Proceedings of the 2003 IEEE/RSJ
Intl. Conference on Intelligent Robots and Systems
Las Vegas, Nevada - October 2003

Real-Time Adaptive Control for Haptic Manipulation with Active Observers

Rui Cortesão*, Jaeheung Park** and Oussama Khatib**

*University of Coimbra, Electrical and Computer Engineering Department, Institute of Systems and Robotics (ISR), 3030 Coimbra, Portugal, E-mail: cortesao@isr.uc.pt.

**Stanford University, Robotics Group, 94305-9010 CA, USA, E-mails: {park73, ok}@robotics.stanford.edu.

Abstract

The paper discusses compliant motion control using Active Observers (AOBs) applied in robotic manipulators. Stochastic estimation strategies for haptic manipulation are introduced. Stability and robustness analysis is made as a function of stiffness mismatches. Real time adaptation is discussed.

1 Introduction

Many robotic tasks are not appropriately implemented due to the lack of unifying and systematic methodologies in the control design. Often, the control synthesis is based on engineering expertise and experience rather than systematic scientific methods. A survey of robust control techniques including feedback linearization [8] linear H_∞ optimal controllers [9], PD and PID control schemes [1] can be seen in [7].

To design a control system, approximate and linearized models are frequently considered, allowing the extensive and rich theory of linear systems to be applied. Inaccurate models have to be tested to evaluate the control performance. The control design has to guarantee practical system stability linked with acceptable performance. Too sub-optimal solutions are unacceptable, even if stability and robustness requirements are fulfilled. To achieve a simple and modular control synthesis, decentralized control should be applied, decoupling the overall system into several subsystems, in which autonomous local controllers are designed. These subsystems typically correspond to one or few DOF of the control system. Unlike the centralized control approach that is more complex and system dependent, the couplings among subsystems are neglected. If the nature of the task creates strong couplings, without additional techniques decentralized control methods have poor results and may be even prohibitive. Sensor (e.g. force) based tasks give information about environment effects on the robot, being appropriate for adaptive control strategies. Compliant motion tasks require special attention, since the task constraints change abruptly (between contact and non-contact states) and the model parameters may have wide variations, particularly for very stiff and unstructured environments. In the sequel, a decentralized compliant motion controller is proposed for a robotic manipulator. The influence of modeling errors and couplings is reduced through feedback linearization techniques and AOBs. This controller has been successfully applied in haptic manipulation[1] [6].

2 Manipulator Dynamics

The manipulator dynamics describes the motion of a manipulator subject to applied torques and external forces. Given a set of generalized coordinates q (usually, joint angles for revolute joints) describing the robot's pose, the well-known robot dynamics is given by

$$M(q)\ddot{q} + v(q,\dot{q}) + g(q) = \tau. \quad (1)$$

$M(q)$ is the mass matrix, $v(q,\dot{q})$ is the vector of Coriolis and centripetal forces, $g(q)$ is the gravity term and τ is the generalized torque acting on q. Defining the Jacobian $J(q)$ as

$$\dot{X} = J(q)\dot{q}, \quad (2)$$

with X the Cartesian position,

$$\ddot{q} = J^{-1}(\ddot{X} - \dot{J}\dot{q}). \quad (3)$$

Applying (3) in (1) and knowing the relation between the joint torque τ and the Cartesian force F at the end-effector, $\tau = J^T F$, (1) can be written in Cartesian coordinates as

$$\Lambda\ddot{X} + V_x(q,\dot{q}) + g_x(q) = F, \quad (4)$$

[1] In this context, haptic manipulation means that a robotic manipulator is controlled by haptic devices.

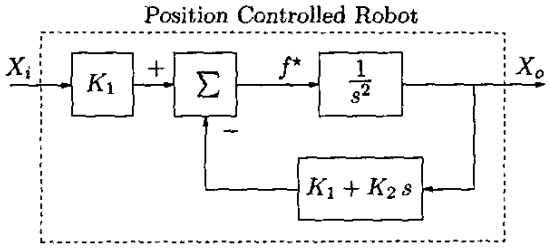

Figure 1: *Modification of the desired plant through feedback.*

with

$$\Lambda = J^{-T} M J^{-1}, \tag{5}$$

$$V_x = V(q,\dot{q}) - \Lambda \dot{J}\dot{q} \tag{6}$$

and

$$g_x = J^{-T} g(q). \tag{7}$$

An external force F_e appears always at the end-effector whenever the robot is in contact. Hence, (4) can be written as

$$\Lambda \ddot{X} + V_x(q,\dot{q}) + g_x(q) = F_c - F_e - F_f, \tag{8}$$

where F_c and F_f are respectively the forces due to the commanded torque and friction.

