On the Feedback Linearization of Robots with Variable Joint Stiffness *and Other Stories...*

Claudio Melchiorri
Dip. di Ingegneria dell'Energia Elettrica e dell'Informazione "G. Marconi"
Alma Mater Studiorum - Università di Bologna

Other stories ...

ADL - Unibo

Cooperation with ADL started some time at the end of the 80's...

- NATO Robotics workshop, Il Ciocco, July 4-6, 1988 (??)
- Italian workshop on Robotics, Rome, March 30, 1990

ADL - Unibo

INCONTRO DI STUDIO ROBOTICA
ROMA, 30 MARZO 1990

Programma di massima:

1° Parte: Roma

11:30 C. Manes, ``Controllo ibrido"
12:00 L. Lanari, ``Modellistica e controllo di robot con bracci flessibili"
12:30 G. Oriolo ``Algoritmi risolutivi della ridondanza"

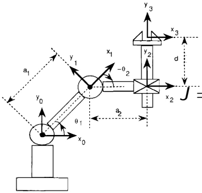
2° Parte: Bologna

14:00 R. Zanasi, ``Controllo Binario: simulazione e progettazione assistita"
14:30 A. Tonielli, C. Rossi, ``Controllo Sliding Mode di motori in corrente alternata con pilotaggio diretto dell'inverter"
15:00 C. Melchiorri, ``Considerazioni sull'uso di criteri di norma minima nella soluzione di problemi cinematici".

3° Parte: Napoli

15:30 P. Chiacchio, ``Analisi cinetostatica di bracci cooperanti"
16:00 B. Siciliano, "Modellistica e controllo di bracci con elementi flessibili"
16:30 S. Chiaverini, ``Approccio parallelo al controllo di forza e posizione"

ADL - Unibo



$$J = \begin{bmatrix} -a_1 s_1 - a_2 s_{12} - d c_{12} & -a_2 s_{12} - d c_{12} & -s_{12} \\ a_1 c_1 + a_2 c_{12} - d s_{12} & a_2 c_{12} - d s_{12} & c_{12} \\ 0 & 0 & 0 \end{bmatrix} \quad \begin{cases} \theta_1 = 20\ [deg] \\ \theta_2 = 20\ [deg] \\ d = 0.5\ [m] = 50\ [cm] \end{cases}$$

$$v_{[m]} = [2, 4, 0]^T\ [m/s] \quad = \quad [200, 400, 0]^T\ [cm/s] = v_{[cm]}, \qquad q = J^+ v = ??$$

$$J^+_{[m]} = \begin{bmatrix} 0.257 & 0.783 & 0.0 \\ -1.613 & -1.533 & 0.0 \\ 0.472 & 0.780 & 0.0 \end{bmatrix} \quad q_{[m]} = J^+_{[m]} v_{[m]} = \begin{bmatrix} 3.6451\ [1/s] \\ -9.358\ [1/s] \\ 4.066\ [m/s] \end{bmatrix} \quad v'_{[m]} = J\, q_{[m]} = \begin{bmatrix} 2\ [m/s] \\ 4\ [m/s] \\ 0\ [m/s] \end{bmatrix}$$

$$J^+_{[cm]} = \begin{bmatrix} 0.0055 & 0.0126 & 0.0 \\ -0.0171 & -0.0168 & 0.0 \\ -0.0001 & 0.0001 & 0.0 \end{bmatrix} \quad q_{[cm]} = J^+_{[cm]} v_{[cm]} = \begin{bmatrix} 6.149\ [1/s] \\ -10.149\ [1/s] \\ 0.058\ [cm/s] \end{bmatrix} \quad v'_{[cm]} = J\, q_{[cm]} = \begin{bmatrix} 200\ [cm/s] \\ 400\ [cm/s] \\ 0\ [cm/s] \end{bmatrix}$$

ADL - Unibo

Invariant properties of hybrid position/force control:

- A. De Luca, C. Manes, and F. Nicolò, 1988, "A task space decoupling approach to hybrid control of manipulators", 2nd IFAC Symp. on Robot Control, SYROCO'88, pp. 54.1-54.6, Karlsruhe, F.R.G.

ELSEVIER

IFAC Proceedings Volumes

Volume 21, Issue 16, October 1988, Pages 157-162

A Task Space Decoupling Approach to Hybrid Control of Manipulators

A. De Luca, C. Manes, F. Nicolò

⊞ Show more

https://doi.org/10.1016/S1474-6670(17)54603-0

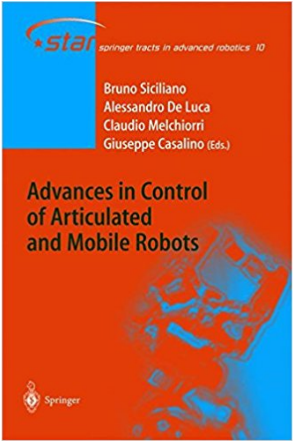
Abstract

A scheme for dynamic hybrid control of robot manipulators is presented. The design is directly achieved in task space coordinates. In this way, the inherent orthogonality between force and velocity description of tasks is preserved and overspecification is avoided in the control synthesis. A nonlinear decoupling and linearizing feedback law is obtained which yields invariant control performances over time-varying tasks. The effect of a robustifying integral action is discussed. Simulation results are reported for a two-link arm.

ADL - Unibo

Research projects:

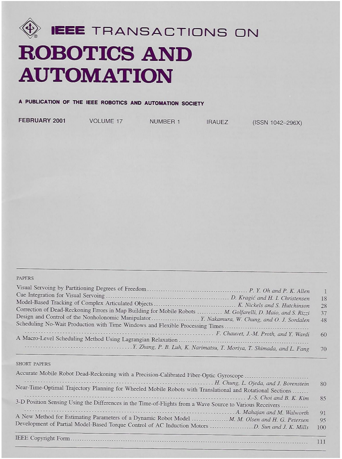
- "Development of an Integrated Mobile Manipulator for Planetary Exploration Tasks", ASI (1998-1999)
- "RAMSETE: Articulated and Mobile Robotics for SErvice and TEchnology", MURST (1999-2000)
- "MISTRAL: Methodologies and Integration of Subsystems and Technologies for Anthropic Robotics and Locomotion", MIUR (2001-2002)
- "MATRICS: Methodologies Applications and Technologies for Robot Interaction Cooperation and Supervision", MIUR (2003-2004)
- "SICURA: Safe Physical Interaction between Robots and Humans", MIUR (2008-2010)



