Lesson 22:

I. Inverse Dynamics Control

A. Joint Space IDC (cont'd) (SHV 9.3.1 / 7.2.1)

2nd edition — pdf version on canvas

B. Task Space IDC (SHV 9.3.2 / 7.2.2)

II. Adaptive IDC (SHV 9.3.4 / 7.3.2)

I. Inverse Dynamics Control

A. Joint Space IDC (cont'd)

<u>Recall</u>: Final control for following a trajectory $q^d(t)$

$$u = M(q)\,a_q + C(q,\dot q)\,\dot q + G(q)$$

$$a_q = \ddot q^d - K_p \tilde q - K_d \dot{\tilde q}$$

$$\Rightarrow \quad q(t) \to q^d(t) \qquad t \to \infty$$

<u>Energy Shaping</u>: Outer loop controller could insert desired dynamics instead of tracking reference trajectory.

$$a_q := \bar M(q)^{-1}\left[v - \bar C(q,\dot q)\,\dot q - \bar G(q)\right]$$

where $\bar M \in \mathbb{R}^{n\times n} \Rightarrow \bar C \in \mathbb{R}^{n\times n}$ and $\bar G \in \mathbb{R}^n$ are <u>desired</u> dynamics terms, and v is the new input in the "shaped" dynamics

plug into original dynamics

$$\ddot q = a_q \Rightarrow \bar M \ddot q + \bar C \dot q + \bar G = v$$

This is called "energy shaping" when choosing $\bar M$ and $\bar G$ to correspond to another physical system with well-defined energy terms:

$$\bar K = \frac{1}{2}\dot q^T \bar M(q)\,\dot q\ , \qquad \bar P = \bar P(q) \ \text{ s.t. } \ \bar G = \nabla \bar P = \frac{\partial \bar P}{\partial q}$$

- <u>Special case</u>: Potential energy shaping $(\bar{M}=M,\ \bar{C}=C)$

  Results in simpler controller when plugged into inner loop.

  Inner loop controller:

$$u = \cancel{M}\,\cancel{\bar{M}^{-1}}\underbrace{\left[v - \cancel{\bar{C}\dot{q}} - \bar{G}\right]}_{a_q} + \cancel{C\dot{q}} + G$$

$$= v + G - \bar{G}$$

  Plug into robot dynamics

$$\Rightarrow \quad M\ddot{q} + C\dot{q} + \cancel{G} = u = v + \cancel{G} - \bar{G}$$

$$\Rightarrow \quad M\ddot{q} + C\dot{q} + \bar{G} = v$$

  - Note: Gravity compensation is a special case of PE shaping $(\bar{G}=0)$
  - Warning: Underactuated cases are <u>much</u> <u>harder</u>
    - Matching conditions (PDE's) determine what can be shaped.
    - Limitations on acheivable dynamics

B. Task-Space IDC

Same inner-loop controller $(\ddot{q} = a_q)$ but consider new outer loop.

<u>Goal</u>: Track desired end effector pose trajectory $X^d(t)$ for some representation of workspace, e.g., SE(3)

Recall $\dot{X} = J(q)\dot{q}$ for manipulator Jacobian $J(q)$

$\rightarrow$ If $\dot{X} = V^b_{st}$, then $J^b_{st}(q) \in \mathbb{R}^{6\times n}$

$\rightarrow$ If $\dot{X} = V^s_{st}$, then $J^s_{st}(q) \in \mathbb{R}^{6\times n}$

Let $n=6$ so that we have a square J.

$$\dot{X} = J(q)\dot{q} \Rightarrow \ddot{X} = J\ddot{q} + \dot{J}\dot{q}$$

new input, desired task space acceleration.

Given $\ddot{q} = a_q$, choose $a_q := J^{-1}(a_x - \dot{J}\dot{q})$

(first outer-loop controller)

$$\Rightarrow \ddot{q} = J^{-1}(a_x - \dot{J}\dot{q}) \Rightarrow J\ddot{q} + \dot{J}\dot{q} = a_x$$

$$\Rightarrow \ddot{X} = a_x \leftarrow$$ new input for end-effector accel.

