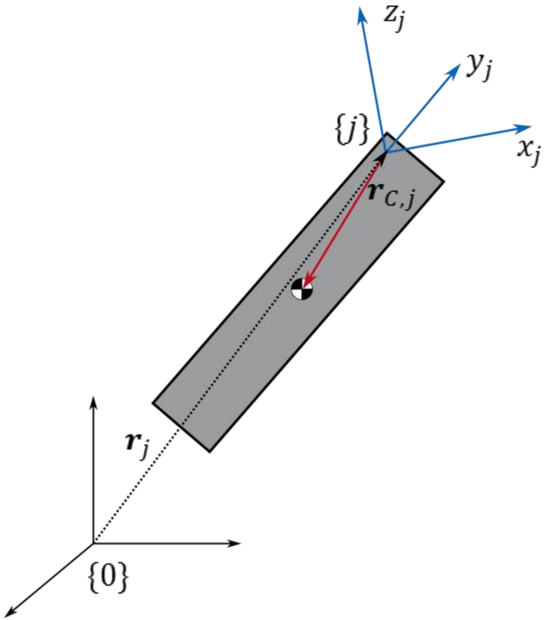
- With the inertia tensor $\boldsymbol{I}_{C,j}$ of link j with respect to its center of mass and mass m_j, we can derive
  - its kinetic energy T_j

$$T_j = \frac{1}{2} m_j \boldsymbol{v}_{C,j}^T \boldsymbol{v}_{C,j} + \frac{1}{2} \boldsymbol{\omega}_j^T \boldsymbol{I}_{C,j} \boldsymbol{\omega}_j$$

  - its potential energy U_j

$$U_j = -m_j \, {}_{(0)}\boldsymbol{g}^T (\boldsymbol{r}_j + \boldsymbol{r}_{C,j})$$

where ${}_{(0)}\boldsymbol{g}$ is the gravity vector expressed in the base frame $\{0\}$



Linear Parameterization of the Energies for one Link

- Expressing $\boldsymbol{v}_{C,j}$ in terms of $\boldsymbol{v}_j$, $\boldsymbol{\omega}_j$ and $\boldsymbol{r}_{C,j}$, and expressing the variables in frame $\{j\}$, the kinematic energy can be written as follows

$$T_j = \frac{1}{2}\left(m_j \, {}_{(j)}\boldsymbol{v}_j^T \, {}_{(j)}\boldsymbol{v}_j + 2 \, {}_{(j)}\boldsymbol{v}_j\left({}_{(j)}\boldsymbol{\omega}_j \times {}_{(j)}\boldsymbol{s}_j\right) + {}_{(j)}\boldsymbol{\omega}_j^T \, {}_{(j)}\boldsymbol{I}_j \, {}_{(j)}\boldsymbol{\omega}_j\right)$$

  where ${}_{(j)}\boldsymbol{I}_j$ is the constant symmetric inertia tensor of link j, expressed in frame $\{j\}$, and the constant vector ${}_{(j)}\boldsymbol{s}_j = m_j \, {}_{(j)}\boldsymbol{r}_{C,j}$

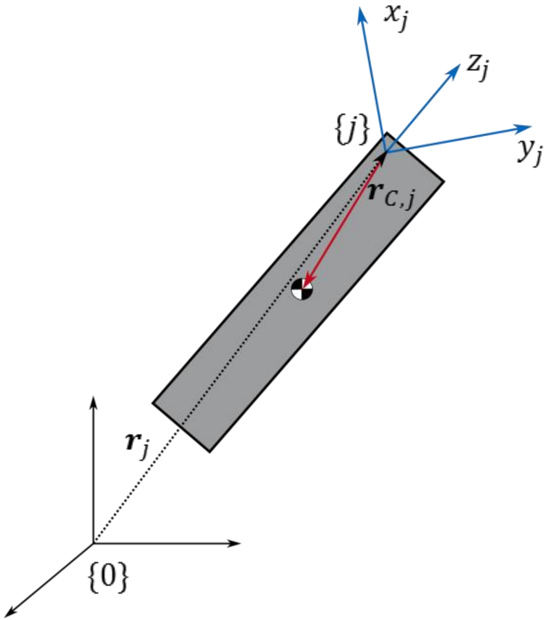
- For the potential energy:

$$U_j = -{}_{(0)}\boldsymbol{g}^T\left(m_j \boldsymbol{r}_j + {}^0\boldsymbol{R}_j \, {}_{(j)}\boldsymbol{s}_j\right)$$

  with ${}^0\boldsymbol{R}_j$ as the orientation of frame $\{j\}$ with regard to the base frame $\{0\}$

