Chapter 9
Robust Control of Cooperative Manipulators

9.1 Introduction

In this chapter we deal with the problem of robust position control for cooperative manipulators rigidly connected to an undeformable load. Design paradigms to solve force/position control problems have been established in the literature to improve the performance of the cooperation. In [8], a control strategy for cooperative manipulators was proposed based on the independence of the position and force controls. The applied force between the manipulator end-effectors and the object is decomposed into motion force and squeeze force, which must be controlled. In [7], the hybrid position/force controller proposed in [8] is extended to underactuated cooperative manipulators.

Cooperative manipulators, like any electromechanical system, are subject to parametric uncertainties and external disturbances. In [4], a semi-decentralized adaptive fuzzy controller with H_1-performance is developed for fully-actuated cooperative manipulators. In that work, the dynamic model is derived using the order reduction procedure proposed in [5] for constraint manipulators; only simulation results are presented to validate the controller's performance.

In this chapter, two nonlinear H_1 control techniques based on centralized control strategies are evaluated for underactuated cooperative manipulators: H_1 control for linear parameter varying (LPV) systems [3] and H_1 control based on game theory [2]. These controllers are applied considering the control strategy proposed in [8], where the squeeze force control is designed independently of the position control. In these cases, the H_1 performance index considers only the position control problem.

The adaptive fuzzy-based controller for fully-actuated cooperative manipulators proposed in [4] includes the position and squeeze force errors in the H_1 performance index. Following a similar approach, in this chapter we present a neural network-based H_1 control for fully-actuated and underactuated manipulators [6]. We assume that a nominal dynamic model is available for the neural network to approximate the model. As in [4], the H_1 performance index includes the position

A. A. G. Siqueira et al., *Robust Control of Robots*,
DOI: 10.1007/978-0-85729-898-0_9, Springer-Verlag London Limited 2011

and squeeze force errors, which guarantees an overall disturbance rejection. Furthermore, for underactuated manipulators, a practical solution is presented for the squeeze force control. In this case, only some components of the squeeze force can be controlled and constraints are imposed on the components that are not controlled. The approach guarantees asymptotic convergence of the motion tracking errors in spite of parametric uncertainties and external disturbances. Experimental results obtained with two cooperative UARM systems illustrate their efficiency.

This chapter is organized as follows. Sects. 9.2 and 9.3 present the dynamic equations for fully-actuated and underactuated cooperative manipulators, respectively. Section 9.4 presents the squeeze force control problem. Section 9.5 presents the necessary dynamic model formulation to implement H_1 controllers via quasi-LPV representation and via game theory. Section 9.6 develops the neural network-based adaptive H_1 control approach for cooperative manipulators. Section 9.7 concludes the chapter with design procedures, experimental results, and comparative studies.

9.2 Fully-Actuated Cooperative Manipulators

Consider a system composed of m fully-actuated cooperative manipulators, each one with n degrees of freedom. Let $q_i \in \Re^n$ be the vector of generalized coordinates of manipulator i and $x_o \in \Re^n$ the position and orientation of the load, which is rigidly connected to the end-effectors of the individual manipulators. As discussed in Chap. 7, this configuration generates a geometric constraint of the form $x_o = \phi_i(q_i)$, for $i = 1, 2, \ldots, m$. The corresponding velocity constraints are given by Eq. 7.2, which can be written as:

$$\dot{q}_i = J_i^{-1}(q_i) J_{o_i}(x_o) \dot{x}_o,$$

for $i = 1, 2, \ldots, m$, where $J_i(q_i)$ is the Jacobian matrix from joint velocities to end-effector velocities of arm i and $J_{o_i}(x_o)$ is the Jacobian matrix from load velocities to end-effector velocities of arm i. We assume that $J_i^{-1}(q_i)$ is well-defined. The kinematic constraints can be expressed by:

$$\dot{\theta} = \begin{bmatrix} I_n \\ J^{-1}(q) J_o(x_o) \end{bmatrix} \dot{x}_o \triangleq B(x_o)\dot{x}_o, \qquad (9.1)$$

where $\theta = [x_o^T q_1^T \cdots q_m^T]^T$, $q = [q_1^T \cdots q_m^T]^T$, $J_o(x_o) = [J_{o_1}^T(x_o) \cdots J_{o_m}^T(x_o)]^T$, and

$$J(q) = \begin{bmatrix} J_1(q_1) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & J_m(q_m) \end{bmatrix}.$$

The dynamic equation of the load is given by:

$$M_o(x_o)\ddot{x}_o + C_o(x_o, \dot{x}_o)\dot{x}_o + g_o(x_o) = J_o^T(x_o)h, \tag{9.2}$$

where $M_o(x_o)$ is the inertia matrix, $C_o(x_o, \dot{x}_o)$ is the Coriolis and centripetal matrix, $g_o(x_o)$ is the gravitational torque vector, and $h = [h_1^T \cdots h_m^T]^T$, with $h_i \in \mathbb{R}^n$, is the vector of forces applied by manipulator i on the load.

The dynamic equation of manipulator i is given by:

$$M_i(q_i)\ddot{q}_i + C_i(q_i, \dot{q}_i)\dot{q}_i + g_i(q_i) = \tau_i + J_i^T(q_i)h_i, \tag{9.3}$$

where $M_i(q_i)$ is the inertia matrix, $C_i(q_i, \dot{q}_i)$ is the Coriolis and centripetal matrix, $g_i(q_i)$ is the gravitational torque vector, and τ_i is the applied torque vector. The dynamic equation of the overall cooperative system can be written as:

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + g(\theta) = \begin{bmatrix} 0 \\ \tau \end{bmatrix} + \begin{bmatrix} J_o^T(x_o) \\ J^T(q) \end{bmatrix} h, \tag{9.4}$$

where $g(\theta) = [g_o(x_o)^T g_1(q_1)^T \cdots g_m^T(q_m)]^T$, and $\tau = [\tau_1^T \cdots \tau_m^T]^T$,

$$M(\theta) = \begin{bmatrix} M_o(x_o) & 0 & \cdots & 0 \\ 0 & M_1(q_1) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & M_m(q_m) \end{bmatrix}, \quad \text{and}$$

$$C(\theta, \dot{\theta}) = \begin{bmatrix} C_o(x_o, \dot{x}_o) & 0 & \cdots & 0 \\ 0 & C_1(q_1, \dot{q}_1) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & C_m(q_m, \dot{q}_m) \end{bmatrix}.$$

The projection of the applied force h on the frame fixed on the center of mass of the load ($h_o = J_{oq}^T h$) can be orthogonally decomposed as:

$$h_o = h_{os} + h_{om}, \tag{9.5}$$

where $h_{os} \in X_s$ is the vector of squeeze forces and $h_{om} \in X_m$ is the vector of motion forces (see Chap. 7 for details). Considering this decomposition of forces, the dynamic equation of the cooperative manipulator (9.4) can be represented as:

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + g(\theta) = \tau_v + \overline{W}^T(\theta)h_{os}, \tag{9.6}$$

where τ_v is an auxiliary control input given by

$$\tau_v = \begin{bmatrix} W^T h_{om} \\ \tau + J^T(q)\overline{J}_{oq}^{T}(x_o)h_{om} \end{bmatrix},$$

$\overline{W}(\theta) = W J_{oq}^{-1}(x_o) J(q)$ is a Jacobian matrix, and $W^T = [I_n \; I_n \; \cdots \; I_n]$. If the auxiliary control input is partitioned in two vectors $\tau_{v1} = W^T h_{om}$ and $\tau_{v2} = \tau - J^T(q) J_{oq}^{-T}(x_o) h_{om}$, the applied torque vector can be computed by:

$$\tau = \tau_{v2} - J^T(q) J_{oq}^{-T}(x_o)(W^T)^{\#} \tau_{v1}, \tag{9.7}$$

where $(W^T)^{\#} = W(W^T W)^{-1}$ is the pseudo-inverse of W^T. The motion force is given by $h_{om} = (W^T)^{\#} \tau_{v1}$. Hence, the control problem is to find an auxiliary control τ_v that guarantees stability and robustness against disturbances.

Considering the kinematic constraints (9.1) and multiplying the dynamic equation of the cooperative manipulator (9.6) by $B^T(x_o)$ (to eliminate the squeeze force term, since $B^T(x_o)\overline{W}^T(x_o) = 0$) we obtain:

$$\overline{M}(x_o)\ddot{x}_o + \overline{C}(x_o, \dot{x}_o)\dot{x}_o + \overline{g}(x_o) = \overline{\tau}_v, \tag{9.8}$$

where $\overline{M}(x_o) = B^T(x_o) M(x_o) B(x_o)$, $\overline{g}(x_o) = B^T(x_o) g(x_o)$, $\overline{\tau}_v = B^T(x_o)\tau_v$, and $\overline{C}(x_o, \dot{x}_o) = B^T(x_o) M(x_o) \dot{B}(x_o) + B^T(x_o) C(x_o, \dot{x}_o) B(x_o)$.

