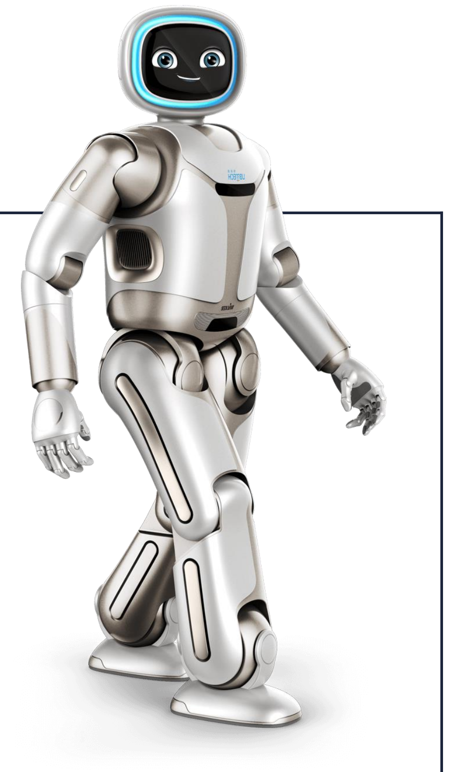
Advanced Robotics

ENGG5402 Spring 2023

Fei Chen

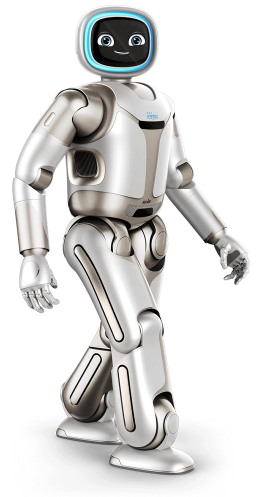


Topics:

- Trajectory Tracking Control

Readings:

- Siciliano: Sec. 8.5



Inverse Dynamics Control

given the robot dynamic model

$$\boxed{\underbrace{M(q)\ddot{q} + n(q,\dot{q})}_{c(q,\dot{q}) + g(q) + \text{friction model}} = u}$$

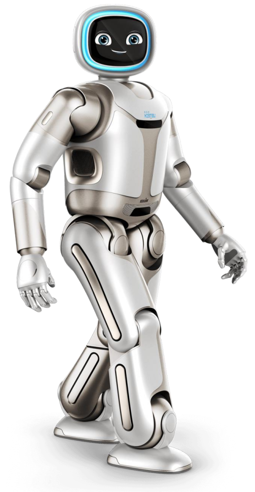
and a twice-differentiable desired trajectory for $t \in [0, T]$

$$q_d(t) \rightarrow \dot{q}_d(t), \ddot{q}_d(t)$$

applying the feedforward torque in nominal conditions

$$\boxed{u_d = M(q_d)\ddot{q}_d + n(q_d, \dot{q}_d)}$$

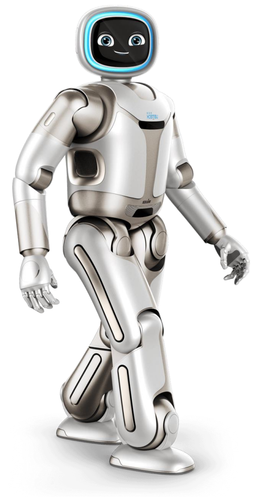
yields exact reproduction of the desired motion, provided that $q(0) = q_d(0), \dot{q}(0) = \dot{q}_d(0)$ (initial matched state)



In Practice …

a number of differences from the nominal condition

- initial state is "not matched" to the desired trajectory $q_d(t)$
- disturbances on the actuators, truncation errors on data, …
- inaccurate knowledge of robot dynamic parameters (link masses, inertias, center of mass positions)
- unknown value of the carried payload
- presence of unmodeled dynamics (complex friction phenomena, transmission elasticity, …)



Introducing Feedback

$$\hat{u}_d = \widehat{M}(q_d)\ddot{q}_d + \hat{n}(q_d, \dot{q}_d)$$

With $\widehat{M}, \hat{n}$ estimates of terms (or coefficients) in the dynamic model

note: $\hat{u}_d$ can be computed off line [e.g., by $\widehat{NE}_\alpha(q_d, \dot{q}_d, \ddot{q}_d)$]

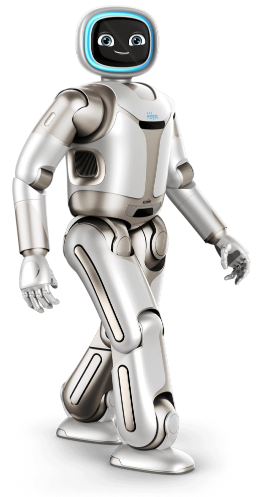
feedback is introduced to make the control scheme more robust

different possible implementations depending on amount of computational load share

- OFF LINE ↔ (open loop)
- ON LINE ↔ (closed loop)

two-step control design:

1. compensation (feedforward) or cancellation (feedback) of nonlinearities
2. synthesis of a linear control law stabilizing the trajectory error to zero



A Series of Trajectory Controllers

1. inverse dynamics compensation (FFW) + PD

$$\boxed{u = \hat{u}_d + K_P(q_d - q) + K_D(\dot{q}_d - \dot{q})}$$

2. inverse dynamics compensation (FFW) + variable PD

$$\boxed{u = \hat{u}_d + \widehat{M}(q_d)[K_P(q_d - q) + K_D(\dot{q}_d - \dot{q})]}$$

typically, only local stabilization of trajectory error $e(t) = q_d(t) - q(t)$

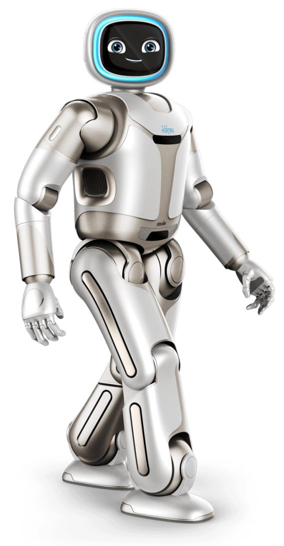
3. feedback linearization (FBL) + [PD+FFW] = “COMPUTED TORQUE”

$$\boxed{u = \widehat{M}(q)[\ddot{q}_d + K_P(q_d - q) + K_D(\dot{q}_d - \dot{q})] + \hat{n}(q, \dot{q})}$$