Define 2^{nd} outer-loop control law to insert linear dynamics:

$$a_x := \ddot{X}^d - K_p\underbrace{(X - X^d)}_{\tilde{X}} - K_d\underbrace{(\dot{X} - \dot{X}^d)}_{\dot{\tilde{X}}}$$

$$\Rightarrow \ddot{\tilde{X}} + K_d\dot{\tilde{X}} + K_p\tilde{X} = 0$$ (closed-loop task error dynamics)

Choose $K_p, K_d > 0$ then $(\tilde{X}, \dot{\tilde{X}}) \xrightarrow[\text{fast}]{\text{exp}} (0,0)$

exponentially stable $\Rightarrow X \rightarrow X^d(t)$

Benefit: No need for inverse kinematics in real-time.

Limitations:

- Must avoid singularities to maintain invertibility of J in a_q control law.
- We considered square $J \in \mathbb{R}^{n \times n}$
  - 6-DOF robot for $W \subset SE(3)$
  - 3-DOF " " $W \subset SE(2)$
  - 3-DOF " " $W \subset SO(3)$
- For non-square J, can use pseudo-inverse in some cases.

II. Adaptive Control

Problem: Model uncertainty (e.g., parametric uncertainty)
Unmodeled dynamics (e.g., actuator)

Recall IDC input: $u = \hat{M}(q)\, a_q + \hat{C}(q,\dot{q})\dot{q} + \hat{G}(q)$

where $\hat{(\cdot)}$ is a model estimate and

$$\tilde{(\cdot)} = \hat{(\cdot)} - (\cdot) \text{ is model error}$$

Two classical ways of dealing with model uncertainty

- Robust Control: enables bounded tracking by defining certain bounds on model error.
  - Requires many parameters to be identified for these bounds.
  - Covered in SHV 9.3.3 / 7.3.1

- Adaptive Inverse Dynamics Control:

Deals with parametric uncertainty

Recall "linearity in the parameters"

regressor matrix   parameter vector

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = Y(q,\dot{q},\ddot{q})\,\Theta$$

We will vary parameters in $\hat{M}, \hat{C}, \hat{G}$ over time (adapt)

We want to find $\hat{\Theta}$ s.t. $Y(q,\dot{q},\ddot{q})\,\Theta = Y(q,\dot{q},\ddot{q})\,\hat{\Theta}$

(parameter estimate)

$$M\ddot{q} + C\dot{q} + G = u := \hat{M}a_q + \hat{C}\dot{q} + \hat{G} \qquad (*)$$

$$\Rightarrow \ddot{q} = M^{-1}\left(\hat{M}a_q + \tilde{C}\dot{q} + \tilde{G}\right) \qquad \text{recall } \hat{M} = \tilde{M} + M$$

$$= a_q + M^{-1}\left(\tilde{M}a_q + \tilde{C}\dot{q} + \tilde{G}\right)$$

Plug in $a_q = \ddot{q}^d(t) - K_d\dot{\tilde{q}} - K_p\tilde{q}$, where $\tilde{q} = q - q^d$

Simplify (ex. 9-11 in SHV)

$\Rightarrow$ closed-loop

$$\ddot{\tilde{q}} + K_d\dot{\tilde{q}} + K_p\tilde{q} = \hat{M}^{-1}\,Y(q,\dot{q},\ddot{q})\,\tilde{\Theta}$$

where $\tilde{\Theta} = \hat{\Theta} - \Theta$ is parametric error vector.

Let $e = \begin{bmatrix} \tilde{q} \\ \dot{\tilde{q}} \end{bmatrix}$ be tracking error state vector

$$A = \underbrace{\begin{bmatrix} 0 & I \\ -K_p & -K_d \end{bmatrix}}_{\text{designd Hurwitz by choice of } K_p, K_d > 0} \qquad B = \begin{bmatrix} 0 \\ I \end{bmatrix}$$

(all e-values are in LHS of complex plane, i.e. all e-values have negative real part)

$$\Rightarrow \dot{e} = Ae + B \underline{\hat{M}^{-1} Y(q, \dot{q}, \ddot{q})} \tilde{\Theta}$$

$$= Ae + B \Phi \tilde{\Theta} \quad \text{where} \quad \Phi := \hat{M}^{-1} Y(q, \dot{q}, \ddot{q})$$

To be continued! ...