- Now we define the constant 10-dimensional parametric vector of

$$\boldsymbol{X}_j = \left({}_{(j)}\boldsymbol{I}_j(1{,}1{:}3), {}_{(j)}\boldsymbol{I}_j(2{,}2{:}3), {}_{(j)}\boldsymbol{I}_j(3{,}3), {}_{(j)}\boldsymbol{s}_j(1{:}3)^T, m_j\right)^T$$
$$= \left({}_{(j)}I_{j_{xx}}, {}_{(j)}I_{j_{xy}}, {}_{(j)}I_{j_{xz}}, {}_{(j)}I_{j_{yy}}, {}_{(j)}I_{j_{yz}}, {}_{(j)}I_{j_{zz}}, m_j \, {}_{(j)}r_{C,j_x}, m_j \, {}_{(j)}r_{C,j_y}, m_j \, {}_{(j)}r_{C,j_z}, m_j\right)^T$$



Linear Parameterization of the Energies for one Link

- Applying the Modified Denavit-Hartenberg Parameter convention, the rotation matrix ${}^{0}\boldsymbol{R}_j$ and the translational vector $\boldsymbol{r}_j$ can be obtained as values depending only on kinematic quantities
- Representing ${}_{(j)}\boldsymbol{\omega}_j$ and ${}_{(j)}\boldsymbol{v}_j$ in a recursive way

$$
{}_{(j)}\boldsymbol{\omega}_j = {}^{j}\boldsymbol{R}_{j-1}\ {}_{(j-1)}\boldsymbol{\omega}_j + \dot{q}_j \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}
$$

$$
{}_{(j)}\boldsymbol{v}_j = {}^{j}\boldsymbol{R}_{j-1}\ {}_{(j-1)}\boldsymbol{v}_{j-1} + {}^{j}\boldsymbol{R}_{j-1}\left({}_{(j-1)}\boldsymbol{\omega}_{(j-1)} \times \boldsymbol{r}_{(j-1),j}\right)
$$

- For ${}_{(1)}\boldsymbol{\omega}_1$ and ${}_{(1)}\boldsymbol{v}_1$, we have

$$
{}_{(1)}\boldsymbol{\omega}_1 = \dot{q}_1 \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}
$$

$$
{}_{(1)}\boldsymbol{v}_1 = {}_{(1)}\boldsymbol{\omega}_1 \times {}_{(1)}\boldsymbol{r}_{C,1}
$$

- Due to recursive formulation, the remaining ${}_{(j)}\boldsymbol{\omega}_j$ and ${}_{(j)}\boldsymbol{v}_j$ can be expressed as functions of $\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}$
  - ➢ T_j and U_j can be described in a linear form of $\boldsymbol{X}_j$

Overall Energies and Robot Regressor Formulation

- Robot kinetic and gravitational energies are the summation of the energies of each link:

$$T = \sum_{j=1}^{n} T_j, \qquad U = \sum_{j=1}^{n} U_j$$

- Considering the Lagrangian $L = T - U$, the Euler-Lagrange equation will only involve linear operations on T *and* U:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\boldsymbol{q}}}\right) - \frac{\partial L}{\partial \boldsymbol{q}} = \boldsymbol{\tau}$$

- Consequently, we can represent the robot dynamics in a linear way:

$$\boldsymbol{\tau} = \boldsymbol{Y}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})\boldsymbol{X}$$

  - $\boldsymbol{X} \in \mathbb{R}^{10n}, \boldsymbol{X}^T = [\boldsymbol{X}_1^T, \dots, \boldsymbol{X}_n^T]$
  - $\boldsymbol{Y} \in \mathbb{R}^{n \times 10n}$, depends only on kinematic quantities