ADL - Unibo

IEEE:

- Editorial Board, IEEE Transactions on Robotics and Automation
- ICRA 2007 in Rome



IEEE TRANSACTIONS ON
ROBOTICS AND AUTOMATION

A PUBLICATION OF THE IEEE ROBOTICS AND AUTOMATION SOCIETY

FEBRUARY 2001 VOLUME 17 NUMBER 1 IRAUEZ (ISSN 1042–296X)

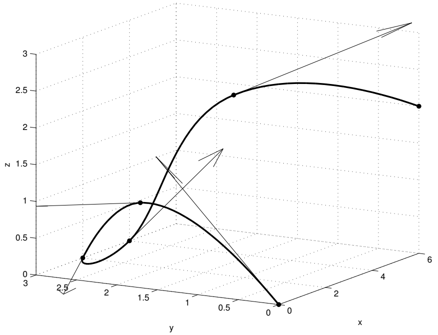
PAPERS

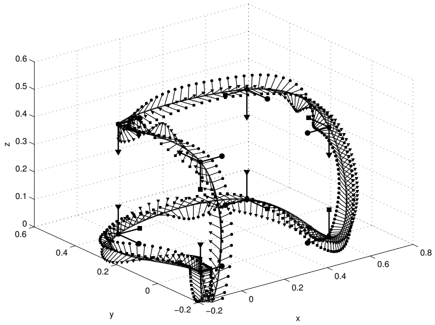
SHORT PAPERS

ADL - Unibo

Trajectory planning:

- A. De Luca, L. Lanari, G. Oriolo, “Generation and computation of optimal smooth trajectories for robot arms” Rapporto DIS 13.88, Università di Roma La Sapienza, Sep. 1988.
- A. De Luca, “A spline trajectory generator for robot arms,” Report RAL 68, Rensselaer Polytechnic Institute, Apr. 1986.

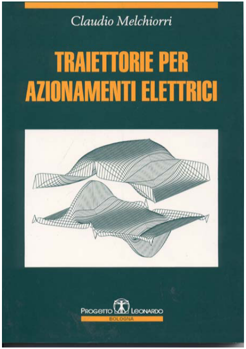


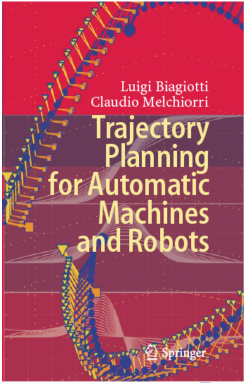


ADL - Unibo

Trajectory planning:

- A. De Luca, L. Lanari, G. Oriolo, "Generation and computation of optimal smooth trajectories for robot arms" Rapporto DIS 13.88, Università di Roma La Sapienza, Sep. 1988.
- A. De Luca, "A spline trajectory generator for robot arms," Report RAL 68, Rensselaer Polytechnic Institute, Apr. 1986.





ADL - Unibo

Il tempo totale di percorrenza della spline è dato da

$$T = \sum_{k=1}^{n-1} T_k = t_n - t_1$$

È possibile impostare un problema di ottimo che minimizza il tempo di percorrenza. Si tratta di calcolare i valori T_k in modo da minimizzare T, con i vincoli sulle massime velocità ed accelerazioni ai giunti.

Formalmente, il problema si imposta come

$$\begin{cases} \min_{T_k} \quad T = \sum_{k=1}^{n-1} T_k & \\ & |\dot{q}(\tau, T_k)| < v_{max} \qquad \tau \in [0\, T] \\ \text{tale che} & \\ & |\ddot{q}(\tau, T_k)| < a_{max} \qquad \tau \in [0\, T] \end{cases}$$

ed è quindi un problema di ottimo non lineare con funzione obiettivo lineare, risolvibile con tecniche classiche della ricerca operativa.

ADL - Unibo

Si vuole determinare la spline passante per i punti $q_1 = 0$, $q_2 = 2$, $q_3 = 12$, $q_4 = 5$, *che minimizza il tempo di percorrenza totale* T *rispettando i seguenti vincoli:* $v_{max} = 3$, $a_{max} = 2$.

Occorre risolvere il seguente problema di ottimizzazione non lineare:

$$min \quad T = T_1 + T_2 + T_3$$

soggetto ai vincoli riportati di seguito.

Risolvendo questo problema (p.e. con Optimization Toolbox *di MATLAB) si ottengono i seguenti valori:*

$$T_1 = 1.5549, \qquad T_2 = 4.4451, \qquad T_3 = 4.5826, \qquad \Rightarrow \qquad T = 10.5826 \ sec$$

ADL - Unibo

Vincoli del problema di ottimizzazione:

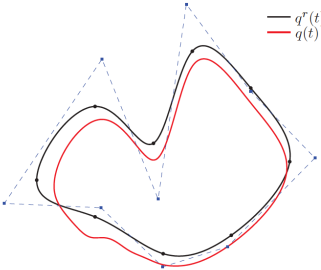
$$
\left\{
\begin{array}{lllllll}
a_{11} & & & & & \leq & v_{max} \quad \textit{(velocità iniziale del I tratto} \leq v_{max}\textit{)} \\
a_{21} & & & & & \leq & v_{max} \quad \textit{(velocità iniziale del II tratto} \leq v_{max}\textit{)} \\
a_{31} & & & & & \leq & v_{max} \quad \textit{(velocità iniziale del III tratto} \leq v_{max}\textit{)} \\
a_{11} & + & 2a_{12}\,T_1 & + & 3a_{13}\,T_1^2 & \leq & v_{max} \quad \textit{(velocità finale del I tratto} \leq v_{max}\textit{)} \\
a_{21} & + & 2a_{22}\,T_2 & + & 3a_{23}\,T_2^2 & \leq & v_{max} \quad \textit{(velocità finale del II tratto} \leq v_{max}\textit{)} \\
a_{31} & + & 2a_{32}\,T_3 & + & 3a_{33}\,T_3^2 & \leq & v_{max} \quad \textit{(velocità finale del III tratto} \leq v_{max}\textit{)} \\
a_{11} & + & 2a_{12}\left(-\frac{a_{12}}{3a_{13}}\right) & + & 3a_{13}\left(-\frac{a_{12}}{3a_{13}}\right)^2 & \leq & v_{max} \quad \textit{(velocità all'interno del I tratto} \leq v_{max}\textit{)} \\
a_{21} & + & 2a_{22}\left(-\frac{a_{22}}{3a_{23}}\right) & + & 3a_{23}\left(-\frac{a_{22}}{3a_{23}}\right)^2 & \leq & v_{max} \quad \textit{(velocità all'interno del II tratto} \leq v_{max}\textit{)} \\
a_{31} & + & 2a_{32}\left(-\frac{a_{32}}{3a_{33}}\right) & + & 3a_{33}\left(-\frac{a_{32}}{3a_{33}}\right)^2 & \leq & v_{max} \quad \textit{(velocità all'interno del III tratto} \leq v_{max}\textit{)} \\
2a_{12} & & & & & \leq & a_{max} \quad \textit{(accelerazione iniziale del I tratto} \leq a_{max}\textit{)} \\
2a_{22} & & & & & \leq & a_{max} \quad \textit{(accelerazione iniziale del II tratto} \leq a_{max}\textit{)} \\
2a_{32} & & & & & \leq & a_{max} \quad \textit{(accelerazione iniziale del III tratto} \leq a_{max}\textit{)} \\
2a_{12} & & & + & 6a_{13}\,T_1 & \leq & a_{max} \quad \textit{(accelerazione finale del I tratto} \leq a_{max}\textit{)} \\
2a_{22} & & & + & 6a_{23}\,T_2 & \leq & a_{max} \quad \textit{(accelerazione finale del II tratto} \leq a_{max}\textit{)} \\
2a_{32} & & & + & 6a_{33}\,T_3 & \leq & a_{max} \quad \textit{(accelerazione finale del III tratto} \leq a_{max}\textit{)}
\end{array}
\right.
$$