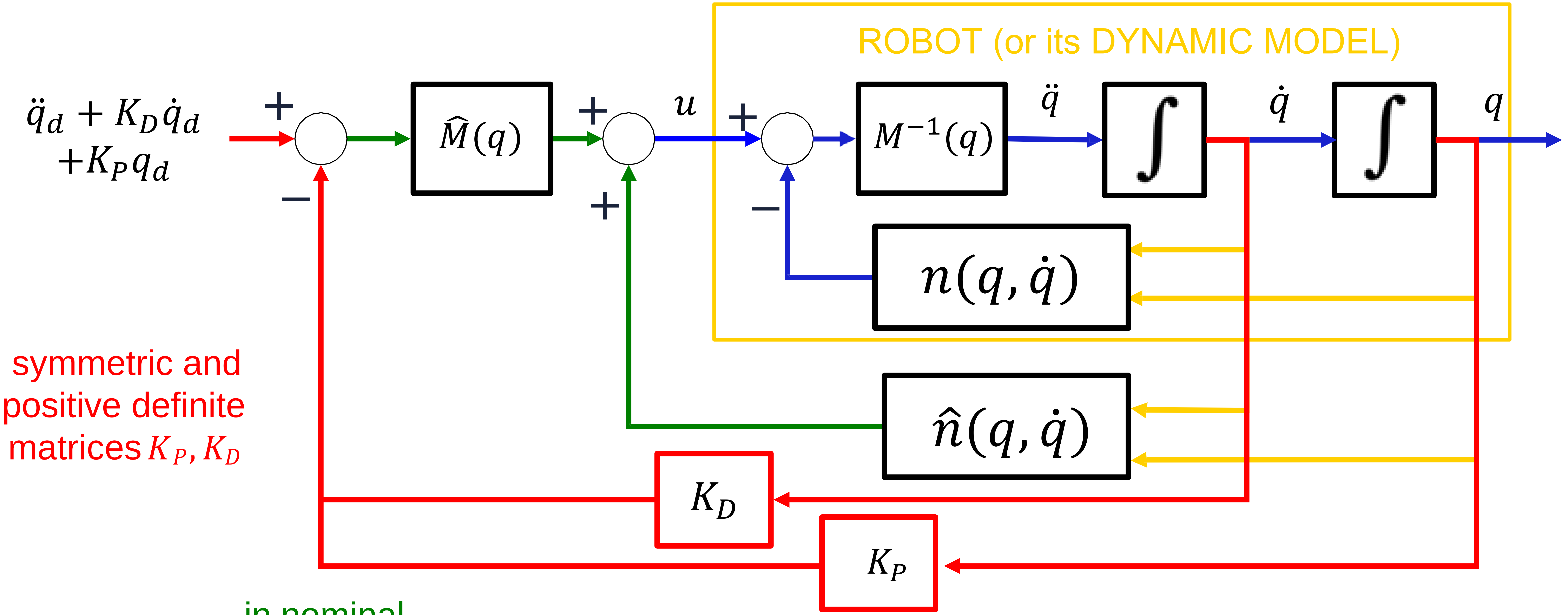
4. feedback linearization (FBL) + [PID+FFW]

$$\boxed{u = \widehat{M}(q)\left[\ddot{q}_d + K_P(q_d - q) + K_D(\dot{q}_d - \dot{q}) + K_I \int (q_d - q)\, dt\right] + \hat{n}(q, \dot{q})}$$

more robust to uncertainties, but also more complex to implement in real time



Feedback Linearization Control



symmetric and positive definite matrices K_P, K_D

in nominal conditions ($\hat M = M, \hat n = n$)

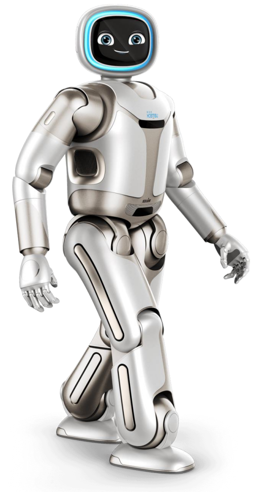
$$\underbrace{M(q)\ddot q + n(q,\dot q)}_{\text{nonlinear robot dynamics}} = u = \underbrace{M(q)a + n(q,\dot q)}_{\text{nonlinear control law}}$$

$$\ddot q = a$$

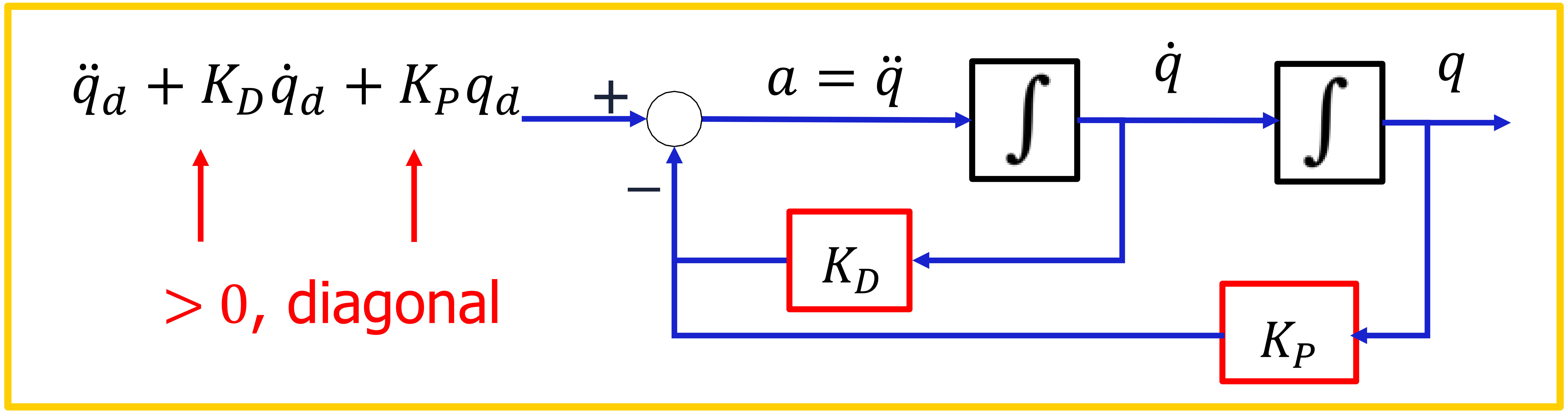
linear and decoupled system

global asymptotic stabilization

$$a = \ddot q_d + K_D(\dot q_d - \dot q) + K_P(q_d - q)$$



Interpretation in the Linear Domain



under feedback linearization control, the robot has a dynamic behavior that is invariant, linear and decoupled in its whole workspace ($\forall(q, \dot{q})$)