The Minimum Parameter Set

The Regressor Matrix

- Regressor matrix has a block upper triangular structure:

$$Y(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}) = \begin{bmatrix} \boldsymbol{Y}_{1,1} & \boldsymbol{Y}_{1,2} & \boldsymbol{Y}_{1,3} & \cdots & \boldsymbol{Y}_{1,n-1} & \boldsymbol{Y}_{1,n} \\ \boldsymbol{0} & \boldsymbol{Y}_{2,2} & \boldsymbol{Y}_{2,3} & \cdots & \boldsymbol{Y}_{2,n-1} & \boldsymbol{Y}_{2,n} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{Y}_{3,3} & \cdots & \boldsymbol{Y}_{3,n-1} & \boldsymbol{Y}_{3,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{Y}_{n-1,n-1} & \boldsymbol{Y}_{n-1,n} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{Y}_{n,n} \end{bmatrix}$$

  where each $\boldsymbol{Y}_{i,j},\ i,j = 1, \ldots, n$ is a 1×10 row vector
- In general, not all parameters can be independently identified:
  - Some parameters may play no role in the dynamic model → associated column in $\boldsymbol{Y}$ is zero
  - Some parameters may appear only combined with others → associated columns in $\boldsymbol{Y}$ are linearly dependent

To be more specific about these parameters there are some conditions…

A closer Look on the Inertial Parameters

- If **first joint** is **rotational**, the parameters ${}_{(1)}\boldsymbol{I}_1(1{,}1{:}3)$, ${}_{(1)}\boldsymbol{I}_1(2{,}2{:}3)$, ${}_{(1)}\boldsymbol{s}_1(3)$ and m_1 have **no effect on the dynamical model**
- If **axis** of **first joint** is parallel to vector of **gravity,** the parameters ${}_{(1)}\boldsymbol{s}_1(1{:}2)$ have **no effect on the dynamical model**
- Similar conditions for translational joints can be found in the literature

- Therefore, instead of $10n$ parameters, we divide the parameters into two groups
  I. Independent parameters $\boldsymbol{X}_b$, corresponding column group $\boldsymbol{Y}_b$
  II. Dependent parameters $\boldsymbol{X}_d$, corresponding column group $\boldsymbol{Y}_d$
  $\rightarrow \boldsymbol{Y}_d = \boldsymbol{Y}_b \boldsymbol{K}$

Now, one can show:

$$\begin{aligned}\boldsymbol{\tau} &= \boldsymbol{Y}\boldsymbol{X} \\ &= [\boldsymbol{Y}_b \ \ \boldsymbol{Y}_d]\begin{bmatrix}\boldsymbol{X}_b \\ \boldsymbol{X}_d\end{bmatrix} \\ &= \boldsymbol{Y}_b \underbrace{[\boldsymbol{X}_b + \boldsymbol{K}\boldsymbol{X}_d]}_{\boldsymbol{\beta}_b} \\ \Rightarrow \boldsymbol{\tau} &= \boldsymbol{Y}_b \boldsymbol{\beta}_b\end{aligned}$$

These grouped parameters are called **base parameters**

Example: 2DoF planar Manipulator

Example: 2DoF planar Manipulator

- $n = 2 \rightarrow$ initially 20 standard parameters
- Aforementioned conditions and center of mass locations → some can be ignored:

$$m_1,\ {}_{(j)}\boldsymbol{I}_j(1{,}1{:}3),\ {}_{(j)}\boldsymbol{I}_j(2{,}2{:}3),\ {}_{(j)}\boldsymbol{s}_j(2{:}3)$$