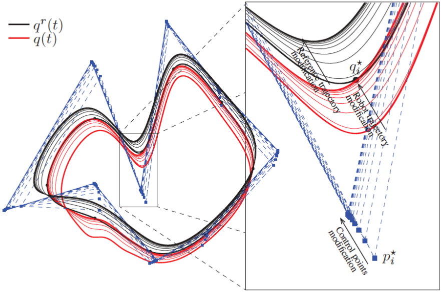
ADL - Unibo

Sloshing suppression - movies

ADL - Unibo

Repetitive control & Splines:





ADL - Unibo

ADL - Unibo

ADL - Unibo

ADL - Unibo

ADL Surprise Festschrift

Thanks Alex!!

Dynamic Model of Robots with Variable Joint Stiffness

On the Feedback Linearization of Robots with Variable Joint Stiffness

G. Palli[1], C. Melchiorri[1], A. De Luca[2]

[1] Dip. di Ingegneria dell'Energia Elettrica e dell'Informazione "G. Marconi"
Alma Mater Studiorum - Università di Bologna

[2] Dip. di Ingegneria Informatica Automatica e Gestionale "A. Ruberti"
Università di Roma "La Sapienza"

Dynamic Model of Robots with Variable Joint Stiffness

- Robot (+actuators) dynamic equations

$$\begin{aligned} M(q)\,\ddot{q} + N(q,\dot{q}) + K\,(q-\theta) &= 0 \\ B\,\ddot{\theta} + K\,(\theta - q) &= \tau \end{aligned}$$

- The diagonal joint stiffness matrix is considered time-variant

$$K = \mathrm{diag}\{k_1, \ldots, k_n\}, \quad K = K(t) > 0$$

- Alternative notation

$$K\,(q-\theta) = \Phi k, \quad \Phi = \mathrm{diag}\{(q_1-\theta_1), \ldots, (q_n-\theta_n)\}, \quad k = [k_1, \ldots, k_n]^T$$

1. The joint stiffness k can be directly changed by means of a (suitably scaled) additional command τ_k

$$k = \tau_k$$

2. Alternatively, the variation of joint stiffness may be modeled as a second-order dynamic system

$$\ddot{k} = \phi(x, k, \dot{k}, \tau_k)$$

Dynamic Model of Robots with Variable Joint Stiffness

- Robot (+actuators) dynamic equations

$$\begin{aligned} M(q)\,\ddot{q} + N(q,\dot{q}) + K\,(q-\theta) &= 0 \\ B\,\ddot{\theta} + K\,(\theta - q) &= \tau \end{aligned}$$

- The diagonal joint stiffness matrix is considered time-variant

$$K = \text{diag}\{k_1, \ldots, k_n\}, \quad K = K(t) > 0$$

- Alternative notation

$$K\,(q-\theta) = \Phi k, \quad \Phi = \text{diag}\{(q_1-\theta_1), \ldots, (q_n-\theta_n)\}, \quad k = [k_1, \ldots, k_n]^T$$

1. The joint stiffness k can be directly changed by means of a (suitably scaled) additional command τ_k

$$k = \tau_k$$

2. Alternatively, the variation of joint stiffness may be modeled as a second-order dynamic system

$$\ddot{k} = \phi(x, k, \dot{k}, \tau_k)$$

Dynamic Model of Robots with Variable Joint Stiffness

- Robot (+actuators) dynamic equations

$$
\begin{aligned}
M(q)\,\ddot{q} + N(q,\dot{q}) + K\,(q-\theta) &= 0 \\
B\,\ddot{\theta} + K\,(\theta - q) &= \tau
\end{aligned}
$$

- The diagonal joint stiffness matrix is considered time-variant

$$K = \text{diag}\{k_1,\,\ldots,\,k_n\}, \quad K = K(t) > 0$$

- Alternative notation

$$K\,(q-\theta) = \Phi k, \quad \Phi = \text{diag}\{(q_1-\theta_1),\,\ldots,(q_n-\theta_n)\}, \quad k = [k_1,\,\ldots,\,k_n]^T$$

1. The joint stiffness k can be directly changed by means of a (suitably scaled) additional command τ_k

$$k = \tau_k$$

2. Alternatively, the variation of joint stiffness may be modeled as a second-order dynamic system

$$\ddot{k} = \phi(x, k, \dot{k}, \tau_k)$$

Dynamic Model of Robots with Variable Joint Stiffness

- The input u and the robot state x are:

$$u = \begin{bmatrix} \tau \\ \tau_k \end{bmatrix} \in \mathbb{R}^{2n}, \quad x = \begin{bmatrix} q^T & \dot{q}^T & \theta^T & \dot{\theta}^T \end{bmatrix}^T \in \mathbb{R}^{4n}$$