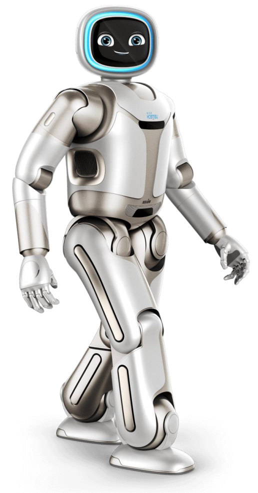
linearity

error transients $e_i = q_{di} - q_i \rightarrow 0$ exponentially, prescribed by K_{PI}, K_{DI} choice

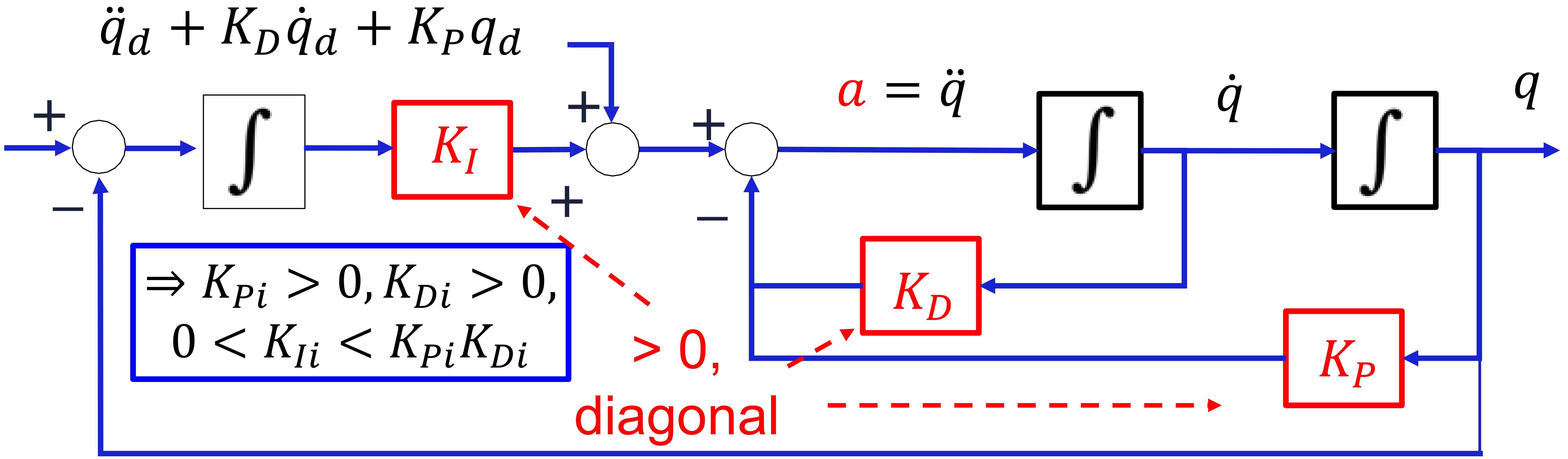
decoupling

each joint coordinate q_i evolves independently from the others, forced by a_i

$$\ddot{e} + K_D\dot{e} + K_P e = 0 \Leftrightarrow \ddot{e}_i + K_{Di}\dot{e}_i + K_{Pi}e_i = 0$$



If PID



$\Rightarrow K_{Pi} > 0, K_{Di} > 0,$
$0 < K_{Ii} < K_{Pi}K_{Di}$

$> 0,$ diagonal

$$\ddot{q} = a = \ddot{q}_d + K_D(\dot{q}_d - \dot{q}) + K_P(q_d - q) + K_I \int (q_d - q)\, d\tau \qquad e = q_d - q$$

$\underset{(1)}{\Rightarrow} e_i = q_{di} - q_i (i = 1, \dots, N)$ $\underset{(2)}{\Rightarrow} \ddot{e}_i + K_{Di}\dot{e}_i + K_{Pi}e_i + K_{Pi} \int e_i \, d\tau = 0$

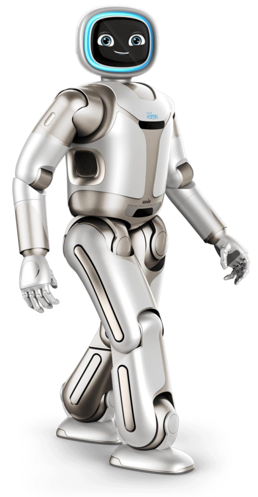
$\mathcal{L}[e_i(t)] \underset{(3)}{\Rightarrow} \left(s^2 + K_{Di}s + K_{Pi} + K_{Ii}\frac{1}{s}\right) e_i(s) = 0$

$s \times \underset{(4)}{\Rightarrow} (s^3 + K_{Di}s^2 + K_{Pi}s + K_{Ii}) e_i(s) = 0$ $\underset{(5)}{\Rightarrow}$