3 System Plant

A control architecture robust to model uncertainties should be designed to accomplish enhanced performance in compliant motion tasks. If the desired system plant is

$$\ddot{X} = f^*, \tag{9}$$

F_c should be[2]

$$F_c = \hat{F}_e + \hat{F}_f + \hat{V}_x(q,\dot{q}) + \hat{g}_x(q) + \hat{\Lambda} f^*. \tag{10}$$

Equation (9) defines a decoupled system for each Cartesian dimension with unitary mass. The estimation of F_e, $\hat{F}_e$, affects the control strategy, as will be explained in Section 5. The terms $\hat{V}_x(q,\dot{q})$, $\hat{g}_x(q)$ and $\hat{\Lambda}$ can be computed for a given robot. $\hat{F}_f$ is difficult to obtain. However, if low velocities are associated with the contact task, $\hat{F}_f$ as well as $\hat{V}_x(q,\dot{q})$ can be neglected. The estimation errors present in (10) corrupt (9). To increase robustness to model errors, the desired plant poles at the origin are "shifted" to the left using feedback, as shown in Figure 1. For a critically damped response (damping factor $\zeta = 1$) with time constant τ_o, the feedback is given by

$$K_2 = \frac{2}{\tau_o} \text{ and } K_1 = \frac{K_2^2}{4}. \tag{11}$$

[2]The symbol ˆ means estimate.

The problem of this approach is that the force controlled robot becomes a position controlled robot. External forces (e.g. human contact) applied to the robot's body experience a very stiff contact due to position feedback. Eliminating the position loop and inserting the deadtime and the system stiffness, the plant represented in Figure 1 becomes

$$G(s) = \frac{K_s\, e^{-sT_d}}{s(s + K_2\, e^{-sT_d})}. \tag{12}$$

If T_d is small, (12) can be approximated by

$$G_a(s) = \frac{K_s\, e^{-sT_d}}{s(s + K_2)} \tag{13}$$

for a wide range of frequencies. Its equivalent temporal representation is

$$\dddot{y} + K_2 \ddot{y} = K_s \dot{u}(t - T_d), \tag{14}$$

where y is the plant output (Cartesian force at the robot's end effector), and u is the plant input (force). Defining the state variables $x_1 = y$ and $x_2 = \dot{y}$, (14) can be written as

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -p_1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ k_1 \end{bmatrix} u(t - t_d). \tag{15}$$

In compact form,

$$\begin{cases} \dot{x} = Ax(t) + Bu(t - T_d) \\ y(t) = x_1 \end{cases}. \tag{16}$$

Discretizing (16) with sampling time h, the equivalent discrete time system is

$$\begin{cases} x_{r,k} = \Phi_r x_{r,k-1} + \Gamma_r u_{k-1} \\ y_k = C_r x_{r,k}, \end{cases}, \tag{17}$$

with

$$T_d = (d-1)h + \tau', \tag{18}$$

$$0 < \tau' \leq h, \tag{19}$$

$$x_{r,k} = \begin{bmatrix} x_k & u_{k-d} & \cdots & u_{k-2} & u_{k-1} \end{bmatrix}^T, \tag{20}$$

$$\Phi_r = \begin{bmatrix} \Phi_1 & \Gamma_1 & \Gamma_0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}, \tag{21}$$

$$\Gamma_r = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \end{bmatrix}^T \tag{22}$$

and

$$C_r = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \end{bmatrix}. \tag{23}$$