- In the case of second-order stiffness variation model, the state vector of the robot becomes:

$$x_e = \begin{bmatrix} q^T & \dot{q}^T & \theta^T & \dot{\theta}^T & k^T & \dot{k}^T \end{bmatrix}^T \in \mathbb{R}^{6n}$$

- In any case, the goal will be to simultaneously control the following set of outputs

$$y = \begin{bmatrix} q \\ k \end{bmatrix} \in \mathbb{R}^{2n}$$

namely the link positions (and thus, through the robot direct kinematics, the end-effector pose) and the joint stiffness

Dynamic Model of Robots with Variable Joint Stiffness

- The input u and the robot state x are:

$$u = \begin{bmatrix} \tau \\ \tau_k \end{bmatrix} \in \mathbb{R}^{2n}, \quad x = \begin{bmatrix} q^T & \dot{q}^T & \theta^T & \dot{\theta}^T \end{bmatrix}^T \in \mathbb{R}^{4n}$$

- In the case of second-order stiffness variation model, the state vector of the robot becomes:

$$x_e = \begin{bmatrix} q^T & \dot{q}^T & \theta^T & \dot{\theta}^T & k^T & \dot{k}^T \end{bmatrix}^T \in \mathbb{R}^{6n}$$

- In any case, the goal will be to simultaneously control the following set of outputs

$$y = \begin{bmatrix} q \\ k \end{bmatrix} \in \mathbb{R}^{2n}$$

namely the link positions (and thus, through the robot direct kinematics, the end-effector pose) and the joint stiffness

Dynamic Model of Robots with Variable Joint Stiffness

- The input u and the robot state x are:

$$u = \begin{bmatrix} \tau \\ \tau_k \end{bmatrix} \in \mathbb{R}^{2n}, \quad x = \begin{bmatrix} q^T & \dot{q}^T & \theta^T & \dot{\theta}^T \end{bmatrix}^T \in \mathbb{R}^{4n}$$

- In the case of second-order stiffness variation model, the state vector of the robot becomes:

$$x_e = \begin{bmatrix} q^T & \dot{q}^T & \theta^T & \dot{\theta}^T & k^T & \dot{k}^T \end{bmatrix}^T \in \mathbb{R}^{6n}$$

- In any case, the goal will be to simultaneously control the following set of outputs

$$y = \begin{bmatrix} q \\ k \end{bmatrix} \in \mathbb{R}^{2n}$$

namely the link positions (and thus, through the robot direct kinematics, the end-effector pose) and the joint stiffness

Dynamic Inversion

- The motion is specified in terms of a desired smooth position trajectory $q = q_d(t)$ and joint stiffness matrix $K = K_d(t)$ (or, equivalently, of the vector $k = k_d(t)$)
- Assuming $k = \tau_k$, we have simply $\tau_{k,d} = k_d(t)$ and only the computation of the nominal motor torque τ_d is of actual interest
- The robot dynamic equation is differentiated twice with respect to time

$$M(q)\,q^{[3]} + \dot{M}(q)\,\ddot{q} + \dot{N}(q,\dot{q}) + \dot{K}\,(q-\theta) + K\,(\dot{q}-\dot{\theta}) = 0$$

and

$$M(q)\,q^{[4]} + 2\,\dot{M}(q)\,q^{[3]} + \ddot{M}(q)\,\ddot{q} + \ddot{N}(q,\dot{q}) + \\ + K\,(\ddot{q}-\ddot{\theta}) + 2\,\dot{K}\,(\dot{q}-\dot{\theta}) + \ddot{K}\,(q-\theta) = 0$$

Dynamic Inversion

- The motion is specified in terms of a desired smooth position trajectory $q = q_d(t)$ and joint stiffness matrix $K = K_d(t)$ (or, equivalently, of the vector $k = k_d(t)$)
- Assuming $k = \tau_k$, we have simply $\tau_{k,d} = k_d(t)$ and only the computation of the nominal motor torque τ_d is of actual interest
- The robot dynamic equation is differentiated twice with respect to time

$$M(q)\, q^{[3]} + \dot{M}(q)\, \ddot{q} + \dot{N}(q, \dot{q}) + \dot{K}\,(q - \theta) + K\,(\dot{q} - \dot{\theta}) = 0$$

and

$$M(q)\, q^{[4]} + 2\,\dot{M}(q)\, q^{[3]} + \ddot{M}(q)\, \ddot{q} + \ddot{N}(q, \dot{q}) + \\ + K\,(\ddot{q} - \ddot{\theta}) + 2\,\dot{K}\,(\dot{q} - \dot{\theta}) + \ddot{K}\,(q - \theta) = 0$$

Dynamic Inversion

- The motion is specified in terms of a desired smooth position trajectory $q = q_d(t)$ and joint stiffness matrix $K = K_d(t)$ (or, equivalently, of the vector $k = k_d(t)$)
- Assuming $k = \tau_k$, we have simply $\tau_{k,d} = k_d(t)$ and only the computation of the nominal motor torque τ_d is of actual interest
- The robot dynamic equation is differentiated twice with respect to time

$$M(q)\, q^{[3]} + \dot{M}(q)\, \ddot{q} + \dot{N}(q, \dot{q}) + \dot{K}\,(q - \theta) + K\,(\dot{q} - \dot{\theta}) = 0$$

and

$$\begin{aligned} M(q)\, q^{[4]} + 2\, \dot{M}(q)\, q^{[3]} + \ddot{M}(q)\, \ddot{q} + \ddot{N}(q, \dot{q}) + \\ + K\,(\ddot{q} - \ddot{\theta}) + 2\, \dot{K}\,(\dot{q} - \dot{\theta}) + \ddot{K}\,(q - \theta) = 0 \end{aligned}$$

Dynamic Inversion

- Reference motor position along the desired robot trajectory

$$\theta_d = q_d + K_d^{-1}\left(M(q_d)\ddot{q}_d + N(q_d, \dot{q}_d)\right).$$

- Reference motor velocity

$$\begin{aligned}\dot{\theta}_d \;&=\; \dot{q}_d + K_d^{-1}\Big(M(q_d)q_d^{[3]} + \dot{M}(q_d)\ddot{q}_d + \dot{N}(q_d, \dot{q}_d)\\ &\quad - \dot{K}_d K_d^{-1}(M(q_d)\ddot{q}_d + N(q_d, \dot{q}_d))\Big).\end{aligned}$$