3	1	K_{Pi}
2	K_{Di}	K_{Ii}
1	$(K_{Di}K_{Pi} - K_{Ii})/K_{Di}$	
0	K_{Ii}	

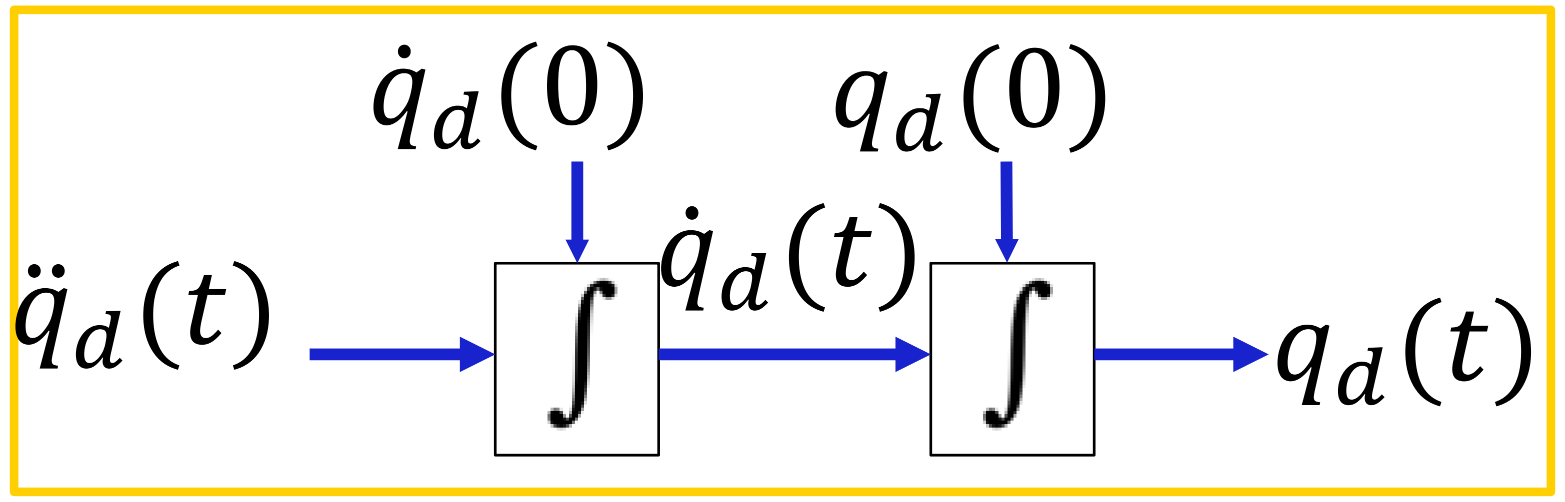
$\Rightarrow$(6)
exponential stability conditions by Routh criterion

Self-study



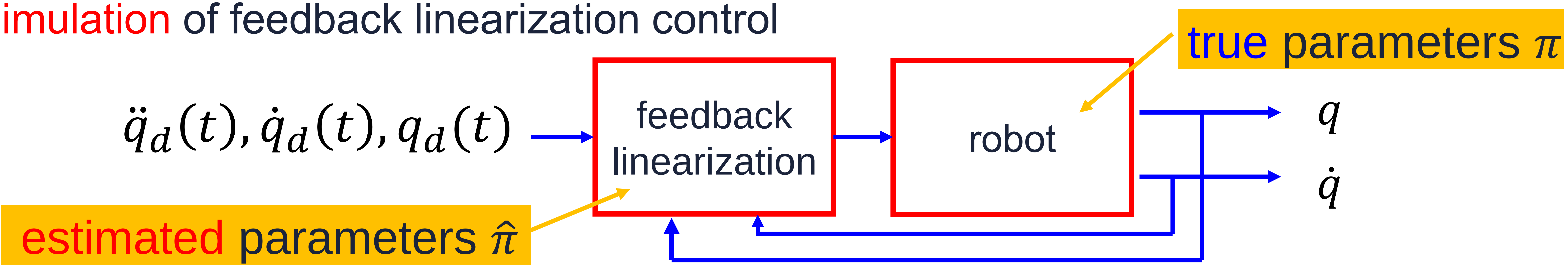
Remarks

- desired joint trajectory can be generated from Cartesian data $\ddot{p}_d(t), \dot{p}_d(0), p_d(0)$



$$q_d(0) = f^{-1}(p_d(0))$$
$$\dot{q}_d(0) = J^{-1}(q_d(0))\dot{p}_d(0)$$
$$\ddot{q}_d(t) = J^{-1}(q_d)[\ddot{p}_d(t) - \dot{J}(q_d)\dot{q}_d]$$

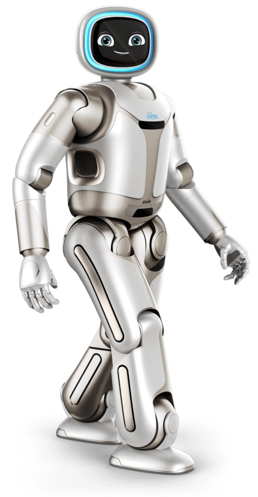
- real-time computation by Newton-Euler algo: $u_{FBL} = \widehat{NE}_\alpha(q, \dot{q}, a)$
- simulation of feedback linearization control



true parameters π

estimated parameters $\hat{\pi}$

Hint: there is no use in simulating this control law in ideal case ($\hat{\pi} = \pi$); robot behavior will be identical to the linear and decoupled case of stabilized double integrators!!

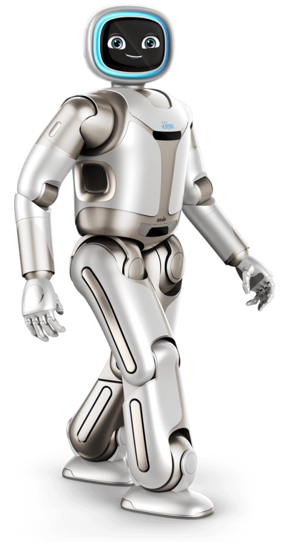


Further Comments

- choice of the diagonal elements of K_P, K_D (and K_I)
  - for shaping the error transients, with an eye to motor saturations...

$$e(t) = q_d(t) - q(t)$$

- parametric identification
  - to be done in advance, using the property of linearity in the dynamic coefficients of the robot dynamic model
- choice of the sampling time of a digital implementation
  - compromise between computational time and tracking accuracy, typically $T_c = 0.5 \div 10$ ms
- exact linearization by (state) feedback is a general technique of nonlinear control theory
  - can be used for robots with elastic joints, wheeled mobile robots, ...
  - non-robotics applications: satellites, induction motors, helicopters, ...