Φ_1, Γ_0 and Γ_1 are given by (24) to (26), respectively [2],

$$\Phi_1 = e^{Ah} = \phi(h), \tag{24}$$

$$\Gamma_0 = \int_0^{h-\tau'} \phi(\lambda)\, d\lambda\, B \qquad (25)$$

and

$$\Gamma_1 = \phi(h-\tau') \int_0^{\tau'} \phi(\lambda)\, d\lambda\, B. \qquad (26)$$

x_k has two states representing the force and force derivative. The other states appear due to dead-time. The continuous state transition and command matrices are

$$\phi(t) = \begin{bmatrix} 1 & \frac{1-e^{-K_2 t}}{K_2} \\ 0 & e^{-K_2 t} \end{bmatrix} \text{ and } B = \begin{bmatrix} 0 \\ K_s \end{bmatrix}. \qquad (27)$$

From (27), the computation of Φ_1, Γ_0 and Γ_1 is straightforward.

4 AOB Design

Knowing the discrete state space representation (17), the theory of AOBs introduced in [3] can be applied in a straightforward way to achieve robust adaptive control in the presence of uncertainties. The AOB uses discrete Kalman techniques to estimate the system state and its disturbances. The main goal is to fit a physical system (i.e. its input/output behavior) into a linear mathematical model, rather than to fit a mathematical model into a physical system. A special Kalman filter (AOB) has to be designed to achieve a model reference adaptive control architecture. The desired closed loop system is imposed to the state estimation. An extra state p_k (called active state) is generated to eliminate an equivalent disturbance referred to the system input[3]. This equivalent disturbance exists whenever the physical system is different from the desired model. The first-order AOB algorithm (AOB-1) is summarized in Section 4.1.

4.1 AOB-1 Algorithm

Controlling the system of (17) through state feedback from an observer and inserting p_k in the loop, the overall system can be described by

$$\begin{bmatrix} x_{r,k} \\ p_k \end{bmatrix} = \begin{bmatrix} \Phi_r & \Gamma_r \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_{r,k-1} \\ p_{k-1} \end{bmatrix} + \begin{bmatrix} \Gamma_r \\ 0 \end{bmatrix} u_{k-1} + \xi_{x_k} \qquad (28)$$

and

$$y_k = C_a \begin{bmatrix} x_{r,k-1} & p_{k-1} \end{bmatrix}^T + \eta_k, \qquad (29)$$

where

$$u_{k-1} = r_{k-1} - \begin{bmatrix} L_r & 1 \end{bmatrix} \begin{bmatrix} \hat{x}_{r,k-1} \\ \hat{p}_{k-1} \end{bmatrix}. \qquad (30)$$

[3]The general AOB algorithm uses N extra states to describe p_k [4], [3].

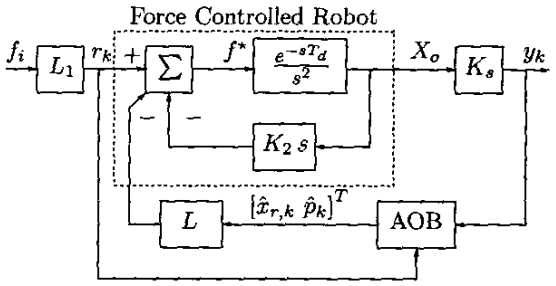

Figure 2: *Compliant motion control with the AOB in the loop. L_1 is the first element of L, and f_i is the force input.*

The stochastic inputs ξ_k and η_k represent respectively model and measure uncertainties. The state estimate of (28) is based on the desired closed loop (i.e. $\hat{p}_k = p_k$ and $\hat{x}_{r,k} = x_{r,k}$). It is

$$\begin{bmatrix} \hat{x}_{r,k} \\ \hat{p}_k \end{bmatrix} = \begin{bmatrix} \Phi_r - \Gamma_r L_r & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{x}_{r,k-1} \\ \hat{p}_{k-1} \end{bmatrix} + \begin{bmatrix} \Gamma_r \\ 0 \end{bmatrix} r_{k-1} + K_k(y_k - \hat{y}_k), \qquad (31)$$

with

$$\hat{y}_k = C_a \left(\begin{bmatrix} \Phi_r - \Gamma_r L_r & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{x}_{r,k-1} \\ \hat{p}_{k-1} \end{bmatrix} + \begin{bmatrix} \Gamma_r \\ 0 \end{bmatrix} r_{k-1} \right) \qquad (32)$$