9.3 Underactuated Cooperative Manipulators

We assume now that the joints of the cooperative manipulator include n_a active joints (with actuators) and n_p passive joins (without actuators or whose actuators failed). The kinematic constraints (9.1) can be rewritten as:

$$\dot{\widetilde{\theta}} = \begin{bmatrix} I_n \\ -J_{AP}^{-1}(q) J_o(x_o) \end{bmatrix} \dot{x}_o - \widetilde{B}(x_o)\dot{x}_o, \tag{9.9}$$

where $\widetilde{\theta} = [x_o^T \; q_a^T \; q_p^T]^T$, $q_a \in \Re^{n_a}$ is vector of active joint positions, $q_p \in \Re^{n_p}$ is the vector of passive joint positions and $J_{AP}(q)$ is a Jacobian matrix generated from the orthogonal permutation matrix P_{AP} [7]. Therefore, if

$$\widetilde{q} = [q_a^T \; q_p^T]^T = P_{AP}[q_1^T \; q_2^T \; \cdots \; q_m^T]^T,$$

then

$$J_{AP}(q) = [J_a(q_a) \; J_p(q_p)] = J(q) P_{AP}.$$

The dynamic equation of the underactuated cooperative manipulator system is given by:

$$\widetilde{M}(\widetilde{\theta})\ddot{\widetilde{\theta}} + \widetilde{C}(\widetilde{\theta}, \dot{\widetilde{\theta}})\dot{\widetilde{\theta}} + \widetilde{g}(\widetilde{\theta}) = \begin{bmatrix} 0 \\ \tau_a \\ 0 \end{bmatrix} + \begin{bmatrix} J_o^T(x_o) \\ J_a^T(q_a) \\ J_p^T(q_p) \end{bmatrix} h, \tag{9.10}$$

where $\widetilde{g}(\widetilde{\theta}) = [g_o(x_o)^T \; g_{AP}(\widetilde{q})^T]^T$, $g_{AP} = P_{AP}[g_1^T(q_1) \; \cdots \; g_2^T(q_2)]^T$,

$$\widetilde{M}(\widetilde{\theta}) = \begin{bmatrix} M_o(x_o) & 0 \\ 0 & M_{AP}(\widetilde{q}) \end{bmatrix}, \quad \widetilde{C}(\widetilde{\theta}, \dot{\widetilde{\theta}}) = \begin{bmatrix} C_o(x_o, \dot{x}_o) & 0 \\ 0 & C_{AP}(\widetilde{q}, \dot{\widetilde{q}}) \end{bmatrix},$$

$$M_{AP}(\widetilde{q}) = P_{AP} \begin{bmatrix} M_1(q_1) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & M_m(q_m) \end{bmatrix} P_{AP}^T, \quad \text{and}$$

$$C_{AP}(\widetilde{q}, \dot{\widetilde{q}}) = P_{AP} \begin{bmatrix} C_1(q_1, \dot{q}_1) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & C_m(q_m, \dot{q}_m) \end{bmatrix} P_{AP}^T.$$

Taking into account the orthogonal decomposition of the applied forces' projection, we can rewrite Eq. 9.10 as:

$$\widetilde{M}(\widetilde{\theta})\ddot{\widetilde{\theta}} + \widetilde{C}(\widetilde{\theta}, \dot{\widetilde{\theta}})\dot{\widetilde{\theta}} + \widetilde{g}(\widetilde{\theta}) = \tau_v + \widetilde{W}^T(\widetilde{\theta})h_{os}, \tag{9.11}$$

where τ_v is the auxiliary control input given by

$$\tau_v = \begin{bmatrix} W^T h_{om} \\ \tau_a + J_a^T(q_a) J_{oq}^{\square T}(x_o) h_{om} \\ J_p^T(q_p) J_{oq}^{\square T}(x_o) h_{om} \end{bmatrix},$$

and $\widetilde{W}(\widetilde{\theta}) = \ker W \quad J_{oq}^{\square 1}(x_o) J_a(q_a) \quad J_{oq}^{\square 1}(x_o) J_p(q_p)$ is a Jacobian matrix. If the auxiliary control input is partitioned in three vectors, $\tau_{v1} = W^T(x_o) h_{om}$, $\tau_{v2} = \tau_a + J_a^T(q_a) J_{oq}^{\square T}(x_o) h_{om}$, and $\tau_{v3} = J_p^T(q_p) J_{oq}^{\square T}(x_o) h_{om}$, the applied torque in the active joints can be computed as:

$$\tau_a = \tau_{v2} - J_a^T(q_a) J_{oq}^{\square T}(x_o) \begin{bmatrix} W^T \\ J_p^T(q_p) J_{oq}^{\square T}(x_o) \end{bmatrix}^{\#} \begin{bmatrix} \tau_{v1} \\ \tau_{v3} \end{bmatrix}. \tag{9.12}$$

Considering the kinematic constraints (9.9) and multiplying Eq. 9.11 by $\widetilde{B}^T(x_o)$, the dynamic equation of the underactuated cooperative manipulator is given by:

$$\widetilde{M}(x_o)\ddot{x}_o + \widetilde{C}(x_o, \dot{x}_o)\dot{x}_o + \widetilde{g}(x_o) = \widetilde{\tau}_v, \tag{9.13}$$

where

$$\begin{aligned} \widetilde{M}(x_o) &= \widetilde{B}^T(x_o)\widetilde{M}(\widetilde{\theta})\widetilde{B}(x_o), \\ \widetilde{C}(x_o, \dot{x}_o) &= \widetilde{B}^T(x_o)\left(\widetilde{M}(\widetilde{\theta})\dot{\widetilde{B}}(x_o) + \widetilde{C}(\widetilde{\theta}, \dot{\widetilde{\theta}})\widetilde{B}(x_o)\right), \\ \widetilde{g}(x_o) &= \widetilde{B}^T(x_o)\widetilde{g}(\widetilde{\theta}), \\ \widetilde{\tau}_v &= \widetilde{B}^T(x_o)\tau_v. \end{aligned}$$

9.4 Motion and Squeeze Force Control

From the control paradigm for cooperative manipulators introduced in [8], the position and squeeze force control problems can be decomposed and solved independently. In this case, the applied torque can be computed by:

$$\tau = \tau_{mg} + \tau_s,$$

where τ_{mg} are torques generated by the position controller and τ_s are torques generated by the squeeze force controller. In this chapter, τ_{mg} is given by (9.7) if all joints of every manipulator are actuated. When at least one joint in the entire cooperative system is underactuated, we use $\tau_{mg} = P_{AP}^{\dagger T}\tau_a^T \; 0^T$, with τ_a given by (9.12). In Sect. 9.5, the dynamic equations (9.8) and (9.13) are used to design robust controllers for position control of cooperative manipulators, taking into account parametric uncertainties and external disturbances in the manipulator and in the object.

In the case of fully-actuated cooperative manipulators, Wen and Kreutz–Delgado [8] propose an integral squeeze force controller with torque given by:

$$\tau_s = D^T(\theta)\left[h_{sc}^d + K_i\int_{t_0}^{t}(h_{sc}^d - h_{sc})dt\right], \qquad (9.14)$$

where h_{sc} is the vector of squeeze forces unaffected by the motion, h_{sc}^d is the vector of desired squeeze forces, K_i is a positive de nite matrix, and

$$D(\theta) = \begin{bmatrix} J_{o_1}^{\dagger}(x_o)J_1(q_1) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & J_{o_m}^{\dagger}(x_o)J_m(q_m) \end{bmatrix}.$$

In the case of underactuated cooperative manipulators, Eq. 9.14 can be partitioned as:

$$\begin{bmatrix} \tau_{sa} \\ 0 \end{bmatrix} = \begin{bmatrix} D_{sa}^T(\widetilde{\theta}) \\ D_{sp}^T(\widetilde{\theta}) \end{bmatrix} W_c^T \gamma_s, \qquad (9.15)$$

where $D_{sa}(\widetilde{\theta}) \; D_{sp}(\widetilde{\theta}) = D(\theta)P_{AP}$, W_c^T is the full rank matrix that projects the null space of W^T, i.e., $Im(W_c^T) = X_S$, and γ_s is the squeeze force control variable. Note that n_p constraints are imposed in the components of γ_s since it is not possible to apply torque on the passive joints ($\tau_{sp} = 0$). As the manipulators considered here are nonredundant, not all components of γ_s can be independently controlled.

The vector γ_s is partitioned in two parts: the independently-controlled components $\gamma_{sc} \in \Re^{n_e}$, where $n_e = n(m-1) - n_p$ if $n_a > n$ and $n_e = 0$ if $n_a < n$; and

the uncontrolled components $\gamma_{sn} \in \mathbb{R}^{n_p}$. Note that if $n_a < n$, none of the components of γ_s can be controlled. The squeeze force controller is given by:

$$\gamma_{sc} = \Gamma_{sc}^d + K_{is} \int_{t_0}^{t} (\Gamma_{sc}^d - \Gamma_{sc}) dt, \qquad (9.16)$$

where Γ_{sc}^d is the desired value for γ_{sc}, Γ_{sc} is the vector of measured squeeze forces, and K_{is} is a positive definite matrix. γ_{sn} is computed from the constraints as a function of γ_{sc}. The torque applied in the active joints is related to the squeeze force control (Eq. 9.15) as:

$$\tau_{sa} = D_{sa}^T(\tilde{\theta}) W_c^T \gamma_{sc}. \qquad (9.17)$$

9.5 Nonlinear $\mathcal{H}_\infty$ Control

In this section the nonlinear H_1 control solutions presented in Chaps. 3 and 5 for fully-actuated and single underactuated manipulators, respectively, are applied to cooperative manipulators. The main idea is to consider the reduced-order models (9.8) and (9.13), which represent the dynamic model of cooperative manipulators in a similar way as it is commonly used in the representation of single manipulators.

9.5.1 Control Design via Quasi-LPV Representation

We develop in this section the quasi-LPV representations of fully actuated and underactuated cooperative manipulators based on the following dynamic equation:

$$\hat{M}_0(x_o)\ddot{x}_o + \hat{C}_0(x_o, \dot{x}_o)\dot{x}_o + \hat{g}_0(x_o) + \hat{\tau}_d = \hat{\tau}_v, \qquad (9.18)$$

where $\hat{M}_0(x_o) = \overline{M}_0(x_o)$, $\hat{C}_0(x_o, \dot{x}_o) = \overline{C}_0(x_o, \dot{x}_o)$, $\hat{g}_0(x_o) = \overline{g}_0(x_o)$, and $\hat{\tau}_v = \overline{\tau}_v$ if the manipulators are fully-actuated (Eq. 9.8); or $\hat{M}_0(x_o) = \widetilde{M}_0(x_o)$, $\hat{C}_0(x_o, \dot{x}_o) = \widetilde{C}_0(x_o, \dot{x}_o)$, $\hat{g}_0(x_o) = \widetilde{g}_0(x_o)$, and $\hat{\tau}_v = \widetilde{\tau}_v$ if any of the manipulators is underactuated (Eq. 9.13). The index 0 indicates nominal values for the matrices and vectors. $\hat{\tau}_d$ is the vector of parametric uncertainties and external disturbances in the manipulators and load. The state tracking error is defined as:

$$\tilde{x} = \begin{bmatrix} x_o - x_o^d \\ \dot{x}_o - \dot{x}_o^d \end{bmatrix} = \begin{bmatrix} \tilde{x}_o \\ \dot{\tilde{x}}_o \end{bmatrix}, \qquad (9.19)$$

where x_o^d and $\dot{x}_o^d \in \Re^n$ are the reference trajectory and desired velocity of the load, respectively. The quasi-LPV representations of cooperative manipulators are found using Eqs 9.18 and 9.19:

$$\dot{\tilde{x}} = A(x_o, \dot{x}_o)\tilde{x} + Bu + Bw, \tag{9.20}$$

with $w = \hat{M}_0^{-1}(x_o)\hat{\tau}_d$, $B = [0 \;\; I_n]^T$, and

$$A(x_o, \dot{x}_o) = \begin{bmatrix} 0 & I_n \\ 0 & -\hat{M}_0^{-1}(x_o)\hat{C}_0(x_o, \dot{x}_o) \end{bmatrix}.$$

From this equation, the variable $\hat{\tau}_v$ can be represented as:

$$\hat{\tau}_v = \hat{M}_0(x_o)(\ddot{x}_o^d + u) + \hat{C}_0(x_o, \dot{x}_o)\dot{x}_o^d + \hat{g}_0(x_o).$$

Although the matrices $\hat{M}_0(x_o)$ and $\hat{C}_0(x_o, \dot{x}_o)$ explicitly depend on the load position, x_o, we can consider it as function of the position error, $\tilde{x}_o$. Hence, Eq. 9.20 is a quasi-LPV representation of fully-actuated and underactuated cooperative manipulators.