- Actuators dynamic model inversion

$$\ddot{\theta} = B^{-1}\left[\tau - K(\theta - q)\right],$$

Actuator Torques Computation

- Reference motor torque along the desired trajectory

$$\tau_d = M(q_d)\ddot{q}_d + N(q_d, \dot{q}_d) + BK_d^{-1}\alpha_d\left(q_d, \dot{q}_d, \ddot{q}_d, q_d^{[3]}, q_d^{[4]}, k_d, \dot{k}_d, \ddot{k}_d\right)$$

- Some minimal smoothness requirements are imposed

$$q_d(t) \in \mathbb{C}^4 \quad \text{and} \quad k_d(t) \in \mathbb{C}^2$$

- Discontinuous models of friction or actuator dead-zones on the motor side can be considered without problems
- Discontinuous phenomena acting on the link side should be approximated by a smooth model
- The command torques τ_d can be kept within the saturation limits by a suitable time scaling of the manipulator trajectory

Actuator Torques Computation

- Reference motor torque along the desired trajectory

$$\tau_d = M(q_d)\ddot{q}_d + N(q_d, \dot{q}_d) + BK_d^{-1}\alpha_d\left(q_d, \dot{q}_d, \ddot{q}_d, q_d^{[3]}, q_d^{[4]}, k_d, \dot{k}_d, \ddot{k}_d\right)$$

- Some minimal smoothness requirements are imposed

$$q_d(t) \in \mathbb{C}^4 \quad \text{and} \quad k_d(t) \in \mathbb{C}^2$$

- Discontinuous models of friction or actuator dead-zones on the motor side can be considered without problems
- Discontinuous phenomena acting on the link side should be approximated by a smooth model
- The command torques τ_d can be kept within the saturation limits by a suitable time scaling of the manipulator trajectory

Actuator Torques Computation

- Reference motor torque along the desired trajectory

$$\tau_d = M(q_d)\ddot{q}_d + N(q_d, \dot{q}_d) + BK_d^{-1}\alpha_d\left(q_d, \dot{q}_d, \ddot{q}_d, q_d^{[3]}, q_d^{[4]}, k_d, \dot{k}_d, \ddot{k}_d\right)$$

- Some minimal smoothness requirements are imposed

$$q_d(t) \in \mathbb{C}^4 \quad \text{and} \quad k_d(t) \in \mathbb{C}^2$$

- Discontinuous models of friction or actuator dead-zones on the motor side can be considered without problems
- Discontinuous phenomena acting on the link side should be approximated by a smooth model
- The command torques τ_d can be kept within the saturation limits by a suitable time scaling of the manipulator trajectory

Actuator Torques Computation

- Reference motor torque along the desired trajectory

$$\tau_d = M(q_d)\ddot{q}_d + N(q_d, \dot{q}_d) + BK_d^{-1}\alpha_d\left(q_d, \dot{q}_d, \ddot{q}_d, q_d^{[3]}, q_d^{[4]}, k_d, \dot{k}_d, \ddot{k}_d\right)$$

- Some minimal smoothness requirements are imposed

$$q_d(t) \in \mathbb{C}^4 \quad \text{and} \quad k_d(t) \in \mathbb{C}^2$$

- Discontinuous models of friction or actuator dead-zones on the motor side can be considered without problems
- Discontinuous phenomena acting on the link side should be approximated by a smooth model
- The command torques τ_d can be kept within the saturation limits by a suitable time scaling of the manipulator trajectory

Second-Order Stiffness Model

- The dynamics of the joint stiffness k is written as a generic nonlinear function of the system configuration

$$\ddot{k} = \beta(q,\theta) + \gamma(q,\theta)\,\tau_k$$

- Double differentiation wrt time of the robot dynamics

$$\begin{aligned} M\,q^{[4]} + 2\,\dot{M}\,q^{[3]} + \ddot{M}\,\ddot{q} + \ddot{N} & \\ + K\,\left(\ddot{q} - B^{-1}\left[\tau - K(\theta - q)\right]\right) & \\ + 2\,\dot{K}\,(\dot{q} - \dot{\theta}) + \Phi\,(\beta + \gamma\,\tau_k) &= 0 \end{aligned}$$

where both the inputs τ and τ_k appear

Feedback Linearized Model

- The overall system can be written as

$$\begin{bmatrix} q^{[4]} \\ \ddot{k} \end{bmatrix} = \begin{bmatrix} \alpha(x_e) \\ \beta(q,\theta) \end{bmatrix} + Q(x_e) \begin{bmatrix} \tau \\ \tau_k \end{bmatrix}$$

where $Q(x_e)$ is the decoupling matrix:

$$Q(x_e) = \begin{bmatrix} M^{-1}KB^{-1} & M^{-1}\Phi\,\gamma(q,\theta) \\ 0_{n\times n} & \gamma(q,\theta) \end{bmatrix}$$

- Non-Singularity Conditions $\left.\begin{array}{l} k_i > 0 \\ \gamma_i(q_i,\theta_i) \neq 0 \end{array}\right\} \forall\, i = 1,\dots,n$

- By applying the static state feedback

$$\begin{bmatrix} \tau \\ \tau_k \end{bmatrix} = Q^{-1}(x_e)\left(-\begin{bmatrix} \alpha(x_e) \\ \beta(q,\theta) \end{bmatrix} + \begin{bmatrix} v_q \\ v_k \end{bmatrix}\right)$$

the full feedback linearized model is obtained

$$\begin{bmatrix} q^{[4]} \\ \ddot{k} \end{bmatrix} = \begin{bmatrix} v_q \\ v_k \end{bmatrix}$$

Feedback Linearized Model

- The overall system can be written as

$$\begin{bmatrix} q^{[4]} \\ \ddot{k} \end{bmatrix} = \begin{bmatrix} \alpha(x_e) \\ \beta(q,\theta) \end{bmatrix} + Q(x_e) \begin{bmatrix} \tau \\ \tau_k \end{bmatrix}$$

where $Q(x_e)$ is the decoupling matrix:

$$Q(x_e) = \begin{bmatrix} M^{-1}KB^{-1} & M^{-1}\Phi\,\gamma(q,\theta) \\ 0_{n\times n} & \gamma(q,\theta) \end{bmatrix}$$

- Non-Singularity Conditions $\left.\begin{array}{l} k_i > 0 \\ \gamma_i(q_i,\theta_i) \neq 0 \end{array}\right\} \forall\ i = 1, \ldots, n$