and

$$C_a = \begin{bmatrix} C_r & 0 \end{bmatrix}. \qquad (33)$$

The Kalman gain K_k reflects the uncertainty associated to each state based on model and measure uncertainties. It is computed from

$$K_k = P_{1k}\, C_a^T \left[C_a\, P_{1k}\, C_a^T + R_k \right]^{-1}, \qquad (34)$$

with

$$P_{1k} = \Phi_n\, P_{k-1}\, \Phi_n^T + Q_k \qquad (35)$$

and

$$P_k = P_{1k} - K_k\, C_a\, P_{1k}. \qquad (36)$$

Φ_n is the augmented open loop matrix,

$$\Phi_n = \begin{bmatrix} \Phi_r & \Gamma_r \\ 0 & 1 \end{bmatrix}. \qquad (37)$$

Q_k is the system noise matrix and represents model uncertainty. It is given by

$$Q_k = \begin{bmatrix} Q_{x_{r,k}} & 0 \\ 0 & Q_{p_k} \end{bmatrix}. \qquad (38)$$

The measurement noise matrix R_k represents measure uncertainty. P_k is the mean square error matrix. Its initial value should reflect at least the uncertainty in the state estimation. It should not be lower than the initial matrix Q_k. Figure 2 shows the control architecture with the AOB in the loop.

5 AOB Estimation Strategies for Haptic Manipulation

Model reference adaptive control appears if $Q_{x_{r,k}}$ is much smaller than Q_{p_k}. In this case, the estimation for the system state follows the reference model. Everything that does not fit in the $x_{r,k}$ model goes to p_k. However, for compliant motion tasks (with or without haptic devices), the estimation of force (first state) from the model is very inaccurate, since the system stiffness K_s may have abrupt and unpredictable changes. Providing methods for on-line estimation of K_s [6] and increasing $Q_{x_{r,k}}$ for the first state creates better conditions to estimate the force. Knowing the structure of Q_k, the relation between R_k and Q_k makes the estimates more (R_k low) or less (R_k high) sensitive to measures. These stochastic parameters are a powerful tool in the control design, creating enough space to explore complex estimation strategies for highly unstructured tasks.

6 Pole Placement for Haptic Manipulation

In force-based tasks, force overshoots/undershoots are usually undesired. Hence, the state feedback gain L_r can be computed by Ackermann's formula to achieve a critically damped system ($\zeta = 1$). The other poles due to deadtime should be mapped far away from the dominant poles, to neglect their influence in the system response. In our setup they were mapped at $z = 0$. The closed loop time constant τ_c should be related to the open loop (plant) time constant τ_o. τ_c should be small enough to enable the task execution with comfortable performance. However, it should not be too small to avoid saturation effects in the command effort. In our setup, $\tau_c = 3\tau_o$. The closed loop settling time is about $5 \times \tau_c$ (0.375 [s]), which is adequate for human-controlled tasks. Mathematically,

$$L_r = \begin{bmatrix} 0 & \cdots & 0 & 1 \end{bmatrix} W_c^{-1} P(\Phi_r). \quad (39)$$

W_c is the reachability matrix,

$$W_c = \begin{bmatrix} \Gamma_r & \Phi_r\Gamma_r & \cdots & \Phi_r^{n-1}\Gamma_r \end{bmatrix} \quad (40)$$

and $P(\Phi_r)$ is the characteristic polynomial in Φ_r,

$$P(\Phi_r) = \Phi_r^{d+2} + a_1\Phi_r^{d+1} + a_2\Phi_r^{d}, \quad (41)$$

with[4]

$$a_1 = -2e^{-\zeta w_n h} cos\left(\sqrt{1-\zeta^2} w_n h\right) \quad (42)$$

and

$$a_2 = e^{-2\zeta w_n h}. \quad (43)$$

[4]For a critically damped system $\tau_c = 1/w_n$.