Another Example

Another example of feedback linearization design

- dynamic model of robots with elastic joints
  - q = link position
  - θ = motor position (after reduction gears)
  - B_m = diagonal matrix (> 0) of inertia of the (balanced) motors
  - K = diagonal matrix (> 0) of (finite) stiffness of the joints

$2N$ generalized coordinates (q, θ)

4N state variables $x = (q, \theta, \dot{q}, \dot{\theta})$

$$\begin{cases} M(q)\ddot{q} + c(q, \dot{q}) + g(q) + K(q - \theta) = 0 \\ B_m\ddot{\theta} + K(\theta - q) = u \end{cases}$$

- is there a control law that achieves exact linearization via feedback?

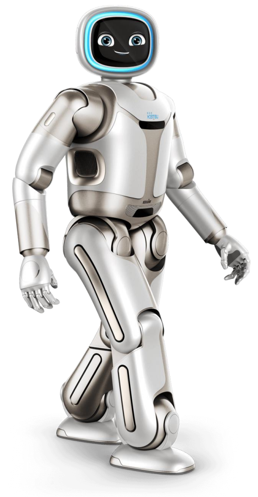
$$u = \alpha(q, \theta, \dot{q}, \dot{\theta}) + \beta(q, \theta, \dot{q}, \dot{\theta})a$$

YES

and it yields $$\frac{d^4 q_i}{dt^4} = a_i, \quad i = 1, \dots, N$$

linear and decoupled system: N chains of 4 integrators (to be stabilized by linear control design)

Hint: differentiate (1) *w.r.t. time until motor acceleration* $\ddot{\theta}$ *appears; substitute this from* (2)*; choose* u *so as to cancel all nonlinearities …*



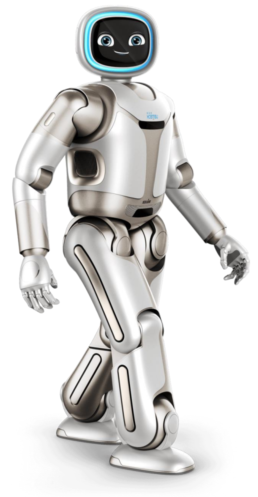
Alternative Controller

$$u = M(q)\ddot{q}_d + S(q,\dot{q})\dot{q}_d + g(q) + F_V\dot{q}_d + K_P e + K_D\dot{e}$$

SPECIAL factorization such that
$\dot{M} - 2S$ is skew-symmetric

symmetric and positive definite
matrices

- global asymptotic stability of $(e \quad \dot{e}) = (0 \quad 0)$ (trajectory tracking)
- proven by Lyapunov+Barbalat+LaSalle
- does not produce a complete cancellation of nonlinearities
  - the $\dot{q}$ and $\ddot{q}$ that appear linearly in the model are evaluated on the desired trajectory
- does not induce a linear and decoupled behavior of the trajectory error $e(t) = q_d(t) - q(t)$ in the closed-loop system
- lends itself more easily to an adaptive version
- cannot be computed directly by the standard NE algorithm...



Analysis

Analysis of asymptotic stability
(of the trajectory error - 1)

$$M(q)\ddot{q} + S(q,\dot{q})\dot{q} + g(q) + F_V\dot{q} = u$$ robot dynamics (including friction)

control law $$u = M(q)\ddot{q}_d + S(q,\dot{q})\dot{q}_d + g(q) + F_V\dot{q}_d + K_P e + K_D\dot{e}$$

- Lyapunov candidate and its time derivative

$$V = \frac{1}{2}\dot{e}^T M(q)\dot{e} + \frac{1}{2}e^T K_P e \geq 0 \Rightarrow \dot{V} = \frac{1}{2}\dot{e}^T \dot{M}(q)\dot{e} + \dot{e}^T M(q)\ddot{e} + e^T K_P\dot{e}$$

- the closed-loop system equations yield

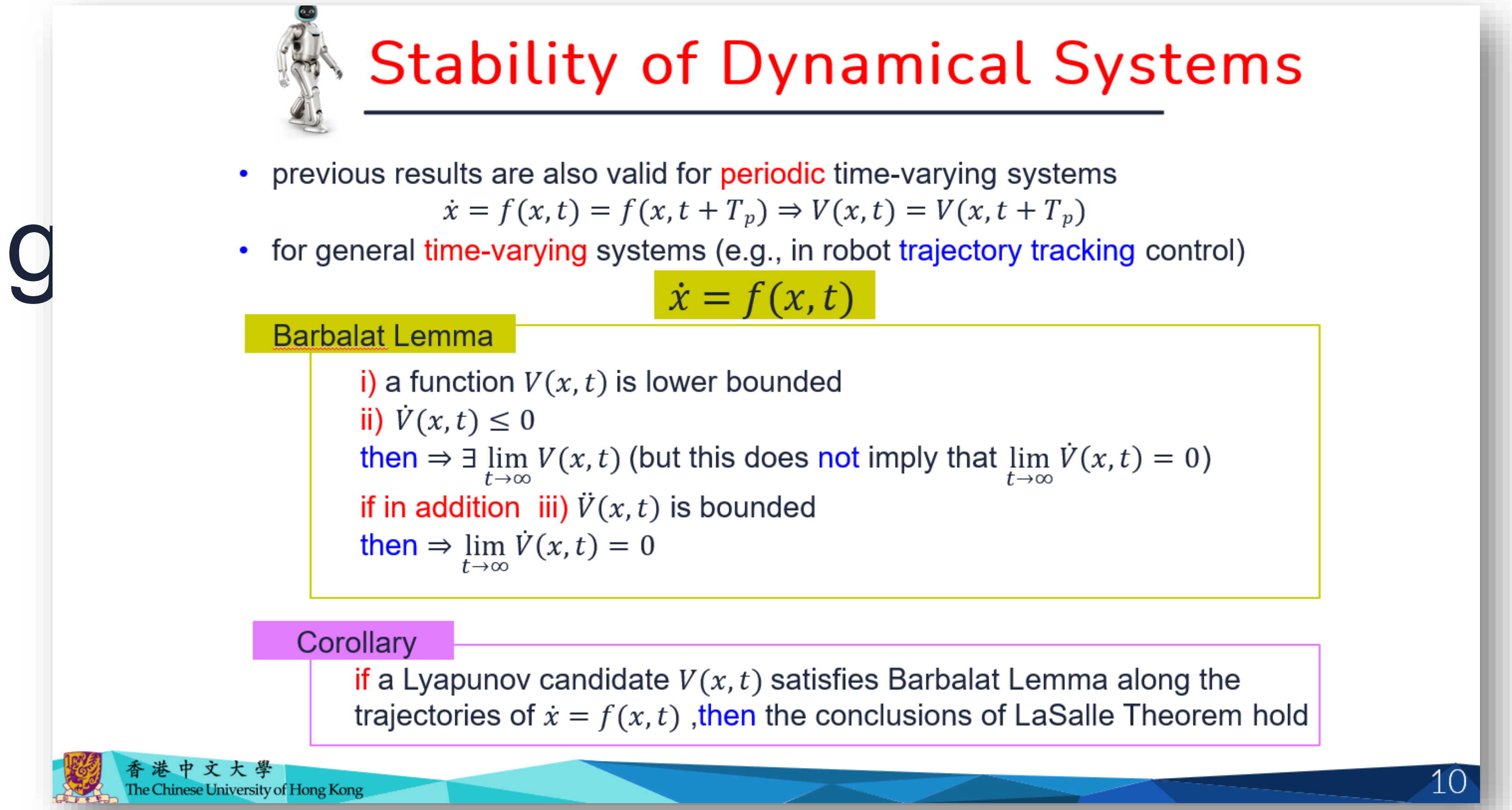
$$M(q)\ddot{e} = -S(q,\dot{q})\dot{e} - (K_D + F_V)\dot{e} - K_P e$$