9.5.2 Control Design via Game Theory

In this section, game theory is used to solve the H_∞ control problem of cooperative manipulators. The solution is based on the results presented in Chap. 3. From (9.19), after the state transformation given by:

$$\tilde{z} = \begin{bmatrix} \tilde{z}_1 \\ \tilde{z}_2 \end{bmatrix} = T_0\tilde{x} = \begin{bmatrix} T_1 \\ T_2 \end{bmatrix}\tilde{x} = \begin{bmatrix} I & 0 \\ T_{11} & T_{12} \end{bmatrix}\begin{bmatrix} \tilde{x}_o \\ \dot{\tilde{x}}_o \end{bmatrix}, \tag{9.21}$$

where $T_{11}, \; T_{12} \in \Re^{n \times n}$ are constant matrices to be determined, the dynamic equation of the state tracking error becomes

$$\dot{\tilde{x}} = A_T(\tilde{x}, t)\tilde{x} + B_T(\tilde{x}, t)u + B_T(\tilde{x}, t)w, \tag{9.22}$$

with $w = \hat{M}_0(x_o)T_{12}\hat{M}_0^{-1}(x_o)\hat{\tau}_d$,

$$A_T(\tilde{x}, t) = T_0^{-1}\begin{bmatrix} -T_{12}^{-1}T_{11} & T_{12}^{-1} \\ 0 & -\hat{M}_0^{-1}(x_o)\hat{C}_0(x_o, \dot{x}_o) \end{bmatrix}T_0,$$

$$B_T(\tilde{x}, t) = T_0^{-1}\begin{bmatrix} 0 \\ \hat{M}_0^{-1}(x_o) \end{bmatrix}.$$

The relationship between the auxiliary control input, $\hat{\tau}_v$, and the control input, u, is given by:

$$\hat{\tau}_v = \hat{M}_0(x_o)(\ddot{x}_o^c + u) + \hat{C}_0(x_o, \dot{x}_o)\dot{x}_o + \hat{g}_0(x_o), \tag{9.23}$$

with $\ddot{x}_o^c = \ddot{x}_o^d - T_{12}^{-1}T_{11}\dot{\tilde{x}}_o - T_{12}^{-1}\widehat{M}_0^{-1}(x_o)(\widehat{C}_0(x_o,\dot{x}_o)B^T T_0\tilde{x} - u)$. The procedure for finding the solution for the control input u that ensures H_1 performance follows the guidelines presented in Chap. 3 for single manipulators.

9.6 Neural Network-Based Adaptive Nonlinear $\mathcal{H}_\infty$ Control

In the previous section we assumed that the dynamic model of the cooperative manipulator system is well-known. In this section we develop another control strategy based on the estimation of the uncertain part of the dynamic model through a neural network-based adaptive control law. Consider again the kinematic constraints (9.1) and the dynamic model for fully-actuated cooperative manipulators (9.6):

$$M(x_o)B(x_o)\ddot{x}_o + \left(M(x_o)\dot{B}(x_o) + C(x_o,\dot{x}_o)B(x_o)\right)\dot{x}_o + g(x_o) + \tau_d = \tau_v + \overline{W}^T(x_o)h_{os}. \tag{9.24}$$

Consider also a bounded desired trajectory for the load $x_o^d \in \Re^n$, and a bounded desired squeeze force $h_{os}^d \in \Re^{nm}$, and define the following auxiliary variable:

$$q_r = B(x_o)(\Lambda_o e_o + \dot{x}_o^d) - \eta E_2 e_f,$$

where $q_r \in \Re^{n(m+1)}$, $e_o = x_o^d - x_o$, Λ_o is a symmetric positive definite matrix, $\eta > 0$, $E_2 = [0\ I_{nm}]^T \in \Re^{n(m+1)\times nm}$, and $e_f \in \Re^{nm}$ is the output of the following stable filter

$$\dot{e}_f + \eta e_f = -\lambda_f E_2^T \overline{W}^T(x_o)\tilde{h}_{os}, \tag{9.25}$$

with a symmetric positive definite matrix Λ_f and the squeeze force error $\tilde{h}_{os} = h_{os}^d - h_{os}$.

A composite error signal can be defined as:

$$s = q_r - B(x_o)\dot{x}_o = B(x_o)(\Lambda_o e_o + \dot{e}_o) - \eta E_2 e_f,$$

where $s \in \Re^{n(m+1)}$. Another representation of the composite error can be obtained by applying the stable filter (9.25) and defining the error terms:

$$e_1 = [e_o^T \quad (J^T(x_o)J_{oq}^{-T}(x_o)\tilde{h}_{os})^T]^T,$$

and $e_2 = [\dot{e}_o^T \quad \dot{e}_f^T]^T$:

$$\begin{aligned} s &= L\left[E_1(\Lambda_o e_o + \dot{e}_o) + E_2\left(\lambda_f E_2^T \overline{A}^T(x_o)\tilde{h}_{os} + \dot{e}_f\right)\right] \\ &= L(\Lambda e_1 + e_2), \end{aligned} \tag{9.26}$$

where $L = [B(x_o) \; E_2] \in \mathbb{R}^{n(m+1)\times n(m+1)}$, $E_1 = [I_n \; 0]^T \in \mathbb{R}^{n(m+1)\times n}$ and $\Lambda = diag[\Lambda_o, \Lambda_f]$. From (9.24) and (9.26), the error dynamic model is given by:

$$\begin{aligned} M(x_o)\dot{s} &= M(x_o)\dot{q}_r - M(x_o)\left(B(x_o)\ddot{x}_o + \dot{B}(x_o)\dot{x}_o\right) \\ &= -C(x_o, \dot{x}_o)s + F_0(x_e) + \Delta F(x_e) - \overline{W}^T(x_o)h_{os} + \tau_d - \tau_v, \end{aligned} \tag{9.27}$$

where $x_e = [x_o^T \; \dot{x}_o^T \; q_r^T \; \dot{q}_r^T]^T$, $F_0(x_e) = M_0(x_o)\dot{q}_r + C_0(x_o, \dot{x}_o)q_r + g_0(x_o)$, and $\Delta F(x_e) = \Delta M(x_o)\dot{q}_r + \Delta C(x_o, \dot{x}_o)q_r + \Delta g(x_o)$.

The terms $M_0(x_o)$, $C_0(x_o, \dot{x}_o)$, and $g_0(x_o)$ represent the nominal values of the matrices $M(x_o)$, $C(x_o, \dot{x}_o)$, and $g(x_o)$, respectively. The parametric uncertainties are represented by $\Delta M(x_o)$, $\Delta C(x_o, \dot{x}_o)$, and $\Delta g(x_o)$.

Following the development of the neural network-based adaptive control strategy presented in Chap. 4, a set of k ($k = 1, \ldots, n$) neural networks $\Delta F(x_e, \Theta_k)$ is used to approximate the uncertain terms $\Delta F(x_e)$ in (9.27). Each neural network is composed of nonlinear neurons in the hidden layer and linear neurons in the input and output layers, with adjustable parameters Θ_k in the output layers. The single-output neural networks are of the form:

$$\Delta F_k(x_e, \Theta_k) = \sum_{i=1}^{p_k} H\left(\sum_{j=1}^{5n} w_{ij}^k x_{ej} + m_i^k\right)\Theta_{ki} = \xi_k^T \Theta_k, \tag{9.28}$$

with

$$\xi_k = \begin{bmatrix} H\left(\sum_{j=1}^{5n} w_{1j}^k x_{ej} + m_1^k\right) \\ \vdots \\ H\left(\sum_{j=1}^{5n} w_{p_k j}^k x_{ej} + m_{p_k}^k\right) \end{bmatrix}, \quad \Theta_k = \begin{bmatrix} \Theta_{k1} \\ \vdots \\ \Theta_{kp_k} \end{bmatrix},$$

where p_k is the number of neurons in the hidden layer. The weights w_{ij}^k and the biases m_i^k for $1 \le i \le p_k$, $1 \le j \le 5n$ and $1 \le k \le n$ are assumed to be constant and specified by the designer. $H(\cdot)$ is selected to be a hyperbolic tangent function. The complete neural network can be represented as:

$$\begin{aligned} \Delta F(x_e, \Theta) &= \begin{bmatrix} \Delta F_1(x_e, \Theta_1) \\ \vdots \\ \Delta F_n(x_e, \Theta_n) \end{bmatrix} = \begin{bmatrix} \xi_1^T \Theta_1 \\ \vdots \\ \xi_n^T \Theta_n \end{bmatrix} \\ &= \begin{bmatrix} \xi_1^T & 0 & \ldots & 0 \\ 0 & \xi_2^T & \vdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \xi_n^T \end{bmatrix} \begin{bmatrix} \Theta_1 \\ \Theta_2 \\ \vdots \\ \Theta_n \end{bmatrix} = \Xi\Theta. \end{aligned} \tag{9.29}$$

To guarantee the stability of the H_1 controller we consider two fundamental assumptions proposed in [1]:

(1) There exists an optimal parameter value $\Theta^* \in \Omega_\Theta$ such that $\Delta F(x_e, \Theta^*)$ approximates $\Delta F(x_e)$ as closely as possible, where Ω_Θ is a pre-assigned constraint region.
(2) The approximation error $\delta F(x_e) = \Delta F(x_e) - \Delta F(x_e, \Theta^*)$ must be bounded by a state-dependent function; that is, there exists a function $k(x_e) > 0$ such that $|\delta F(x_e)_i| \le k(x_e)$, for all $1 \le i \le n$.