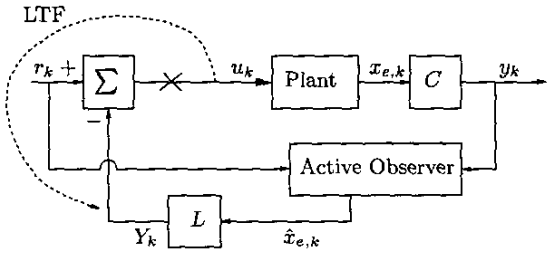

Figure 3: LTF computation with the AOB in the loop.

6.1 Free Space Behavior

The AOB control architecture is kept even for free space conditions (no control switching). In this case, the force output is always zero. Hence, from (31), $y_k = 0$. For steady state conditions ($\hat{x}_{r,k} = \hat{x}_{r,k-1}$, r_k is constant as well as K_k), straightforward analysis of (31) gives[5]

$$\hat{x}_{r,k} = [I - (I - K_kC_r)(\Phi_r - \Gamma_rL_r)]^{-1} \cdot (I - K_kC_r)\Gamma_r r_{k-1} \quad (44)$$

and

$$\hat{p}_k = \hat{p}_{k-1} - K_kC_r\hat{x}_{r,k}. \quad (45)$$

It can be inferred from (44)-(45) that $\hat{x}_{r,k}$ converges to a known constant value and the $\hat{p}_k$ derivative is also constant. The Cartesian velocity $\dot{X}_o$ follows the input

$$r_k - L\begin{bmatrix} \hat{x}_{r,k} & \hat{p}_k \end{bmatrix} \quad (46)$$

that changes linearly with time (See Figure 2).

7 AOB Stability and Robustness

This section analyzes relative stability of AOB based controllers in the presence of model errors. The loop transfer function[6] (LTF) of the control system has to be derived. A schematic representation of it is depicted in Figure 3. Applying u_k to the plant input and considering all other inputs zero (necessary to compute the LTF), (28)-(29) can be written as

$$x_{e,k} = \Phi x_{e,k-1} + \Gamma u_{k-1} \quad (47)$$

and

$$y_k = Cx_{e,k}, \quad (48)$$

with

$$x_{e,k} = \begin{bmatrix} x_{r,k} & p_k \end{bmatrix}^T. \quad (49)$$

[5]In (44), K_k has only the Kalman gains that affect $\hat{x}_{r,k}$ and I is the identity matrix. In (45), K_k is the Kalman gain of the $\hat{p}_k$ state.

[6]The loop transfer function is the product of the transfer functions of forward and feedback loops. Special attention should be made when observers are in the loop [5].

The real system matrix Φ is equal to the nominal matrix Φ_n (i.e. the one used in the design) plus the unknown error $\Delta\Phi$ due to unmodeled terms. Mathematically,

$$\Phi = \Phi_n + \Delta\Phi. \tag{50}$$

The AOB state estimate is[7] of form

$$\hat{x}_{e,k} = \Phi_c \hat{x}_{e,k-1} + K_k \left[y_k - C \left(\Phi_c \hat{x}_{e,k-1} \right) \right], \tag{51}$$

with $\Phi_c = \Phi_n - \Gamma L$, $\Gamma = \begin{bmatrix} \Gamma_r & 0 \end{bmatrix}^T$ and $L = \begin{bmatrix} L_r & 0 \end{bmatrix}$. Defining the estimation error e_k as

$$e_k = x_{e,k} - \hat{x}_{e,k}, \tag{52}$$

$\hat{x}_{e,k}$ and e_k can be written as

$$\begin{bmatrix} \hat{x}_{e,k} \\ e_k \end{bmatrix} = \begin{bmatrix} M_{1,1} & K_k C \Phi \\ M_{2,1} & (I - K_k C) \Phi \end{bmatrix} \begin{bmatrix} \hat{x}_{e,k-1} \\ e_{k-1} \end{bmatrix} + \begin{bmatrix} K_k C \Gamma \\ (I - K_k C) \Gamma \end{bmatrix} u_{k-1}, \tag{53}$$

where

$$M_{1,1} = \Phi_n - \Gamma L + K_k C \left(\Delta\Phi + \Gamma L \right) \tag{54}$$

and

$$M_{2,1} = (I - K_k C) \left(\Delta\Phi + \Gamma L \right). \tag{55}$$

Once the LTF output is $L\hat{x}_{e,k}$,

$$Y_k = \begin{bmatrix} L & 0 \end{bmatrix} \begin{bmatrix} \hat{x}_{e,k} \\ e_k \end{bmatrix}. \tag{56}$$