- substituting and using the skew-symmetric property

$$\dot{V} = -\dot{e}^T(K_D + F_V)\dot{e} \leq 0 \quad \dot{V} = 0 \Leftrightarrow \dot{e} = 0$$

- since the system is time-varying (due to $q_d(t)$), direct applying LaSalle theorem is NOT allowed ⇒ use Barbalat lemma…

$$q = q_d(t) - e, \dot{q} = \dot{q}_d(t) - \dot{e} \Rightarrow V = \underbrace{V(e,\dot{e},t)}_{\text{error state } x} = V(x,t)$$



Stability of Dynamical Systems

- previous results are also valid for periodic time-varying systems

$$\dot{x} = f(x,t) = f(x,t+T_p) \Rightarrow V(x,t) = V(x,t+T_p)$$

- for general time-varying systems (e.g., in robot trajectory tracking control)

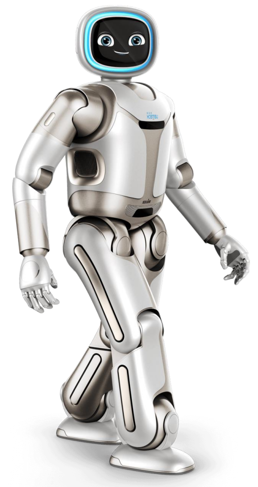
$$\dot{x} = f(x,t)$$

Barbalat Lemma

i) a function $V(x,t)$ is lower bounded
ii) $\dot{V}(x,t) \leq 0$
then ⇒ $\exists \lim_{t\to\infty} V(x,t)$ (but this does not imply that $\lim_{t\to\infty} \dot{V}(x,t) = 0$)
if in addition iii) $\ddot{V}(x,t)$ is bounded
then ⇒ $\lim_{t\to\infty} \dot{V}(x,t) = 0$

Corollary

if a Lyapunov candidate $V(x,t)$ satisfies Barbalat Lemma along the trajectories of $\dot{x} = f(x,t)$,then the conclusions of LaSalle Theorem hold



Analysis

Analysis of asymptotic stability (of the trajectory error - 2)

- since i) V is lower bounded and ii) $\dot{V} \leq 0$,we can check condition iii) in order to apply Barbalat lemma

$$\ddot{V} = -2\dot{e}^T (K_D + F_V)\ddot{e}$$ ... is this bounded?

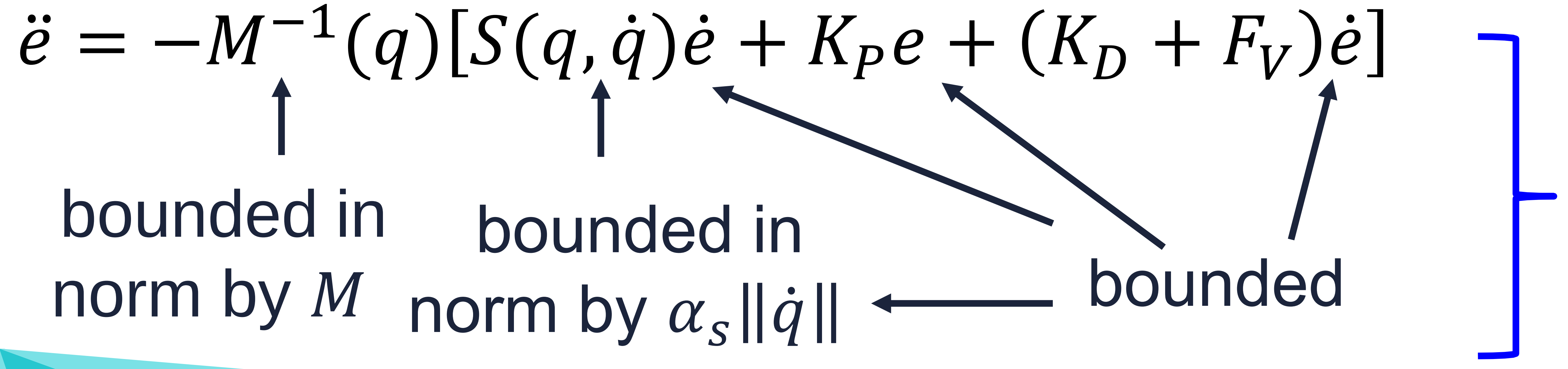
- from i) + ii), V is bounded ⇒ e and $\dot{e}$ are bounded
- assume that the desired trajectory has bounded velocity $\dot{q}$,