Based on these assumptions, the error dynamics can be rewritten as:

$$M(x_o)\dot{s} = -C(x_o, \dot{x}_o)s + F_0(x_e) + \Delta F(x_e, \Theta^*) + \delta F(x_e) - \overline{W}^T(x_o)h_{os} + \tau_d - \tau_v. \tag{9.30}$$

The nonlinear H_1 adaptive neural network control problem for cooperative manipulators can then be formulated as follows: given a level of attenuation γ and an auxiliary control input τ_v such that the following H_1 performance index is achieved:

$$\int_0^T s^T \Psi s\,dt \le \int_0^T e^T Q e\,dt - \frac{1}{2}s^T(0)M(x_o(0))s(0) + \frac{1}{2}\rho e_f^T(0)\Lambda_f^{-1}e_f(0) + \tilde{\Theta}^T(0)Z\tilde{\Theta}(0) + \gamma^2\int_0^T \tau_d^T\tau_d\,dt, \tag{9.31}$$

where $e = \left[e_1^T\ e_2^T\right]^T$, Z is a symmetric positive definite matrix, $\tilde{\Theta} = \Theta^* - \Theta$ denotes the neural network parameter estimation error, τ_d is a square-integrable torque disturbance ($\tau_d \in L_2$), and

$$Q = \begin{bmatrix} \Lambda^T\Psi\Lambda & \Lambda^T\Psi \\ \Psi\Lambda & \Psi \end{bmatrix}, \tag{9.32}$$

with $\Psi = \Psi^T > 0$ and $\Lambda = \mathrm{diag}\{\Lambda_o, \Lambda_f\}$.

Theorem *Consider a cooperative manipulator described by Eq. 9.4. If the control law is defined as:*

$$\dot{\Theta} = Proj\{S^{*T}\Xi^T s\}, \tag{9.33}$$

$$\tau_v = F_0(x_e) + Ks + \Xi\Theta - \overline{W}^T(x_o)h_{os} + \frac{\rho}{\eta}\overline{W}^T(x_o)\tilde{h}_{os} + \tau_s, \tag{9.34}$$

with $\tau_s = k(x_e)\,\mathrm{sgn}(s)$ *and*

$$Proj(S^{T}\Xi^{T}s) = \begin{cases} S^{T}\Xi^{T}s, & \text{if } \Theta^{T}\Theta < M \text{ or } (\Theta^{T}\Theta = M \text{ and } \Theta^{T}S^{T}\Xi^{T}s \le 0), \\ S^{T}\Xi^{T}s - \dfrac{\partial\Theta^{T}\Theta - M)\Theta^{T}S^{T}\Xi^{T}s}{\partial\Theta^{T}\Theta}\Theta, & \text{otherwise} \end{cases}$$

and $K = diag(K_o, K_1, \ldots, K_m)$, *with symmetric positive definite matrices* K_o, $K_1, \ldots, K_m$, *and* $Proj(S^{T}\Xi^{T}s)$ *is a projection algorithm, then the closed-loop error system satisfies*:

(1) $e_o, \dot{e}_o$, *and* $\ddot{e}_o \in L_1$, $e_f, \dot{e}_f$, *and* $\tilde{h}_{os} \in L_1$, *and* $\Theta \in \Omega_\Theta$.
(2) *The* H_1 *performance index* (9.31) *is achieved if* K_i *is selected as* $K_i = P_i + (1/4\gamma^2)I_n$ *with symmetric positive definite matrix* P_i.
(3) *If* $d \in L_2$, *then the motion tracking errors* $e_o, \dot{e}_o$ *converge to zero as* $t \to \infty$.

Proof Consider the Lyapunov function:

$$V = \frac{1}{2}s^{T}Ms + \frac{1}{2}\rho e_f^{T}\Lambda_f^{-1}e_f + \frac{1}{2}\tilde{\Theta}^{T}S\tilde{\Theta}.$$

The time derivative of V along the error dynamics (9.27) and control law (9.34) is given by:

$$\begin{aligned} \dot{V} &= s^{T}M\dot{s} + \frac{1}{2}s^{T}\dot{M}s + \rho e_f^{T}\Lambda_f^{-1}\dot{e}_f + \dot{\tilde{\Theta}}^{T}S\tilde{\Theta} \\ &= s^{T}\left(-Cs - Ks + \Xi\tilde{\Theta} + \delta F(x_e) - \frac{\rho}{\eta}\overline{W}^{T}(x_o)\tilde{h}_{os}\right) + \frac{1}{2}s^{T}\dot{M}s \\ &\quad + \rho e_f^{T}\Lambda_f^{-1}\left(-\eta e_f - \Lambda_f E_2^{T}\overline{W}_2^{T}(x_o)\tilde{h}_{os}\right) + \dot{\tilde{\Theta}}^{T}S\tilde{\Theta} - s^{T}\tau_s + s^{T}\tau_d. \end{aligned}$$

Since $-(\rho/\eta)s^{T}\overline{W}^{T}(x_o)\tilde{h}_{os} - \rho e_f^{T}E_2^{T}\overline{W}^{T}(x_o)\tilde{h}_{os} = 0$ and $(1/2)\dot{M} - C$ is a skew-symmetric matrix, $\dot{V}$ becomes:

$$\dot{V} = -s^{T}Ks + s^{T}\Xi\tilde{\Theta} + s^{T}\delta F(x_e) - \eta\rho e_f^{T}\Lambda_f^{-1}e_f + \dot{\tilde{\Theta}}^{T}S\tilde{\Theta} - s^{T}\tau_s + s^{T}\tau_d. \quad (9.35)$$

From the definition of the update law (9.33), where the projection algorithm is used, we can show that:

$$\dot{\tilde{\Theta}}^{T}S\tilde{\Theta} + s^{T}\Xi\tilde{\Theta} \le 0, \quad (9.36)$$

and $\Theta(t) \in \Omega_\Theta$ for all $t \ge 0$ if $\Theta(0) \in \Omega_0$, with $\Omega_0 = \{\Theta : \Theta^{T}\Theta \le M\}$. Taking into account the control law τ_s and assumption (2), the following inequality can be guaranteed:

$$s^T(\tau_s + \delta F(x_e)) = s^T(k(x_e)sgn(s) + \delta F(x_e)) \le k(x_e)\sum_{i=1}^{n(m+1)}|s_i| + \sum_{i=1}^{n(m+1)}|(\delta F(x_e))_i||s_i| \le 0. \quad (9.37)$$

Substituting Eqs. (9.36) and (9.37) into Eq. (9.35), we obtain:

$$\dot{V} \le -s^T K s - \eta\rho e_f^T \Lambda_f^{-1} e_f + s^T \tau_d. \quad (9.38)$$

According to the inequality $s^T\tau_d \le (1/\gamma^2)s^T s + \gamma^2\tau_d^T\tau_d$, Eq. (9.38) leads to:

$$\dot{V} \le -s^T P s + \gamma^2\tau_d^T\tau_d, \quad (9.39)$$

where the control gains $K_i = P_i + (1/\gamma^2)I_n$ have been used, and $P = \mathrm{diag}[P_o, P_1, \ldots, P_m]$. Substituting (9.26) in (9.39), we obtain:

$$\dot{V} \le -e^T\begin{bmatrix}\Lambda^T L^T P L\Lambda & \Lambda^T L^T P L \\ L^T P L\Lambda & L^T P L\end{bmatrix} e + \gamma^2\tau_d^T\tau_d \le -e^T Q e + \gamma^2\tau_d^T\tau_d. \quad (9.40)$$

Considering $\Psi = L^T P L$, matrix Q satisfies Eq. (9.32) and is symmetric positive semi-definite. By integrating the above inequality, the closed-loop system satisfies the following H_1 performance index:

$$\int_0^T e^T Q e dt \le V(0) - V(t) + \gamma^2\int_0^T \tau_d^T\tau_d dt \le \frac{1}{2}s^T(0)M(0)s(0) + \frac{1}{2}\rho e_f^T(0)\Lambda_f^{-1}e_f(0) + \tilde{\Theta}^T(0)Z\tilde{\Theta}(0) + \gamma^2\int_0^T \tau_d^T\tau_d dt. \quad (9.41)$$

From Eq. (9.39), $\dot{V}$ is convergent, which implies $s \in L_1$, and, therefore, that e_o, $\dot{e}_o$, and $e_f \in L_1$. The squeeze force error can be expressed as a function of the neural network approximation parameters ($\Delta\hat{M}$, $\Delta\hat{C}$, and $\Delta\hat{g}$) as:

$$\eta\overline{W}(x_o)M^{-1}(x_o)\left(M_0(x_o) + \Delta\hat{M}(x_o)\right)E_2\Lambda_f E_2^T + \rho/(\eta^2(n(m+1)))\overline{W}^T(x_o)\tilde{h}_{os} = z + \eta^2\overline{W}(x_o)M^{-1}(x_o)M_0(x_o) + \Delta\hat{M}(x_o)E_2 e_f + \overline{W}(x_o)\dot{B}(x_o)\dot{x}_o, \quad (9.42)$$

with

$$z = \overline{W}(x_o)M^{-1}(x_o)\Big(C(x_o, \dot{x}_o)s + Ks + \Delta\hat{C}q_r + \Delta\hat{g} + \tau_d - \tau_s - (M_0(x_o) + \Delta\hat{M}(x_o))\left(\dot{B}(x_o)(\Lambda_o e_o + \dot{x}_o^d) + B(x_o)(\Lambda_o\dot{e}_o + \ddot{x}_o^d)\right)\Big).$$

Since all terms on the right-hand side of Eq. 9.42 are bounded, a proper value of ρ assures $\tilde{h}_{os}$ is bounded and, therefore, $\dot{e}_f$, $\dot{s}$, and $\ddot{e}_o \in L_1$.