- By applying the static state feedback

$$\begin{bmatrix} \tau \\ \tau_k \end{bmatrix} = Q^{-1}(x_e)\left(-\begin{bmatrix} \alpha(x_e) \\ \beta(q,\theta) \end{bmatrix} + \begin{bmatrix} v_q \\ v_k \end{bmatrix}\right)$$

the full feedback linearized model is obtained

$$\begin{bmatrix} q^{[4]} \\ \ddot{k} \end{bmatrix} = \begin{bmatrix} v_q \\ v_k \end{bmatrix}$$

Feedback Linearized Model

- The overall system can be written as

$$\begin{bmatrix} q^{[4]} \\ \ddot{k} \end{bmatrix} = \begin{bmatrix} \alpha(x_e) \\ \beta(q,\theta) \end{bmatrix} + Q(x_e) \begin{bmatrix} \tau \\ \tau_k \end{bmatrix}$$

where $Q(x_e)$ is the decoupling matrix:

$$Q(x_e) = \begin{bmatrix} M^{-1}KB^{-1} & M^{-1}\Phi\,\gamma(q,\theta) \\ 0_{n\times n} & \gamma(q,\theta) \end{bmatrix}$$

- Non-Singularity Conditions $\left.\begin{array}{l} k_i > 0 \\ \gamma_i(q_i,\theta_i) \neq 0 \end{array}\right\} \forall\; i = 1, \ldots, n$

- By applying the static state feedback

$$\begin{bmatrix} \tau \\ \tau_k \end{bmatrix} = Q^{-1}(x_e)\left(- \begin{bmatrix} \alpha(x_e) \\ \beta(q,\theta) \end{bmatrix} + \begin{bmatrix} v_q \\ v_k \end{bmatrix} \right)$$

the full feedback linearized model is obtained

$$\begin{bmatrix} q^{[4]} \\ \ddot{k} \end{bmatrix} = \begin{bmatrix} v_q \\ v_k \end{bmatrix}$$

Dynamic Feedback Linearization

- Considering the very simple stiffness variation model

$$k_i = \tau_{k_i}$$

the dynamics of the system becomes:

$$\begin{bmatrix} \ddot{q} \\ k \end{bmatrix} = \begin{bmatrix} -M^{-1}N \\ 0_{n\times n} \end{bmatrix} + \begin{bmatrix} 0_{n\times n} & -M^{-1}\Phi \\ 0_{n\times n} & I_{n\times n} \end{bmatrix} \begin{bmatrix} \tau \\ \tau_k \end{bmatrix}$$

Problem

The decoupling matrix of the system is structurally singular

Solution

Dynamic extension on the input τ_k is needed

$$\ddot{\tau}_k = u_k$$

Dynamic Feedback Linearization

- Considering the very simple stiffness variation model

$$k_i = \tau_{k_i}$$

the dynamics of the system becomes:

$$\begin{bmatrix} \ddot{q} \\ k \end{bmatrix} = \begin{bmatrix} -M^{-1}N \\ 0_{n\times n} \end{bmatrix} + \begin{bmatrix} 0_{n\times n} & -M^{-1}\Phi \\ 0_{n\times n} & I_{n\times n} \end{bmatrix} \begin{bmatrix} \tau \\ \tau_k \end{bmatrix}$$

Problem

The decoupling matrix of the system is structurally singular

Solution

Dynamic extension on the input τ_k is needed

$$\ddot{\tau}_k = u_k$$

Dynamic Feedback Linearization

- Considering the very simple stiffness variation model

$$k_i = \tau_{k_i}$$

the dynamics of the system becomes:

$$\begin{bmatrix} \ddot{q} \\ k \end{bmatrix} = \begin{bmatrix} -M^{-1}N \\ 0_{n\times n} \end{bmatrix} + \begin{bmatrix} 0_{n\times n} & -M^{-1}\Phi \\ 0_{n\times n} & I_{n\times n} \end{bmatrix} \begin{bmatrix} \tau \\ \tau_k \end{bmatrix}$$

Problem

The decoupling matrix of the system is structurally singular

Solution

Dynamic extension on the input τ_k is needed

$$\ddot{\tau}_k = u_k$$

Feedback Linearized Model

- The system dynamics can be then rewritten as:

$$\begin{bmatrix} q^{[4]} \\ \ddot{k} \end{bmatrix} = \begin{bmatrix} \alpha(x_e) \\ 0_{n\times n} \end{bmatrix} + Q(x_e) \begin{bmatrix} \tau \\ u_k \end{bmatrix}$$

where

$$Q(x_e) = \begin{bmatrix} M^{-1}KB^{-1} & -M^{-1}\Phi \\ 0_{n\times n} & I_{n\times n} \end{bmatrix}$$

- By defining the control law:

$$\begin{bmatrix} \tau \\ u_k \end{bmatrix} = Q^{-1}(x_e)\left(-\begin{bmatrix} \alpha(x_e) \\ 0_{n\times n} \end{bmatrix} + \begin{bmatrix} v_q \\ v_k \end{bmatrix}\right)$$

we obtain the feedback linearized model:

$$\begin{bmatrix} q^{[4]} \\ \ddot{k} \end{bmatrix} = \begin{bmatrix} v_q \\ v_k \end{bmatrix}$$

Feedback Linearized Model

- The system dynamics can be then rewritten as:

$$\begin{bmatrix} q^{[4]} \\ \ddot{k} \end{bmatrix} = \begin{bmatrix} \alpha(x_e) \\ 0_{n\times n} \end{bmatrix} + Q(x_e) \begin{bmatrix} \tau \\ u_k \end{bmatrix}$$

where

$$Q(x_e) = \begin{bmatrix} M^{-1}KB^{-1} & -M^{-1}\Phi \\ 0_{n\times n} & I_{n\times n} \end{bmatrix}$$

- By defining the control law:

$$\begin{bmatrix} \tau \\ u_k \end{bmatrix} = Q^{-1}(x_e) \left(- \begin{bmatrix} \alpha(x_e) \\ 0_{n\times n} \end{bmatrix} + \begin{bmatrix} v_q \\ v_k \end{bmatrix} \right)$$

we obtain the feedback linearized model:

$$\begin{bmatrix} q^{[4]} \\ \ddot{k} \end{bmatrix} = \begin{bmatrix} v_q \\ v_k \end{bmatrix}$$

Control Strategy

- A static state feedback in the state space of the feedback linearized system is used:

$$v_c = \begin{bmatrix} v_q \\ v_k \end{bmatrix}, \quad v_f = \begin{bmatrix} q_d^{[4]} \\ \ddot{k}_d \end{bmatrix}$$