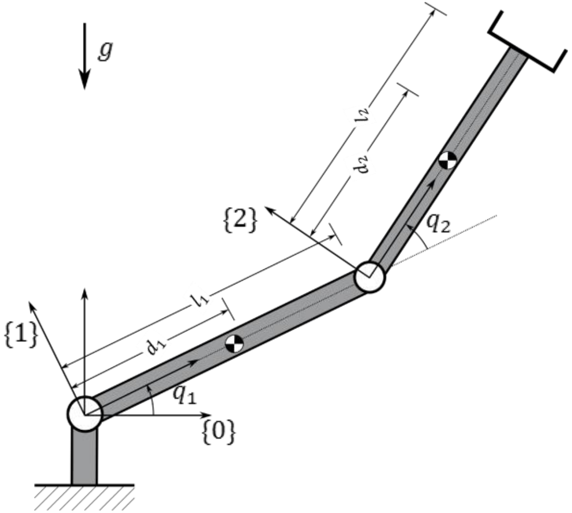
- Only 5 parameters remain → 2×5 Regressor matrix
- After building the regressor matrix, it can be shown that the column associated to m_2 is dependent on two other columns → Regrouping
- Final linear model:

$$\begin{bmatrix} g_0\cos(q_1) & \ddot{q}_1 & l_1\cos(q_2)(2\ddot{q}_1+\ddot{q}_2) - l_1\sin(q_2)\,(\dot{q}_2^2 + 2\,\dot{q}_1\dot{q}_2) + g_0\cos(q_1+q_2) & \ddot{q}_1+\ddot{q}_2 \\ 0 & 0 & l_1\cos(q_2)\,\ddot{q}_1 + l_1\sin(q_2)\,\dot{q}_1^2 + g_0\cos(q_1+q_2) & \ddot{q}_1+\ddot{q}_2 \end{bmatrix} \begin{bmatrix} \beta_{b,1} \\ \beta_{b,2} \\ \beta_{b,3} \\ \beta_{b,4} \end{bmatrix}$$

$$= \boldsymbol{Y}_b \boldsymbol{\beta}_b = \boldsymbol{\tau}$$

- With the following base parameters:

$$\beta_{b,1} = m_1 d_1 + m_2 l_1 \qquad \beta_{b,2} = {}_{(1)}\boldsymbol{I}_1(3{,}3) + m_2 l_1^2 \qquad \beta_{b,3} = m_2 d_2 \qquad \beta_{b,4} = {}_{(2)}\boldsymbol{I}_2(3{,}3)$$



Parameter Identification

Identification Task

- Obtained linear model can be useful only if the values of the base parameters are known
- Robot manufacturers provide at most only a few principle dynamic parameters (e.g. link masses)
- Some estimations are possible (e.g. by CAD models)

  → a way to identify the parameters…
- The more accurate the values for the base parameters: the less difference between estimated torques $\hat{\boldsymbol{\tau}}(= \boldsymbol{Y}_b \boldsymbol{\beta}_b)$ and the measured torques $\boldsymbol{\tau}$
- Hence, identification task would be to find the minimal set of parameters that can minimize the error on torque estimation:

$$\boldsymbol{\beta}_b = \arg\min_{\boldsymbol{\beta}_b} \sum_{k=1}^{p} \| \boldsymbol{Y}_b^{(k)} \boldsymbol{\beta}_b - \boldsymbol{\tau}^{(k)} \|_2^2$$

for p measurements sensed at time steps $[t_1, t_2, \dots, t_p]$

Identification Task

- We try to find $\boldsymbol{\beta}_b$ that satisfies:

$$\underbrace{\begin{bmatrix} \boldsymbol{Y}_b(\boldsymbol{q}(t_1), \dot{\boldsymbol{q}}(t_1), \ddot{\boldsymbol{q}}(t_1)) \\ \boldsymbol{Y}_b(\boldsymbol{q}(t_2), \dot{\boldsymbol{q}}(t_2), \ddot{\boldsymbol{q}}(t_2)) \\ \vdots \\ \boldsymbol{Y}_b(\boldsymbol{q}(t_p), \dot{\boldsymbol{q}}(t_p), \ddot{\boldsymbol{q}}(t_p)) \end{bmatrix}}_{\overline{\boldsymbol{Y}}_b} \boldsymbol{\beta}_b = \underbrace{\begin{bmatrix} \boldsymbol{\tau}(t_1) \\ \boldsymbol{\tau}(t_2) \\ \vdots \\ \boldsymbol{\tau}(t_p) \end{bmatrix}}_{\overline{\boldsymbol{\tau}}}$$