If a square-integrable torque disturbance is assumed, i.e., $\tau_d \in L_2$, then $s \in L_2$ by integrating Eq. 9.39. By Barbalat's lemma, $\lim_{t \to \infty} s(t) = 0$, since $s \in L_2$ and $\dot{s} \in L_1$. Hence, $\lim_{t \to \infty} \dot{e}_o, e_o, e_f = 0$. □

The algorithm $Proj(\cdot)$ was originally defined in [3]. It guarantees that $\Theta(t) \in \Omega_\Theta$ for all t, where $\Omega_\Theta = \{\Theta : \Theta^T\Theta \leq M + \delta\}$, for some $M > 0$ and $\delta > 0$ a pre-assigned constraint region for Θ.

We now consider the underactuated case, where the cooperative manipulator is characterized by n_a active joints and n_p passive joints. Considering the kinematic constraints (9.9), the dynamic equation of the underactuated cooperative manipulator is given by:

$$\widetilde{M}(x_o)\widetilde{B}(x_o)\ddot{x}_o + \left(\widetilde{M}(x_o)\dot{\widetilde{B}}(x_o) + \widetilde{C}(x_o, \dot{x}_o)\widetilde{B}(x_o)\right)\dot{x}_o + \widetilde{g}(x_o) + \tau_d = \widetilde{\tau}_v + \widetilde{W}^T(x_o)h_{os}. \tag{9.43}$$

The update and control laws for underactuated cooperative manipulators can now be defined as:

$$\dot{\widetilde{\Theta}} = Proj(\widetilde{S}^{\square T}\widetilde{\Xi}^T s),$$

$$\widetilde{\tau}_v = \widetilde{F}_0(x_e) + Ks + \widetilde{\Xi}\widetilde{\Theta} - \widetilde{W}^T(x_o)h_{os} + \frac{\rho}{\eta}\widetilde{W}^T(x_o)\tilde{h}_{os} + \tau_s,$$

where $\widetilde{F}_0(x_e) = \widetilde{M}_0(x_o)\dot{q}_r + \widetilde{C}_0(x_o, \dot{x}_o)q_r + \widetilde{g}_0(x_o)$ and $\Delta\widetilde{F}(x_e, \widetilde{\Theta}) = \widetilde{\Xi}\widetilde{\Theta}$ is the neural network used to approximate $\Delta\widetilde{F}(x_e) = \Delta\widetilde{M}(x_o)\dot{q}_r + \Delta\widetilde{C}(x_o, \dot{x}_o)q_r + \Delta\widetilde{g}(x_o)$.

In the fully-actuated cooperative system, m actuated joints are needed to control the components of the motion of the load and $nm - 1$ actuated joints are utilized to control the squeeze forces. However, when underactuated cooperative manipulators are considered, not all squeeze force components can be controlled since some degrees of actuation have been lost.

Consider the squeeze force error defined in (9.25). The dimension of $\tilde{h}_{os}$ is nm, and since the dimension of s is $n(m-1)$, it is possible to write $\tilde{h}_{os} = W_c^T\gamma_s$, where $\gamma_s \in \mathbb{R}^{n(m-1)}$ and $W_c^T \in \mathbb{R}^{nm \times n(m-1)}$ is the full-rank matrix that projects the null space of W^T, that is, $\mathrm{Im}(W_c^T) = X_s$. Hence, the $n(m-1)$-dimensional vector γ_s is a variable to be controlled.

The squeeze force error term $(\rho/\eta)\tilde{W}^T(x_o)\tilde{h}_{os}$, can be described as:

$$\begin{bmatrix} \tau_{os} \\ \tau_{as} \\ \tau_{ps} \end{bmatrix} = (\rho/\eta) \begin{bmatrix} W^T \\ J_a^T(x_o) J_{oq}^{\square T}(x_o) \\ J_p^T(x_o) J_{oq}^{\square T}(x_o) \end{bmatrix} W_c^T \gamma_s,$$

where τ_{os}, τ_{as}, τ_{ps} are the contributions of the squeeze force error in the components of the auxiliary control input $\tilde{\tau}_v$. By imposing that $\tau_{ps} = 0$, since it is assumed that no actuation occurs at passive joints, n_p constraints are created in the components of γ_s:

$$J_p^T(x_o) J_{oq}^{\square T}(x_o) W_c^T \gamma_s = 0. \tag{9.44}$$

Thus, if the manipulators are not kinematically redundant, only n_e components of γ_s can be independently controlled, where

$$n_e = \begin{cases} n(m-1) - n_p, & \text{if } n_a > n, \\ n_e = 0, & \text{otherwise} \end{cases}$$

The vector γ_s is now partitioned in the independently controlled components $\gamma_{sc} \in \mathbb{R}^{n_e}$ and in the uncontrolled components $\gamma_{sn} \in \mathbb{R}^{n_p}$. Note that if $n_a < n$, none of the components of γ_s can be controlled. In the control law implementation, the components of γ_{sn} are computed from the constraints (9.44) as a function of γ_{sc} (see more details in [7]).

9.7 Examples

In this section, the controllers presented in this chapter are implemented and tested in a cooperative manipulator system. We consider two desired trajectories: a straight line for the quasi-LPV and game theory-based controllers, and an arc of circumference for the neural network-based controller.

9.7.1 Design Procedures

The robust controllers can be designed using the Cooperative Manipulator Control Environment (CMCE). The controller can be selected with the menu *Controller* (see Fig. 9.1).

The controllers' gains are loaded in the control environment by executing the file `uarm_gains.m`. To change the controller behavior, the user may click on the button *Design Controller* and select the appropriate controller and fault configuration. Figure 9.2 shows the control design box for the NLH-quasi-LPV controller. The designer can select the following control parameters:

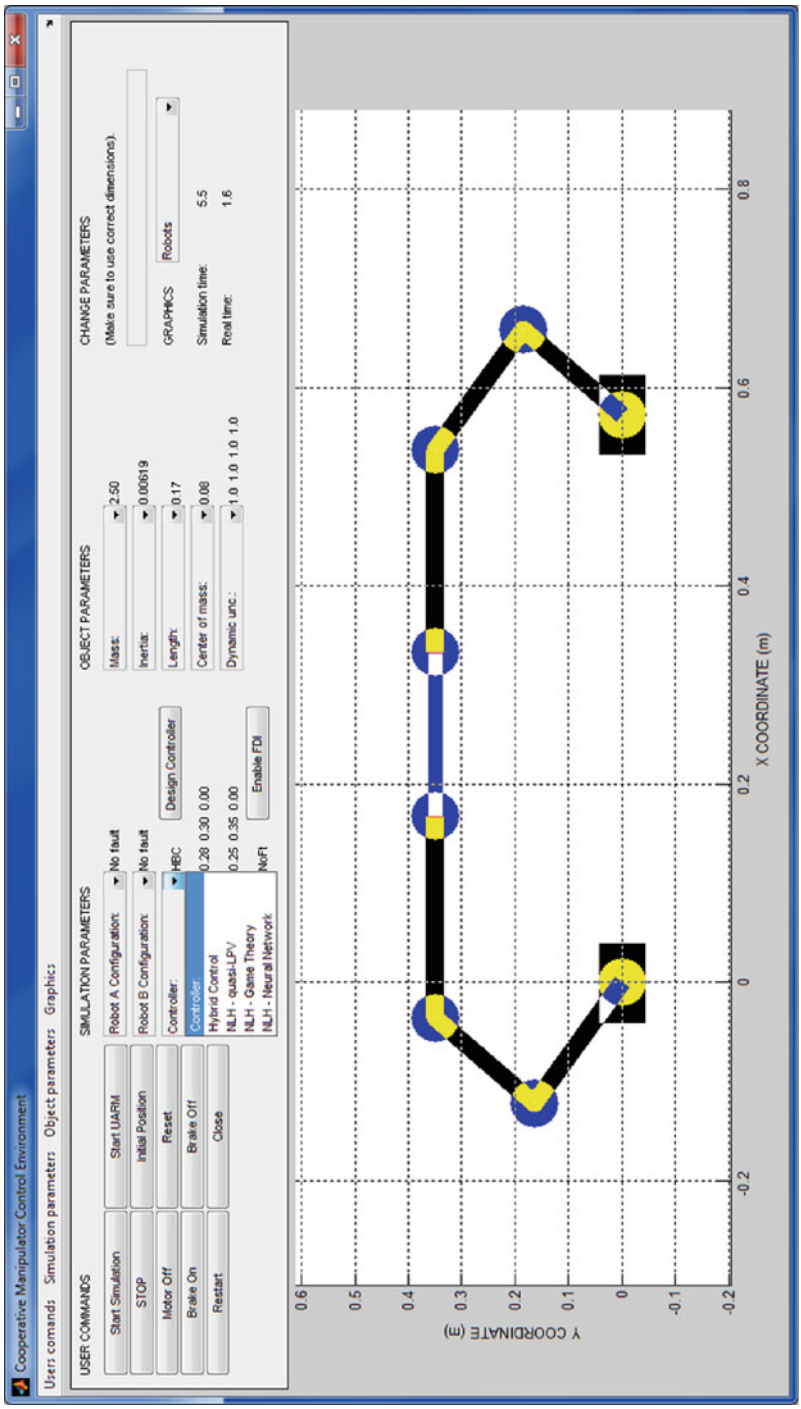


Fig. 9.1 Cooperative Manipulators Control Environment, controller menu

- L: Defines the number of points in the grid of the parameter set P.
- X_{max}: Defines the maximum absolute value for the first element of the parameter vector ρ, defined as the X coordinate of the object. The parameter range is defined as $\rho_1 \in [-X_{max}, X_{max}]$.
- Y_{max}: Defines the maximum absolute value for the second element ρ_2 of the Y coordinate of the object. The parameter range is defined as $\rho_2 \in [-Y_{max}, Y_{max}]$.
- *phimax*: Defines the maximum value for the third element ρ_3 of the orientation of the object ϕ. The parameter range is defined as $\rho_3 \in [-\phi_{max}, \phi_{max}]$.
- V_{max}: Defines the maximum value for the variation rates of the parameters ρ_i. It is a vector with three entries, one for each parameter.
- *gamma*: Defines the value for the attenuation level.
- *Ki*: Defines the squeeze force control parameter.