The transfer function of the state space equations (53) and (56) is the LTF, $H_{\mathrm{LTF}}(z)$, given by

$$H_{\mathrm{LTF}}(z) = \begin{bmatrix} L & 0 \end{bmatrix} \left[I - \phi z^{-1} \right]^{-1} \gamma z^{-1}, \tag{57}$$

in which ϕ and γ are the state transition and command matrices of (53), respectively, and I is the identity matrix. Knowing $H_{\mathrm{LTF}}(z)$, it is straightforward to compute Nyquist/Bode plots and the respective phase and gain margins[8].

8 Experimental Setup

The slave robot is a PUMA$^{\mathrm{TM}}$ 560, which has a stiff JR3 force sensor on the end-effector. The PUMA has 6 DOF and is connected to a computer (Pentium II 333 [MHz], QNX real-time OS) through a TRC205 controller and a ServoToGo$^{\mathrm{TM}}$ board. The sampling time is $h = 2$ [ms] ($f_s = 500$ [Hz]). The deadtime was obtained experimentally. It is

$$T_d = 3 \times h. \tag{58}$$

[7] See (31) for $r_k = 0$.

[8] In the Matlab$^{\mathrm{TM}}$ environment, the LTF representation in state space or transfer function is all that is needed to have Nyquist/Bode plots.

The working space has objects with different stiffnesses. When the robot is manipulating, the system stiffness is approximately given by the object stiffness, since the JR3 sensor and the robot are very stiff. Table 1 presents experimental values.

	free space	sponge	book	desk
K_s [N/m]	100	300	3000	6000

Table 1: *Object Stiffnesses. In free space, the minimum value of K_s represents the one used in the controller.*

8.1 AOB Stochastic Matrices

For each Cartesian dimension, the AOB stochastic matrices are

$$Q_k = \begin{bmatrix} 10^{-3} & \cdots & 0 & 0 \\ 0 & 10^{-12} & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 10^{-12} & 0 \\ 0 & \cdots & 0 & 10^{-5} \end{bmatrix}, \tag{59}$$

$R_k = 8.5$ and $P_0 = Q_0$. This design entails the following steady-state Kalman gains:

$$K_k \times 10^3 \approx \begin{bmatrix} 10.9 & 1.1 & 0.9 & 0.9 & 0.9 & 0.9 \end{bmatrix}^T. \tag{60}$$

8.2 Robustness

In compliant motion tasks it is important to analyze robustness when there are stiffness mismatches. From (18), (19) and (58), $d = 3$ and $\tau' = h$. Hence, from (25), $\Gamma_0 = \begin{bmatrix} 0 & 0 \end{bmatrix}^T$. Moreover, Φ_1 given by (24) does not depend on K_s. If K_s changes to $K_s + \Delta K_s$, Γ_1 changes to $\Gamma_1 + \Delta\Gamma_1$ (see (26)), where

$$\Delta\Gamma_1 = \frac{\Delta K_s}{K_s} \Gamma_1. \tag{61}$$

From (21),

$$\Delta\Phi_r = \begin{bmatrix} 0 & \Delta\Gamma_1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}. \tag{62}$$

Knowing (37),

$$\Delta\Phi = \begin{bmatrix} \Delta\Phi_r & 0 \\ 0 & 0 \end{bmatrix}. \tag{63}$$

Using (53), stability can be analyzed based on the stiffness mismatch. Figure 4 shows that robustness increases with K_s. For a nominal stiffness $K_s = 100$ [N/m] the control structure is stable up

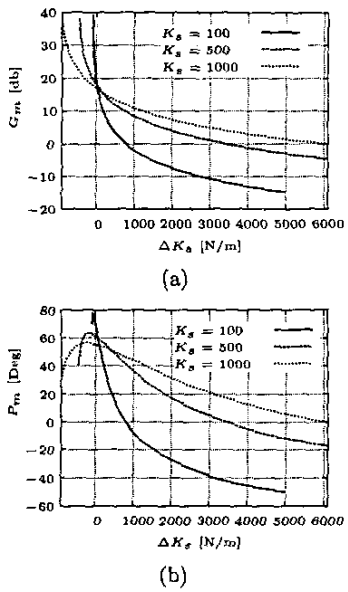

Figure 4: Robustness results. Relative stability with stiffness errors. (a) Gain Margin. (b) Phase Margin.

to $\Delta K_s \approx 800$ [N/m]. This means that if no K_s adaptation is performed, the overall system is unstable in contact with stiff surfaces like a book or desk (see Table 1). Methods for on-line stiffness estimation are proposed in [6].