- Assuming that $\overline{\boldsymbol{Y}}_b$ (known as Information matrix) is a full column rank matrix[3] , $\boldsymbol{\beta}_b$ can be found via:

$$\boldsymbol{\beta}_b = \overline{\boldsymbol{Y}}_b^{\#} \overline{\boldsymbol{\tau}} = \left(\overline{\boldsymbol{Y}}_b^T \overline{\boldsymbol{Y}}_b \right)^{-1} \overline{\boldsymbol{Y}}_b^T \overline{\boldsymbol{\tau}}$$

Trajectory Optimization

Motivation – Trajectory Optimization

- Objective: find $\boldsymbol{\beta}_b = \overline{\boldsymbol{Y}}_b^{\#}\overline{\boldsymbol{\tau}} = \left(\overline{\boldsymbol{Y}}_b^T\overline{\boldsymbol{Y}}_b\right)^{-1}\overline{\boldsymbol{Y}}_b^T\overline{\boldsymbol{\tau}}$

→ $\overline{\boldsymbol{Y}}_b$ should have full rank!
→ We need to consider $\boldsymbol{Y}_b(\boldsymbol{q}(t_i), \dot{\boldsymbol{q}}(t_i), \dot{\boldsymbol{q}}(t_i))$
→ More specifically $\boldsymbol{q}(t)$

$$\underbrace{\begin{bmatrix} \boldsymbol{Y}_b(\boldsymbol{q}(t_1), \dot{\boldsymbol{q}}(t_1), \ddot{\boldsymbol{q}}(t_1)) \\ \boldsymbol{Y}_b(\boldsymbol{q}(t_2), \dot{\boldsymbol{q}}(t_2), \ddot{\boldsymbol{q}}(t_2)) \\ \vdots \\ \boldsymbol{Y}_b(\boldsymbol{q}(t_p), \dot{\boldsymbol{q}}(t_p), \ddot{\boldsymbol{q}}(t_p)) \end{bmatrix}}_{\overline{\boldsymbol{Y}}_b} \boldsymbol{\beta}_b = \underbrace{\begin{bmatrix} \boldsymbol{\tau}(t_1) \\ \boldsymbol{\tau}(t_2) \\ \vdots \\ \boldsymbol{\tau}(t_p) \end{bmatrix}}_{\overline{\boldsymbol{\tau}}}$$

- Limitations:
  - Noisy measurements of joint angles and torques[4]
- What if model has been identified that does not work well for some other sample sets?

Choice of excitation an important phase of parameter identification!

Optimization Goal

- We wish to find $\bar{\boldsymbol{Y}}_b$, such that it is not sensitive to the aforementioned problems
- The condition number κ measures how sensitive our result for $\boldsymbol{\beta}_b$ is to small changes or errors $\boldsymbol{q}(t_i), \dot{\boldsymbol{q}}(t_i), \ddot{\boldsymbol{q}}(t_i)$ and $\boldsymbol{\tau}(t_i)$,

$$\kappa(\bar{\boldsymbol{Y}}_b) = \frac{\lambda_{max}}{\lambda_{min}}$$

where λ_{max} and λ_{min} are the largest and the smallest eigenvalues of $\bar{\boldsymbol{Y}}_b$, respectively

Goal: find a trajectory $\boldsymbol{q}_d(t)$ which eventually minimizes κ

- Defining a trajectory $\boldsymbol{q}_d(t)$ parameterized by a^*_{j,m_f}:

$$\left\{a^*_{j,m_f}\right\} = \underset{a_{j,m_f}}{\operatorname{argmin}}\, \kappa(\boldsymbol{Y}_b(\boldsymbol{q}_d, \dot{\boldsymbol{q}}_d, \ddot{\boldsymbol{q}}_d))$$

for $\boldsymbol{q}_d\left(t_i, a_{j,m_f}\right); t_i = t_0, \dots, t_p$

Subject to:

$$\boldsymbol{q}_{d,min} \leq \boldsymbol{q}_d \leq \boldsymbol{q}_{d,max}$$ To consider joint angle limits

$$\dot{\boldsymbol{q}}_{d,min} \leq \dot{\boldsymbol{q}}_d \leq \dot{\boldsymbol{q}}_{d,max}$$ To consider joint velocity limits

$$P_{EE}(\boldsymbol{q}_d) \in P_{EE,v}$$ To consider workspace of manipulator

Trajectory Structure

Trajectories based on Fourier series:

- Better signal-to-noise ratio (due to periodicity)
- Limits (e.g. joint limits) can be better pointed (by linear scaling of the amplitude of the trigonometric function)
- Smooth behavior (due to the differentiability of trigonometric functions)

For better convergence in the optimization problem, a polynomial term is added:

$$q_{d,j}(t) = \delta_j(t) + \lambda_j(t)$$

$$\delta_j(t) = \sum_{m_f} a_{j,m_f} \cos\left(\frac{m_f \pi}{T_f} t\right)$$

$$\lambda_j(t) = \sum_k \lambda_{jk}\, t^k$$

- $j = 1, \ldots, \mathrm{DoF}$
- a_{j,m_f}: amplitude of the m_f cosine
- T_f: time period

We find the polynomial coefficients by introducing the boundary conditions

- The values of $q_j(0/t_p), \dot{q}_j(0/t_p), \ddot{q}_j(0/t_p)$ can be freely chosen,
- getting the optimized $\delta_j(0/t_p), \dot{\delta}_j(0/t_p), \ddot{\delta}_j(0/t_p)$ after each iteration,
- $\lambda_j(0/t_p), \dot{\lambda}_j(0/t_p), \ddot{\lambda}_j(0/t_p)$ can then be obtained via a linear system of equations

Modeling of Friction

Including Linear Parameterization of Friction Model

- We apply the same approach to model friction
- For this we define an additional regressor matrix $\boldsymbol{Y}_f$ and parameter set $\boldsymbol{\beta}_f$ such that

$$\boldsymbol{Y}_f \boldsymbol{\beta}_f = \underbrace{\operatorname{diag}\left(\operatorname{sign}\left(\dot{q}_j\right) F_{c,j}\right)}_{\text{coulomb friction}} + \underbrace{\operatorname{diag}\left(\dot{q}_j F_{v,j}\right)}_{\text{viscous friction}}$$

$$\boldsymbol{Y}_f = \begin{bmatrix} \operatorname{sign}(\dot{q}_1) & \dot{q}_1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \operatorname{sign}(\dot{q}_2) & \dot{q}_2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdots & \operatorname{sign}(\dot{q}_n) & \dot{q}_n \end{bmatrix}, \qquad \boldsymbol{\beta}_f = \begin{bmatrix} F_{c,1} \\ F_{v,1} \\ F_{c,2} \\ F_{v,2} \\ \vdots \\ F_{c,n} \\ F_{v,n} \end{bmatrix}$$

- Combining the two aforementioned linear models leads to a *unified linear model* in the parameters $\boldsymbol{\beta}_{tot}$