The following MATLAB® code implements the quasi-LPV and game theory controllers for the free-swinging joint fault configuration.

File: uarm_control.m

```
...

% H-infinity controller - quasi LPV
elseif (all(cont.type == 'LPV'))

    Bap = [eye(n/2); -inv(Jap)*(Jo)];
    dBap = [zeros(n/2);inv(Jap)*dJap*inv(Jap)*(Jo_est) - ...
        inv(J_est)*(dJo_est)];
    AAap = [Jo_est Jap];

    Map = Per*M_est*Per';
    Cap = Per*C_est*Per';

    MBap = Bap'*Map*Bap;
    WBap = inv(MBap);
    CBap = Bap'*Map*dBap + Bap'*Cap*Bap;

    derror = -derror;
    error = -error;
    e(:,i) = error;
    de(:,i)= derror;

    B_2 = [  zeros(3); eye(3) ];
    f1 = 1;
    f2(i) = error(1);
    f3(i) = error(2);
    f4(i) = cos(error(3));
    X_nlh = (f1*X1f + f2(i)*X2f + f3(i)*X3f + f4(i)*X4f);
    u(:,i) = -(B_2'*inv(X_nlh))*[error; derror];

    tauv(:,i) = pinv(Bap')*MBap*ddxo_d(:,i) + ...
       pinv(Bap')*CBap*dxo_d(:,i) + pinv(Bap')*MBap*u(:,i);
```

```
        tau_s = [-Dat_m']*fs_d;

        tau_at = (tauv(4:3+nat,i) - Jat'*inv(Joq_est')* ...
           pinv([A'; Jpa'*inv(Joq_est')])*[tauv(1:3,i); ...
           tauv(3+nat+1:9,i)]) + tau_s;

        tau_pa = zeros(npa,1);

    % H-infinity Controller - Game Theory
    elseif (all(cont.type == 'GTH'))

        Bap = [eye(n/2); -inv(Jap)*(Jo_est)];
        dBap = [zeros(n/2);inv(Jap)*dJap*inv(Jap)*(Jo_est) - ...
           inv(J_est)*(dJo_est)];
        AAap = [Jo_est Jap];

        Map = Per*M_est*Per';
        Cap = Per*C_est*Per';

        MBap = Bap'*Map*Bap;
        WBap = inv(MBap);
        CBap = Bap'*Map*dBap + Bap'*Cap*Bap;

        derror = -derror;
        error = -error;
        e(:,i) = error;
        de(:,i)= derror;

        u(:,i) = -inv(Rc)*Be'*T0*[error;derror];
        ddxo_c(:,i) = ddxo_d(:,i) - inv(T12)*T11*derror  - ...
           inv(T12)*inv(MBap)*(CBap*Be'*T0*[error;derror] - ...
           u(:,i));

        tauv(:,i) = pinv(Bap')*MBap*ddxo_c(:,i) + ...
           pinv(Bap')*CBap*dxo(:,i);

        tau_s = [-Dat_m']*fs_d;

        tau_at = (tauv(4:3+nat,i) - Jat'*inv(Joq_est')* ...
           pinv([A'; Jpa'*inv(Joq_est')])*[tauv(1:3,i); ...
           tauv(3+nat+1:9,i)]) + tau_s;
        tau_pa = zeros(npa,1);

        ...
```

9.7.2 Fully-Actuated Configuration

To validate the nonlinear H_1 control methods presented in the previous sections we apply them to the underactuated cooperative manipulator shown in Chap. 7, composed of two identical planar underactuated manipulators UARM. The workspace and the coordinate system for the cooperative manipulator are shown in Fig. 9.3; the load parameters are presented in Table 9.1. The kinematic and dynamic parameters of the manipulators can be found in Chap. 1

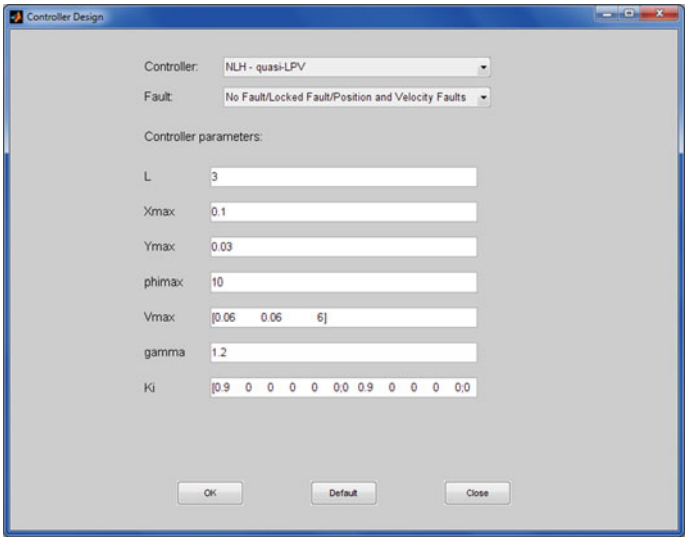


Fig. 9.2 Controller design box, NLH—quasi-LPV control

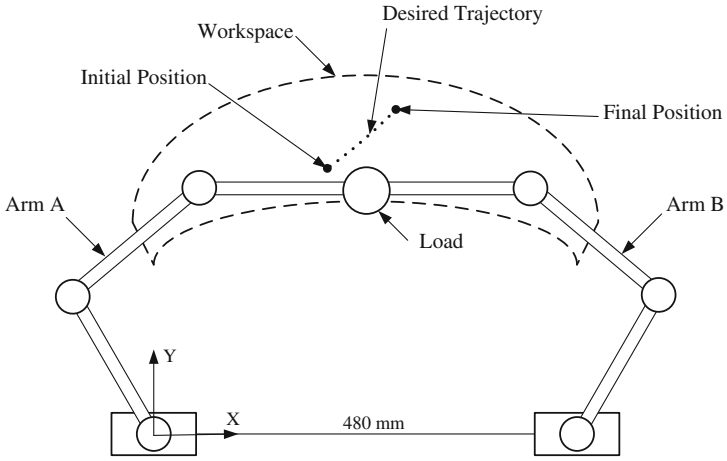


Fig. 9.3 Workspace and coordinate system of the cooperative manipulator UARM and desired trajectory for the model-based controllers

The goal is to move the center of mass of the load along a straight line in the Y plane from $x_o(0) = [-0.20\text{m}\ 0.35\text{m}\ 0]^T$ to $x_o^d(T) = [-0.25\text{m}\ 0.40\text{m}\ 0]^T$, where $T = 5.0$ s is the duration of the motion. The reference trajectory $x_o^d(t)$ is generated using a fifth-degree polynomial. The following external disturbances are introduced to verify the robustness of the proposed controllers:

Table 9.1 Load parameters

Parameter	Value
Mass	$m_o = 0.025$ kg
Length	$l_o = 0.092$ m
Center of mass	$a_o = 0.046$ m
Inertia	$I_o = 0.000023$ kg m^2

$$\tau_{d_1} = \begin{bmatrix} 0.01e^{-\frac{(r-2.5)^2}{8}} \sin(4\pi t) \\ -0.01e^{-\frac{(r-2.5)^2}{8}} \sin(5\pi t) \\ -0.01e^{-\frac{(r-2.5)^2}{8}} \sin(6\pi t) \end{bmatrix}, \quad \tau_{d_2} = \begin{bmatrix} 0.02e^{-\frac{(r-2.5)^2}{8}} \sin(4\pi t) \\ 0.02e^{-\frac{(r-2.5)^2}{8}} \sin(5\pi t) \\ 0.01e^{-\frac{(r-2.5)^2}{8}} \sin(6\pi t) \end{bmatrix}.$$

To apply the nonlinear controller via quasi-LPV representation, we select the parameters $\rho(\tilde{x})$ to represent the object position and orientation errors, i.e., ρ_1, ρ_2, and $\rho_3(\tilde{x}) = \tilde{x}_o$. The following quasi-LPV system matrices are considered:

$$A(\rho(x)) = A(\rho(\tilde{x})),$$
$$B_1(\rho(x)) = B,$$
$$B_2(\rho(x)) = B,$$
$$C_1(\rho(x)) = I_6,$$
$$C_2(\rho(x)) = 0,$$

where $A(\rho(\tilde{x}))$ and B are obtained from (9.20) with $\widehat{M}(x_o) = \overline{M}(x_o)$ and $\widehat{C}(x_o, \dot{x}_o) = \overline{C}(x_o, \dot{x}_o)$.

The compact set P is defined as $\rho \in [-0.1, 0.1]$ m $\times [-0.1, 0.1]$ m $\times [-\pi/9, \pi/9]$. The parameter variation rate is bounded by $|\dot{\rho}| \le [0.06$ m/s 0.06 m/s $\pi/6$ rad/s$]$. The basis functions for $X(\rho)$ are selected as $f_1(\rho(\tilde{x})) = 1$, $f_2(\rho(\tilde{x})) = \tilde{x}_{o_X}$, $f_3(\rho(\tilde{x})) = \tilde{x}_{o_Y}$ and $f_4(\rho(\tilde{x})) = \cos(\tilde{x}_{o_\phi})$, where $\tilde{x}_o = [\tilde{x}_{o_X}\ \tilde{x}_{o_Y}\ \tilde{x}_{o_\phi}]$, $\tilde{x}_{o_X}$ and $\tilde{x}_{o_Y}$ are the X and Y coordinate errors of the load, and $\tilde{x}_{o_\phi}$ is the orientation error. The parameter space was divided considering three points in the set. The best attenuation level is $\gamma = 1.25$.

For the nonlinear H_1 controller designed via game theory (Sect. 9.5.2), the attenuation level is $\gamma = 4.0$. The weighting matrices used are $Q_1 = 10 I_3$, $Q_2 = I_3$, $Q_{12} = 0$, and $R = I_3$. The desired values for the squeeze force are $f_{sc} = [0\ 0\ 0]^T$. The integral gains of the squeeze force controller are $K_{i_1} = 0.9 I_3$ and $K_{i_S} = 0.9 I_3$ for the fully-actuated and underactuated cases, respectively. The resulting Cartesian coordinates and orientation of the object are shown in Figs. 9.4 and 9.5.