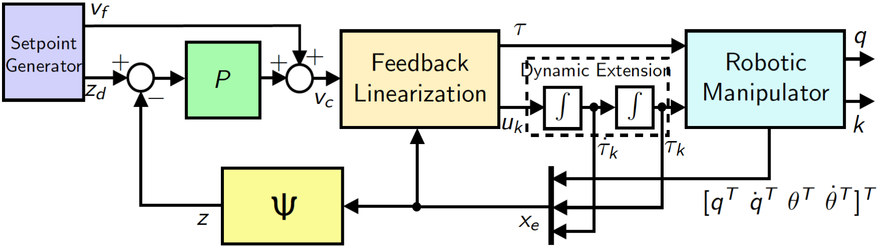
$$z_d = \begin{bmatrix} q_d^T & \dot{q}_d^T & \ddot{q}_d^T & {q_d^{[3]}}^T & k_d^T & \dot{k}_d^T \end{bmatrix}^T$$

- The state vector z of the feedback linearized system and a suitable nonlinear coordinate transformation are defined:

$$z = \begin{bmatrix} q^T & \dot{q}^T & \ddot{q}^T & {q^{[3]}}^T & k^T & \dot{k}^T \end{bmatrix}^T = \Psi(x_e) =$$

$$\begin{bmatrix} q \\ \dot{q} \\ -M^{-1}\left[N + \Phi\, k\right] \\ -M^{-1}\left[-\dot{M}\, M^{-1}\left[N + \Phi\, k\right] + \dot{N} + \Phi\, \dot{k} + \dot{\Phi}\, k\right] \\ k \\ \dot{k} \end{bmatrix}$$

Control System Architecture



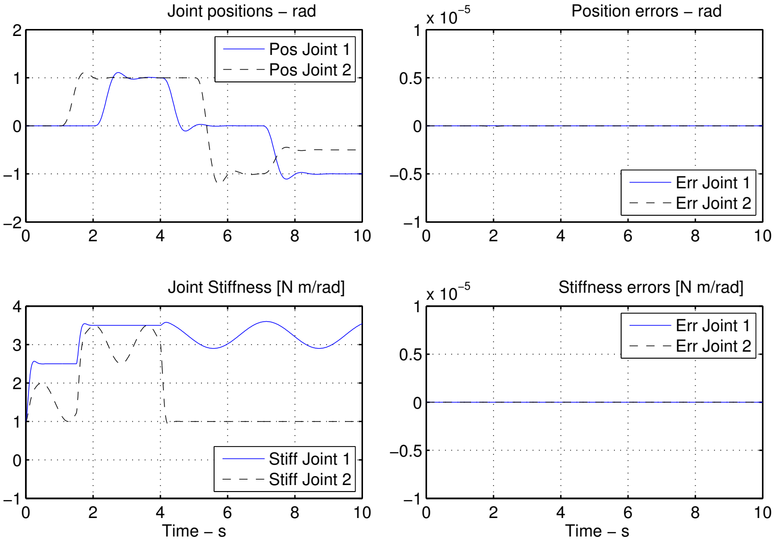
- The controller can be then rewritten as:

$$v_c = v_f + P[z_d - z] = v_f + P[z_d - \Psi(x_e)]$$

where

$$P = \begin{bmatrix} P_{q_0} & P_{q_1} & P_{q_2} & P_{q_3} & 0_{n\times n} & 0_{n\times n} \\ 0_{n\times n} & 0_{n\times n} & 0_{n\times n} & 0_{n\times n} & P_{k_0} & P_{k_1} \end{bmatrix}$$

Simulation of a two-link Planar Manipulator



Application to Antagonistic Variable Stiffness Devices

- Dynamic model of an antagonistic variable stiffness robot

$$
\begin{aligned}
M(q)\,\ddot{q} + N(q,\dot{q}) + \eta_\alpha - \eta_\beta &= 0 \\
B\,\ddot{\theta}_\alpha + \eta_\alpha &= \tau_\alpha \\
B\,\ddot{\theta}_\beta + \eta_\beta &= \tau_\beta
\end{aligned}
$$

- By introducing the auxiliary variables

$p = \frac{\theta_\alpha - \theta_\beta}{2}$	positions of the generalized joint actuators
$s = \theta_\alpha + \theta_\beta$	state of the virtual stiffness actuators
$F(s)$	generalized joint stiffness matrix (diagonal)
$g(q-p)$	strictly monotonically increasing functions (generalized joint displacements)
$h(q-p,s)$	such that $h_i(0,0) = 0$

$\tau = \tau_\alpha - \tau_\beta$, $\tau_k = \tau_\alpha + \tau_\beta$

it is possible to write

$$
\begin{aligned}
M(q)\,\ddot{q} + N(q,\dot{q}) + F(s)g(q-p) &= 0 \\
2B\ddot{p} + F(s)g(p-q) &= \tau \\
B\ddot{s} + h(q-p,s) &= \tau_k
\end{aligned}
$$

Application to Antagonistic Variable Stiffness Devices

- Dynamic model of an antagonistic variable stiffness robot

$$\begin{aligned} M(q)\,\ddot{q} + N(q,\dot{q}) + \eta_\alpha - \eta_\beta &= 0 \\ B\,\ddot{\theta}_\alpha + \eta_\alpha &= \tau_\alpha \\ B\,\ddot{\theta}_\beta + \eta_\beta &= \tau_\beta \end{aligned}$$

- By introducing the auxiliary variables

$p = \frac{\theta_\alpha - \theta_\beta}{2}$	positions of the generalized joint actuators
$s = \theta_\alpha + \theta_\beta$	state of the virtual stiffness actuators
$F(s)$	generalized joint stiffness matrix (diagonal)
$g(q-p)$	strictly monotonically increasing functions (generalized joint displacements)
$h(q-p,s)$	such that $h_i(0,0) = 0$

$\tau = \tau_\alpha - \tau_\beta$, $\tau_k = \tau_\alpha + \tau_\beta$

it is possible to write

$$\begin{aligned} M(q)\,\ddot{q} + N(q,\dot{q}) + F(s)g(q-p) &= 0 \\ 2B\ddot{p} + F(s)g(p-q) &= \tau \\ B\ddot{s} + h(q-p,s) &= \tau_k \end{aligned}$$