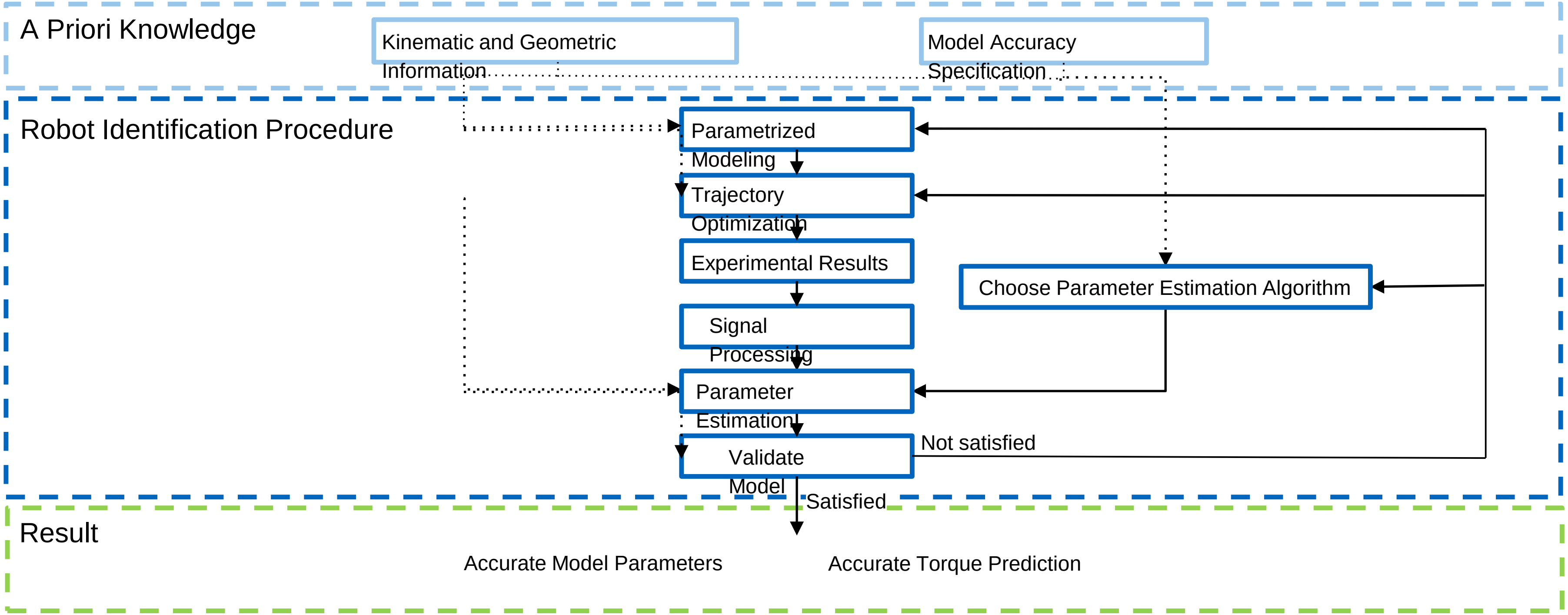
$$\boldsymbol{Y}_{tot} \boldsymbol{\beta}_{tot} = \begin{bmatrix} \overline{\boldsymbol{Y}}_b & \boldsymbol{Y}_f \end{bmatrix} \begin{bmatrix} \boldsymbol{\beta}_b \\ \boldsymbol{\beta}_f \end{bmatrix}$$

➢ Simultaneous identification of manipulator's base parameters and friction parameters

➢ Friction influences can also be taken into account during trajectory optimization

Identification Procedure

Schematic Representation of Identification Procedure

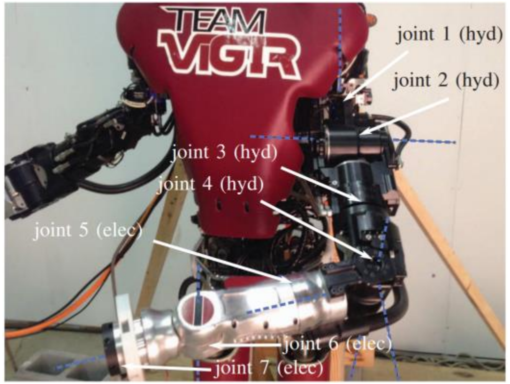


Example: Two 7-DoF Manipulator Arms

“Modeling, Identification and Joint Impedance Control of the Atlas arm”