Three performance indexes are used to compare the performance of the nonlinear H_1 controllers: the L_2 norm of the state vector $\|\tilde{x}\|_2$, the integral of the

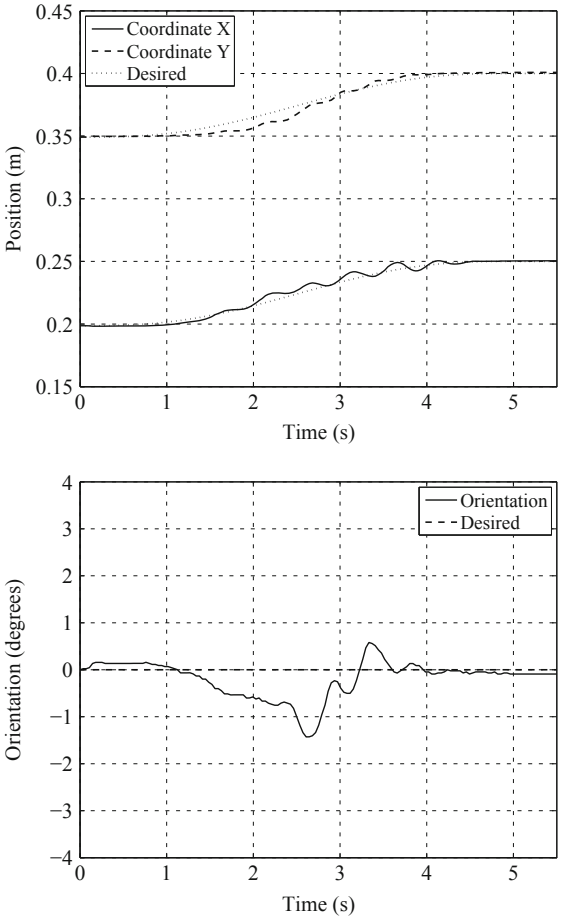


Fig. 9.4 Control of a rigid load by a system of two cooperative fully-actuated manipulators, quasi-LPV formulation

applied torque by the i-th joint for both manipulators, E_{τ}, and the integral of the squeeze force vector:

$$E_{h_{os}} = \sum_{i=1}^{nm} \left(\int_0^{t_r} |h_{oS_i}(t)| dt \right),$$

where t_r is the time it takes for the load to reach the desired position. The results are presented in Tables 9.2 and 9.3 and represent the average of five experiments.

Note that the nonlinear H_1 controller designed via game theory presents the lowest trajectory tracking error, L_2, although the energy usage, E_{τ}, and the squeeze forces, $E_{h_{os}}$, are higher in comparison to the quasi-LPV formulation.

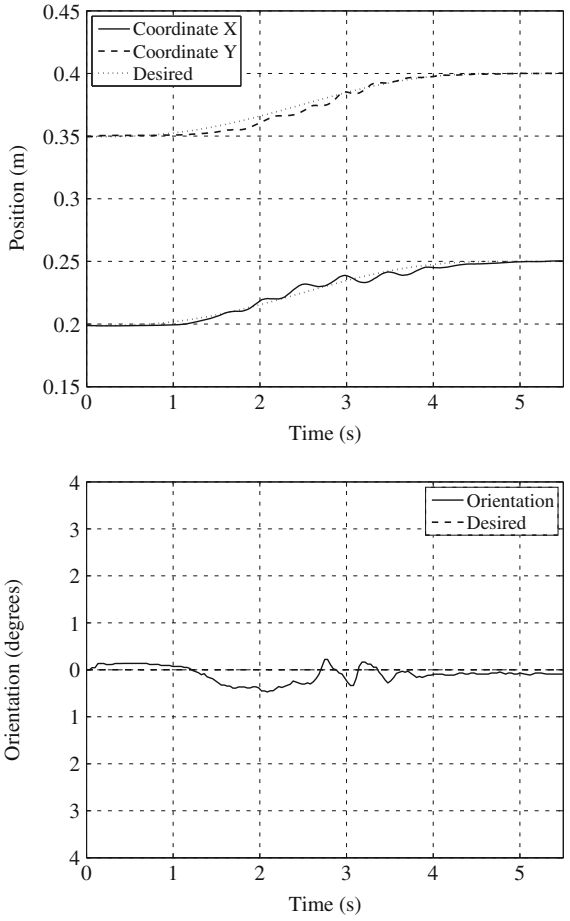


Fig. 9.5 Control of a rigid load by a system of two cooperative fully-actuated manipulators, game theory formulation

Table 9.2 Performance indexes, fully actuated con guration

Control Formulation	$L_2(\tilde{x})$	$E(\tau)$ (N m s)	$E(h_{os})$ (N s)
Quasi-LPV	0.01815	0.8318	0.2193
Game theory	0.01158	1.1200	0.3875

Table 9.3 Performance indexes, underactuated con guration

Nonlinear H_1	$L_2(\tilde{x})$	$E(\tau)$ (N m s)	$E(h_{os})$ (N s)
Quasi-LPV	0.0154	0.9976	0.4477
Game theory	0.0103	1.0609	0.3973

9.7.3 Underactuated Configuration

In this section, we assume that joint 1 of manipulator A in Fig. 9.3 is passive. In this case, $n(e = n(m - 1) - n_p = 2)$ and therefore only two components of the squeeze force can be controlled independently. We choose to control the X and Y components of the squeeze force but not the component relative to the momentum applied to the load. The desired values for the squeeze force are $\Gamma_{sc}^d = [0\ 0]^T$.

The parameters $\rho(\tilde{x})$, the variation rate bounds, and the basis functions needed to compute $X(\rho)$ are the same ones used in the fully-actuated case. The quasi-LPV system matrices are also the same except that $\widehat{M}(x_o) = \widetilde{M}(x_o)$ and $\widehat{C}(x_o, \dot{x}_o) = \widetilde{C}(x_o, \dot{x}_o)$. The parameter space was divided considering three points in the set P. The best level of attenuation is $\gamma = 1.25$. The weighting matrices for the nonlinear H_1 control via game theory are the same defined for the fully-actuated case. The level of attenuation adopted is $\gamma = 4.0$.

The experimental results are shown in Figs. 9.6 and 9.7, and the performance indexes in Table 9.3. Note that, in this case, the nonlinear H_1 controller via game theory presents the lowest trajectory tracking error and squeeze force. The best value for the energy usage is given by the nonlinear H_1 controller via quasi-LPV representation.

Figure 9.8 shows, for the quasi-LPV formulation, the squeeze force components when the squeeze force control is applied (continuous line) and when it is not applied (dashed line). It can be observed that only the components of the squeeze force relative to the linear coordinates are controlled and close to the desired values $\Gamma_{sc}^d = 0$. The component of the squeeze force relative to the momentum is not controlled in both cases, as mentioned before.

For the case where the squeeze force is not controlled, the values of $L_2(\tilde{x})$ and $E(\tau)$ are close to the values in Table 9.3. The values of $E(h_{os})$, however, are on average three times larger than when the squeeze force is controlled.

We ran the same experiment using the hybrid position/force control for underactuated manipulators proposed in [7]. The average performance indexes over five experiments are $L_2(\tilde{x}) = 0.0128$, $E(\tau) = 1.7781$, and $E(h_{os}) = 0.5741$. Although in this case $L_2(\tilde{x})$ is lower than that obtained via the quasi-LPV formulation, the values of $E(\tau)$ and $E(h_{os})$ are approximately 70% and 40% larger, respectively. The conclusion is that, in this case, the robust controllers present practically the same position tracking performance than the hybrid position/force controller, but with less control effort.

9.8 Neural Network-Based Adaptive Controller

We also experimented with controlling the underactuated cooperative manipulator with the nonlinear H_1 neural network-based adaptive controller developed in Sect. 9.6. The load parameters, presented in Table 9.4, represent those of a

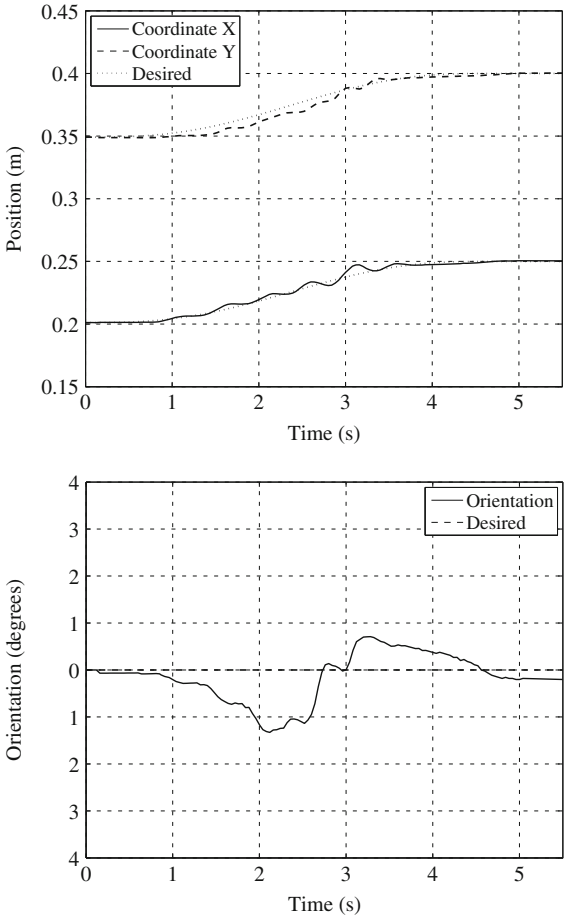


Fig. 9.6 Control of a rigid load by a system of two cooperative underactuated manipulators, quasi-LPV formulation

force-torque sensor attached to the manipulators' end-effectors. Figure 9.6 shows the workspace of the cooperative manipulator and the desired trajectory for the set of experiments presented next.