⇒ $\dot{q}$ is bounded

- using the following two properties of dynamic model terms

$$0 < m \leq \|M^{-1}(q)\| \leq M < \infty \qquad \|S(q,\dot{q})\| \leq \alpha_S \|\dot{q}\|$$

then also $\ddot{e}$ will be bounded (in norm) since



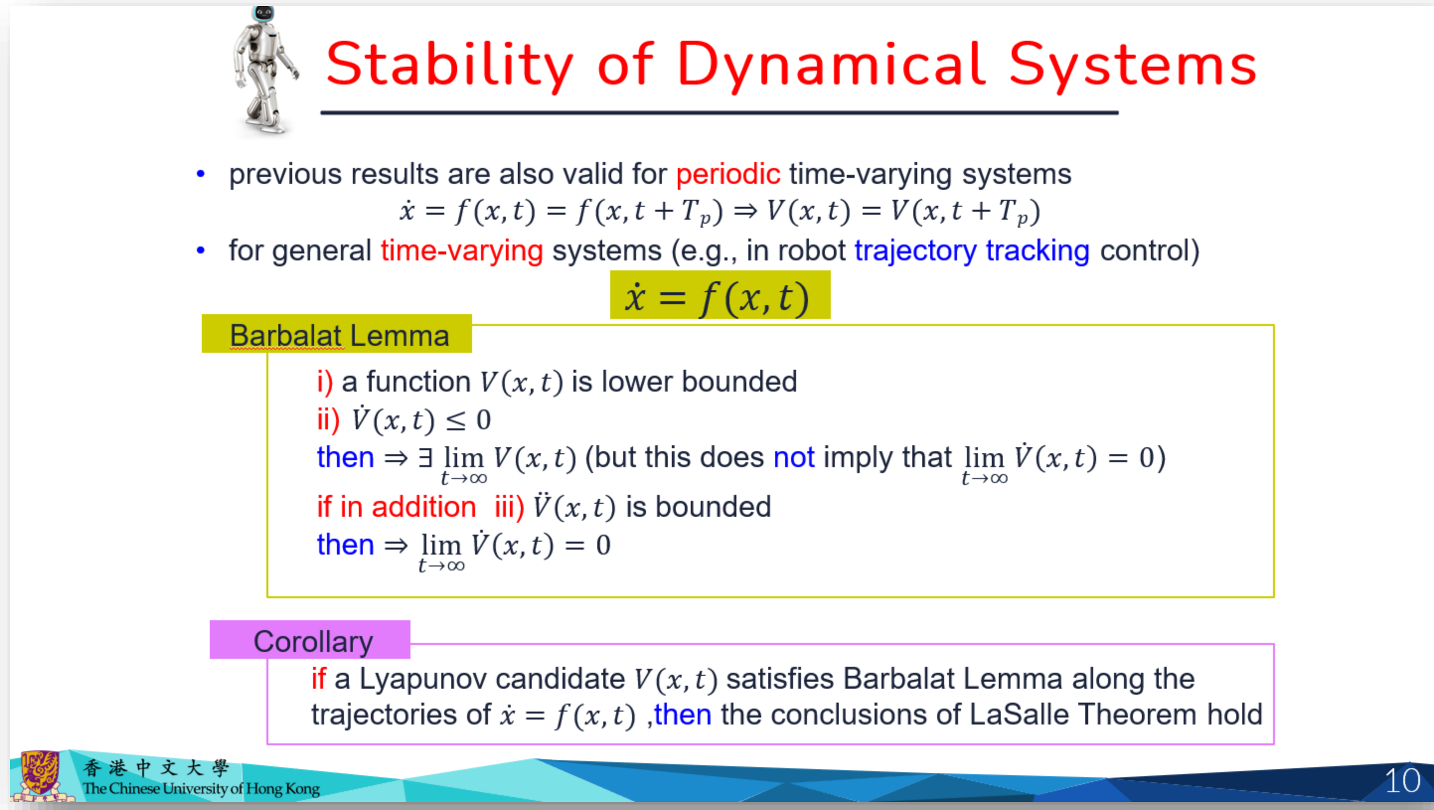
$$\ddot{e} = -M^{-1}(q)[S(q,\dot{q})\dot{e} + K_P e + (K_D + F_V)\dot{e}]$$

bounded in norm by M

bounded in norm by $\alpha_s \|\dot{q}\|$

bounded

$$\Rightarrow \lim_{t\to\infty} \dot{V}(t) = 0$$



Stability of Dynamical Systems

- previous results are also valid for periodic time-varying systems

$$\dot{x} = f(x,t) = f(x, t + T_p) \Rightarrow V(x,t) = V(x, t + T_p)$$

- for general time-varying systems (e.g., in robot trajectory tracking control)

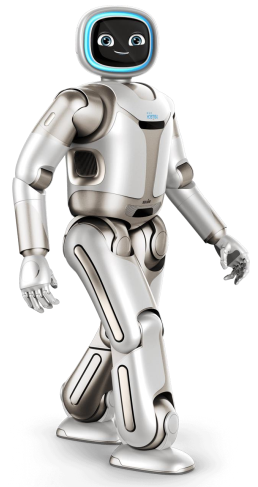
$$\dot{x} = f(x,t)$$

Barbalat Lemma

i) a function $V(x,t)$ is lower bounded
ii) $\dot{V}(x,t) \leq 0$
then ⇒ $\exists \lim_{t\to\infty} V(x,t)$ (but this does not imply that $\lim_{t\to\infty} \dot{V}(x,t) = 0$)
if in addition iii) $\ddot{V}(x,t)$ is bounded
then ⇒ $\lim_{t\to\infty} \dot{V}(x,t) = 0$

Corollary

if a Lyapunov candidate $V(x,t)$ satisfies Barbalat Lemma along the trajectories of $\dot{x} = f(x,t)$,then the conclusions of LaSalle Theorem hold



Analysis

Analysis of asymptotic stability (of the trajectory error – end of proof)