Application to Antagonistic Variable Stiffness Devices

- Dynamic model of an antagonistic variable stiffness robot

$$
\begin{aligned}
M(q)\,\ddot{q} + N(q,\dot{q}) + \eta_\alpha - \eta_\beta &= 0 \\
B\,\ddot{\theta}_\alpha + \eta_\alpha &= \tau_\alpha \\
B\,\ddot{\theta}_\beta + \eta_\beta &= \tau_\beta
\end{aligned}
$$

- By introducing the auxiliary variables

$p = \frac{\theta_\alpha - \theta_\beta}{2}$	positions of the generalized joint actuators
$s = \theta_\alpha + \theta_\beta$	state of the virtual stiffness actuators
$F(s)$	generalized joint stiffness matrix (diagonal)
$g(q-p)$	strictly monotonically increasing functions (generalized joint displacements)
$h(q-p,s)$	such that $h_i(0,0) = 0$

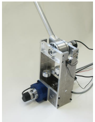
$\tau = \tau_\alpha - \tau_\beta$, $\tau_k = \tau_\alpha + \tau_\beta$

it is possible to write

$$
\begin{aligned}
M(q)\,\ddot{q} + N(q,\dot{q}) + F(s)g(q-p) &= 0 \\
2B\ddot{p} + F(s)g(p-q) &= \tau \\
B\ddot{s} + h(q-p,s) &= \tau_k
\end{aligned}
$$

Actual Variable Stiffness Joint Implementations

- For antagonistic actuated robot with exponential force/compression characteristic (Palli et al. 2007)

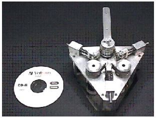


$$
\begin{aligned}
f_i(s_i) &= e^{a\, s_i} \\
g_i(q_i - p_i) &= b \sinh\big(c\,(q_i - p_i)\big) \\
h_i(q_i - p_i, s_i) &= d\,\big[\cosh\big(c\,(q_i - p_i)\big)\, e^{a\, s_i} - 1\big]
\end{aligned}
$$

- If transmission elements with quadratic force/compression characteristic are considered (Migliore et al. 2005)

$$
\begin{aligned}
f_i(s_i) &= a_1\, s_i + a_2 \\
g_i(q_i - p_i) &= q_i - p_i \\
h_i(q_i - p_i, s_i) &= b_1\, s_i^2 + b_2\,(q_i - p_i)^2
\end{aligned}
$$

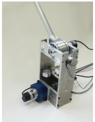
- For the variable stiffness actuation joint (VSA), using a third-order polynomial approximation of the transmission model (Boccadamo, Bicchi et al. 2006)



$$
\begin{aligned}
f_i(s_i) &= a_1\, s_i^2 + a_2\, s_i + a_3 \\
g_i(q_i - p_i) &= q_i - p_i \\
h_i(q_i - p_i, s_i) &= b_1\, s_i^3 + b_2\,(q_i - p_i)^2 s_i + b_3\, s_i
\end{aligned}
$$

Actual Variable Stiffness Joint Implementations

- For antagonistic actuated robot with exponential force/compression characteristic (Palli et al. 2007)

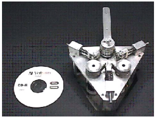


$$
\begin{aligned}
f_i(s_i) &= e^{a\, s_i} \\
g_i(q_i - p_i) &= b \sinh\big(c\,(q_i - p_i)\big) \\
h_i(q_i - p_i, s_i) &= d\,\big[\cosh\big(c\,(q_i - p_i)\big)\, e^{a\, s_i} - 1\big]
\end{aligned}
$$

- If transmission elements with quadratic force/compression characteristic are considered (Migliore et al. 2005)

$$
\begin{aligned}
f_i(s_i) &= a_1\, s_i + a_2 \\
g_i(q_i - p_i) &= q_i - p_i \\
h_i(q_i - p_i, s_i) &= b_1\, s_i^2 + b_2\,(q_i - p_i)^2
\end{aligned}
$$

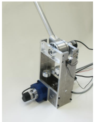
- For the variable stiffness actuation joint (VSA), using a third-order polynomial approximation of the transmission model (Boccadamo, Bicchi et al. 2006)



$$
\begin{aligned}
f_i(s_i) &= a_1\, s_i^2 + a_2\, s_i + a_3 \\
g_i(q_i - p_i) &= q_i - p_i \\
h_i(q_i - p_i, s_i) &= b_1\, s_i^3 + b_2\,(q_i - p_i)^2 s_i + b_3\, s_i
\end{aligned}
$$

Actual Variable Stiffness Joint Implementations

- For antagonistic actuated robot with exponential force/compression characteristic (Palli et al. 2007)

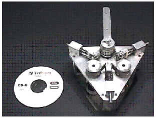


$$
\begin{aligned}
f_i(s_i) &= e^{a\,s_i} \\
g_i(q_i - p_i) &= b\sinh\big(c\,(q_i - p_i)\big) \\
h_i(q_i - p_i, s_i) &= d\left[\cosh\big(c\,(q_i - p_i)\big)\,e^{a\,s_i} - 1\right]
\end{aligned}
$$

- If transmission elements with quadratic force/compression characteristic are considered (Migliore et al. 2005)

$$
\begin{aligned}
f_i(s_i) &= a_1\,s_i + a_2 \\
g_i(q_i - p_i) &= q_i - p_i \\
h_i(q_i - p_i, s_i) &= b_1\,s_i^2 + b_2\,(q_i - p_i)^2
\end{aligned}
$$

- For the variable stiffness actuation joint (VSA), using a third-order polynomial approximation of the transmission model (Boccadamo, Bicchi et al. 2006)



$$
\begin{aligned}
f_i(s_i) &= a_1\,s_i^2 + a_2\,s_i + a_3 \\
g_i(q_i - p_i) &= q_i - p_i \\
h_i(q_i - p_i, s_i) &= b_1\,s_i^3 + b_2\,(q_i - p_i)^2 s_i + b_3\,s_i
\end{aligned}
$$

Conclusions

- The feedforward control action needed to perform a desired motion profile has been computed
- The feedback linearization problem with decoupled control has been solved taking into account different stiffness variation models
- The simultaneous asymptotic trajectory tracking of both the position and the stiffness has been achieved by means of an outer linear control loop
- These results can be easily extended to the mixed rigid/elastic case
- The proposed approach has been used to model several actual implementation of variable stiffness devices

ADL Surprise Festschrift

Thanks Alex!!