8.3 Real Time Issues

This section analyzes properties of AOB based controllers for on-line stiffness adaptation. In haptic tasks, the system stiffness is function of the environment. Contact/non-contact states with stiff objects are critical, since the stiffness changes are big. To achieve force responses independent of the contact object, K_s has to be estimated on-line, to adapt the AOB accordingly.

Control Adaptation. The feedback gain L_r of the controller can be easily adapted for new environment stiffnesses without a complete computation of Ackermann's formula. It can be shown [4] that for a nominal stiffness K_s with corresponding feedback gains

$$L_r = \begin{bmatrix} l_1 & l_2 & l_3 & \cdots & l_n \end{bmatrix}, \qquad (64)$$

if K_s changes ΔK_s, the new L_r vector should be computed from

$$L_r = \begin{bmatrix} \frac{l_1}{(1+\Delta K_s/K_s)} & \frac{l_2}{(1+\Delta K_s/K_s)} & l_3 & \cdots & l_n \end{bmatrix}. \qquad (65)$$

The feedback gains of the state variables due to deadtime do not change. Only a proportional factor needs to be computed to update L_r for the "core state".

State Estimation. When K_s changes ΔK_s, the Φ_r matrix changes to $\Phi_r + \Delta\Phi_r$. Only two elements of this matrix have to be recomputed. The Kalman gains K_k are obtained on-line from (34)-(36). The state estimate of the AOB in (31) needs to be updated, reflecting the changes in Φ_r, L_r and K_k.

9 Conclusions

The paper presents a decentralized compliant motion control with AOBs. No control switching between contact/non-contact states is required. Estimation strategies for haptic manipulation have been proposed. If the system model is inaccurate, sensor-based estimations should be followed. Stability and robustness analysis have shown that on-line stiffness adaptation is necessary if the robot manipulates stiff objects. Moreover, robustness increases with the nominal value of K_s. Real-time methods have been presented to adapt the state estimation and the control gains when the stiffness changes.

Acknowledgment

This work was partially supported by FCT (Portuguese Science and Technology Foundation) project number POSI/1999/SRI/33594 and by the Luso-American foundation (FLAD), project 56/02.

References

[1] S. Arimoto. *Control Theory of Nonlinear Mechanical Systems.* Oxford University Press, 1996.

[2] K. J. Åström and B. Wittenmark. *Computer Controlled Systems: Theory and Design.* Prentice Hall, 1997.

[3] R. Cortesão. *Kalman Techniques for Intelligent Control Systems: Theory and Robotic Experiments.* PhD thesis, University of Coimbra, 2003.

[4] R. Cortesão, R. Koeppe, U. Nunes, and G. Hirzinger. Compliant motion control with stochastic active observers. In *Proc. of the Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 1876–1881, USA, 2001.

[5] J. Doyle and G. Stein. Robustness with observers. *IEEE Trans. on Automatic Control*, 24(4):607–611, August 1979.

[6] J. Park, R. Cortesão, and O. Khatib. Robust and adaptive teleoperation for compliant motion tasks. In *Proc. of Int. Conf. on Advanced Robotics (ICAR'03)*, Portugal, 2003. (submitted).

[7] H. Sage, M. Mathelin, and E. Ostertag. Robust control of robot manipulators: A survey. *Int. J. of Control*, 72(16):1498–1522, 1999.

[8] M. Spong and M. Vidyasagar. *Robot Dynamics and Control.* John Wiley & Sons, 1989.

[9] W. Stout and M. Sawan. Application of H_∞ theory to robot manipulator control. In *Proc. of the Int. Conf. on Control Application*, pages 148–153, USA, 1992.