- we can now conclude by proceeding as in LaSalle theorem

$$\dot{V} = 0 \Leftrightarrow \dot{e} = 0$$

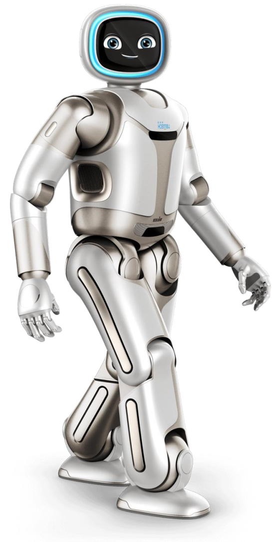
- the closed-loop dynamics in this situation is

$$M(q)\ddot{e} = -K_P e$$

$$\Longrightarrow \ \ddot{e} = 0 \Leftrightarrow e = 0$$ ➡ $$(e, \dot{e}) = (0, 0)$$

is the largest invariant set in $\dot{V} = 0$

➡ (global) asymptotic tracking will be achieved ◁



Comments

Regulation as a special case

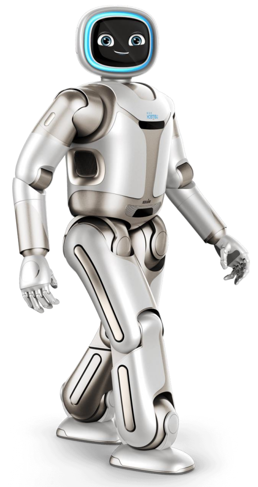
- what happens to the control laws designed for trajectory tracking when q_d is constant? are there simplifications?
- feedback linearization

$$u = M(q)[K_P(q_d - q) - K_D\dot{q}] + c(q,\dot{q}) + g(q)$$

  - no special simplifications
  - however, this is a solution to the regulation problem with exponential stability (and decoupled transients at each joint!)
- alternative global controller

$$u = K_P(q_d - q) - K_D\dot{q} + g(q)$$

  - we recover the PD + gravity cancellation control law!!

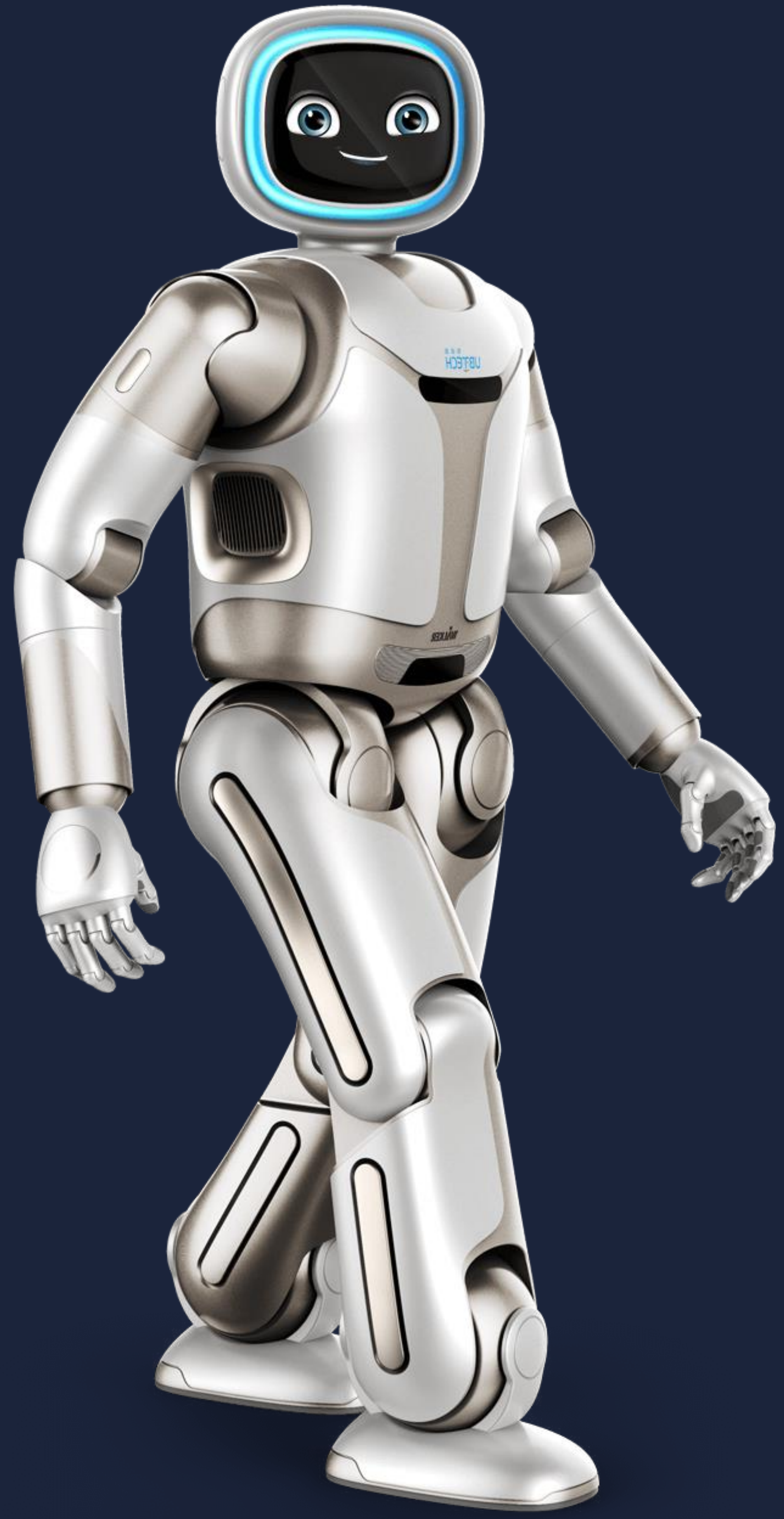


Without a model

Trajectory execution without a model

- is it possible to accurately reproduce a desired smooth joint - space reference trajectory with reduced or no information on the robot dynamic model?
- this is feasible in case of repetitive motion tasks over a finite interval of time
  - trials are performed iteratively, storing the trajectory error information of the current execution [k-th iteration] and processing it off line before the next trial [$(k+1)$ $-$iteration] starts
  - the robot should be reinitialized in the same initial position at the beginning of each trial
  - the control law is made of a non-model based part (typically, a decentralized PD law) + a time-varying feedforward which is updated at every trial
- this scheme is called iterative trajectory learning

https://ieeexplore.ieee.org/document/240467/authors#authors



Q&A