Schappler et al., 2015

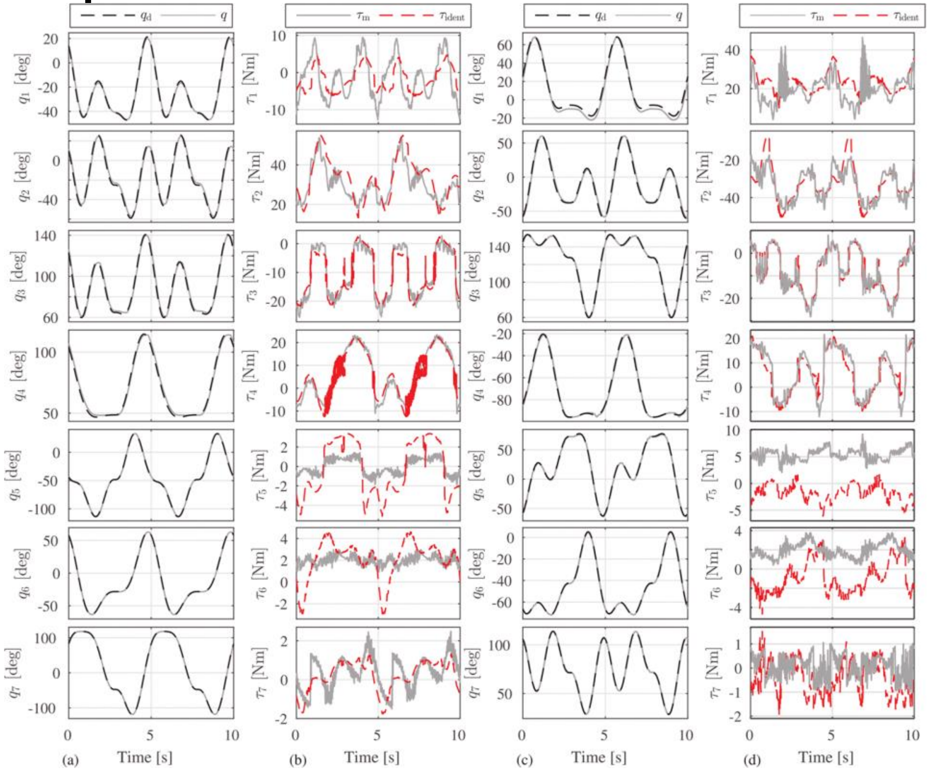
Example: Two 7-DoF Manipulator Arms



Atlas robot upper body, Boston Dynamics Inc.

Left to right:
Position tracking and measured/modeled torques for each joint for left and right arm, respectively

Note: the first 4 joints are hydraulic and the last 3 joints are electric



Adaptive Control

Adaptive Control – Motivation

What if parameters vary in time?
(When can it happen?)

What if there is a variable load on the manipulator?
(What would be an example in real life?)

⇒ The identification should be carried out online!

Slotine & Li[5] Controller Revisited

Control law:

$$\tau = M(q)\dot{v} + C(q,\dot{q})v + g(q) - Kr$$

$$v = \dot{q} - r, \qquad r = (\dot{q}_d - \dot{q}) + \Lambda(q_d - q)$$

Λ, K: Diagonal, constant matrices

After linearly parametrizing the model:

$$\tau = Y(q,\dot{q},v,\dot{v})X - Kr$$

If X is not known in advance, we start with an estimation $\widehat{X}$:

$$\tau = Y(q,\dot{q},v,\dot{v})\widehat{X} - Kr$$

Resulting in the following control law:

$$\tau = M(q)\dot{v} + C(q,\dot{q})v + g(q) + Y(q,\dot{q},v,\dot{v})\widetilde{X} - Kr$$

$$\widetilde{X} = \widehat{X} - X$$

Closed-loop behavior

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = M(q)\dot{v} + C(q,\dot{q})v + g(q) + Y(q,\dot{q},v,\dot{v})\widetilde{X} - Kr$$

$$\Rightarrow M(q)\dot{r} + C(q,\dot{q})r + Kr = Y(q,\dot{q},v,\dot{v})\widetilde{X}$$

Goal: find an adaption law for $\widetilde{X}$, so that tracking error r decreases

[5] Slotine, J. J. E., & Li, W. (1991). *Applied nonlinear control* (Vol. 199, No. 1). Englewood Cliffs, NJ: Prentice hall.

Adaption Law for Slotine & Li Controller

- It is well known that for such a system the adaption law can be as follows:

$$\dot{\tilde{\boldsymbol{X}}} = -\boldsymbol{\Gamma}^{-1}\boldsymbol{Y}^T\boldsymbol{r}$$

  where $\boldsymbol{\Gamma}$ is a symmetric positive definite matrix

- Linear model of manipulator enables adaptive controller design which can compensate for model uncertainties to have better tracking behavior
- Verification of improved tracking behavior: assured decrease of cost function for tracking error