The desired trajectory is an arc of circle centered at $C = (0.24, 0.08)$ m and with radius $R = 0.26$ m. The arc spans from $x_o(0) = (-0.1\ \text{m}\ 0.3\ \text{m}\ 0)^T$ to $x_o^d(T) = (-0.38\ \text{m}\ 0.3\ \text{m}\ 0)^T$, where $T = 3.0$ s is the desired duration of the motion. The reference trajectory for the x-axis is a fifth-degree polynomial, and for the y-axis it is defined by the reference arc. The following external disturbances are introduced to verify the robustness of the proposed controllers:

$$\tau_{d_1} = \begin{bmatrix} 0.01e^{-\frac{(r-1.5)^2}{0.5}}\sin(2\pi t) \\ -0.01e^{-\frac{(r-1.5)^2}{0.5}}\sin(2.5\pi t) \\ -0.01e^{-\frac{(r-1.5)^2}{0.5}}\sin(3\pi t) \end{bmatrix}, \quad \tau_{d_2} = \begin{bmatrix} 0.02e^{-\frac{(r-1.5)^2}{0.5}}\sin(2\pi t) \\ 0.02e^{-\frac{(r-1.5)^2}{0.5}}\sin(2.5\pi t) \\ 0.01e^{-\frac{(r-1.5)^2}{0.5}}\sin(3\pi t) \end{bmatrix}.$$

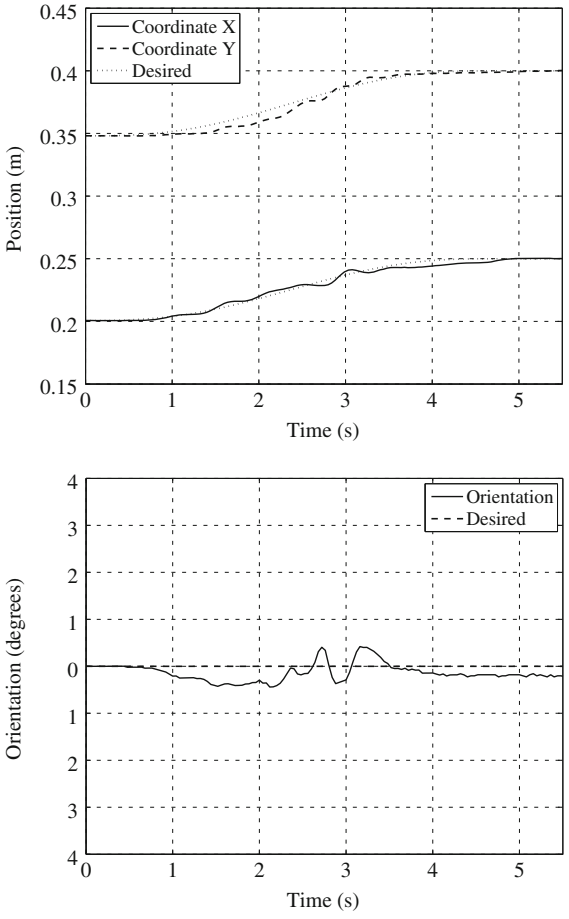


Fig. 9.7 Control of a rigid load by a system of two cooperative underactuated manipulators, game theory formulation

For the fully-actuated con guration, the following control parameters are used: $K = \mathrm{diag}\{3.42I_3, 0.38I_6\}$, $\Lambda = 0.8I_3$, $\Lambda_f = 0.5I_3$, $\rho = 0.4$, $\eta = 1$, and $S = 50$. The desired values for the squeeze force are $f_{0s}^d = [0\ 0\ 0]^T$. To compute the neural network, the following auxiliary variable is de ned:

$$\psi = \sum_{i=1}^{n} (x_o)_i + (\dot{x}_o)_i + (q_r)_i + (\dot{q}_r)_i.$$

The matrix Ξ can be computed as:

$$\Xi = diag\{\xi_1^T, \xi_2^T, \ldots, \xi_9^T\},$$

with $\xi_k = [\xi_{k1}, \ldots, \xi_{k7}]^T$, $\xi_k = (e^{\psi + m_i^k} - e^{-\psi - m_i^k})/(e^{\psi + m_i^k} + e^{-\psi - m_i^k})$, and m_i^k assumes the values $-1.5, -1, -0.5, 0, 0.5, 1, 1.5$, for $i = 1, \ldots, 7$, respectively. Note

Fig. 9.8 Squeeze force control

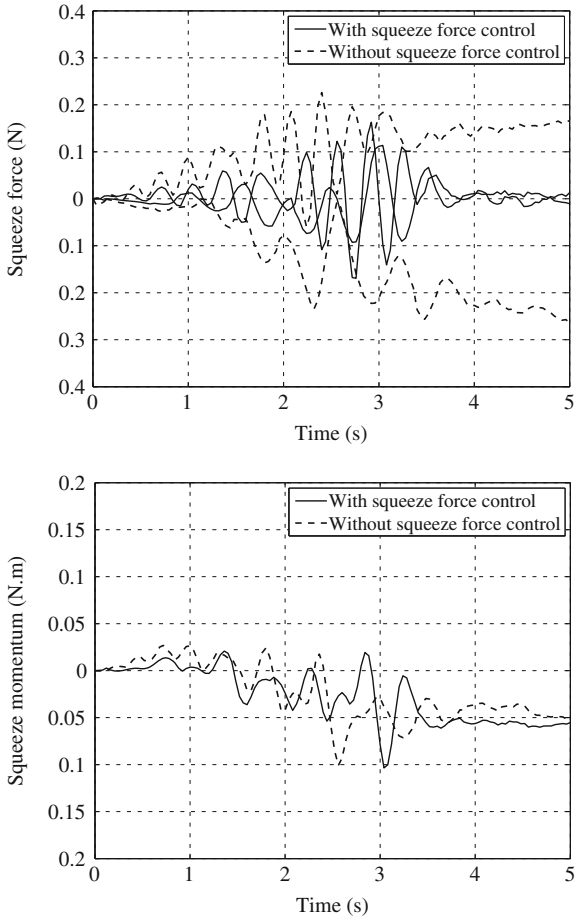


Table 9.4 Load parameters

Parameter	Value
Mass	$m_o = 1.45$ kg
Length	$l_o = 0.120$ m
Center of mass	$a_o = 0.060$ m
Inertia	$I_o = 0.0026$ kg m^2

that, with these definitions, seven neurons in the hidden layer are selected for the neural networks with the weights c_{ij}^k assuming the value 1. The network parameters Θ are defined as $\Theta = \left[\Theta_1^T, \ldots, \Theta_9^T\right]^T$, with $\Theta_k = [\Theta_{k1} \Theta_{k2} \ldots \Theta_{k7}]^T$. It is assumed that the approximation error is bounded by the state-dependent function $k(x_e)$ defined as $k(x_e) = 2\sqrt{\tilde{x}_o^2 + \dot{\tilde{x}}_o^2}$. The resulting Cartesian coordinates and orientation of the load are shown in Fig. 9.10. Table 9.5 compares the values of $\|\tilde{x}\|_2$, $E\{\tau\}$, and $E\{h_{oS}\}$ when all three controllers presented in this chapter are applied to the

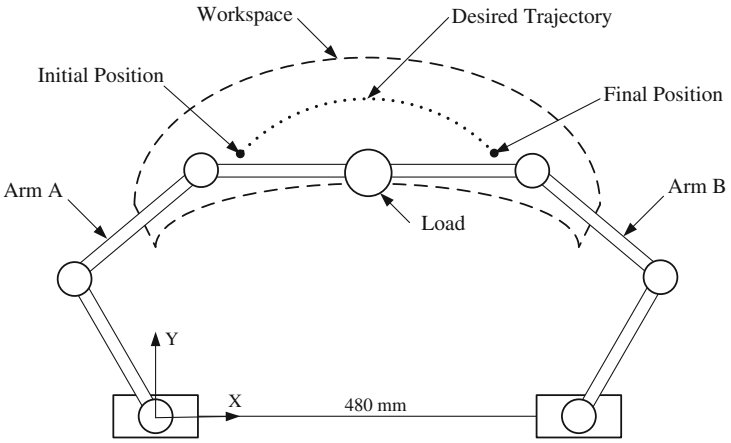
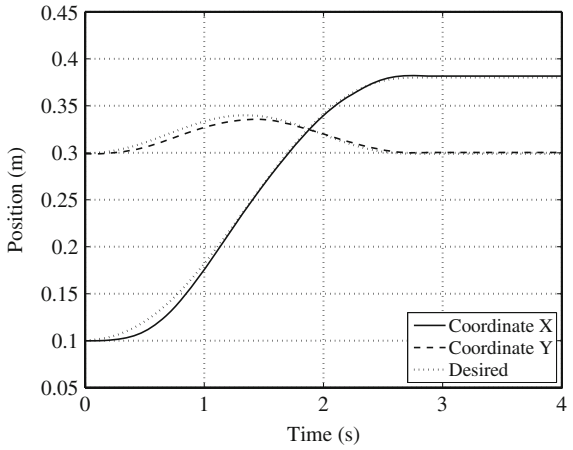


Fig. 9.9 Workspace of the cooperative manipulator system and desired trajectory

Fig. 9.10 Nonlinear H_1 neural network-based adaptive control of a rigid load by a system of two cooperative fully-actuated manipulators



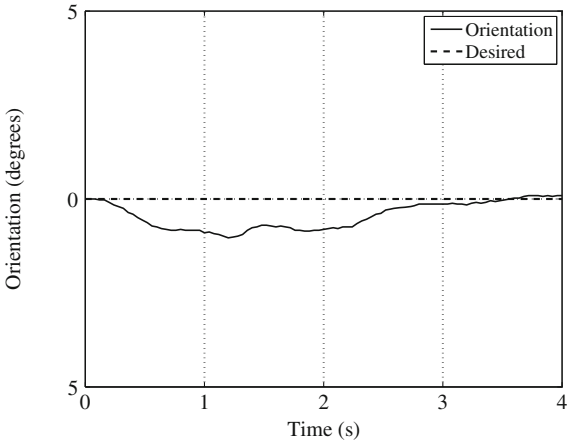


Table 9.5 Performance indexes, fully-actuated con guration

Controller formulation	$L_2[\tilde{x}]$	$E[\tau]$ (N m s)	$E[h_{os}]$ (N s)
Neural network-based	0.0267	1.09	0.9253
Quasi-LPV	0.0514	1.95	1.02
Game theory	0.0617	2.30	1.11

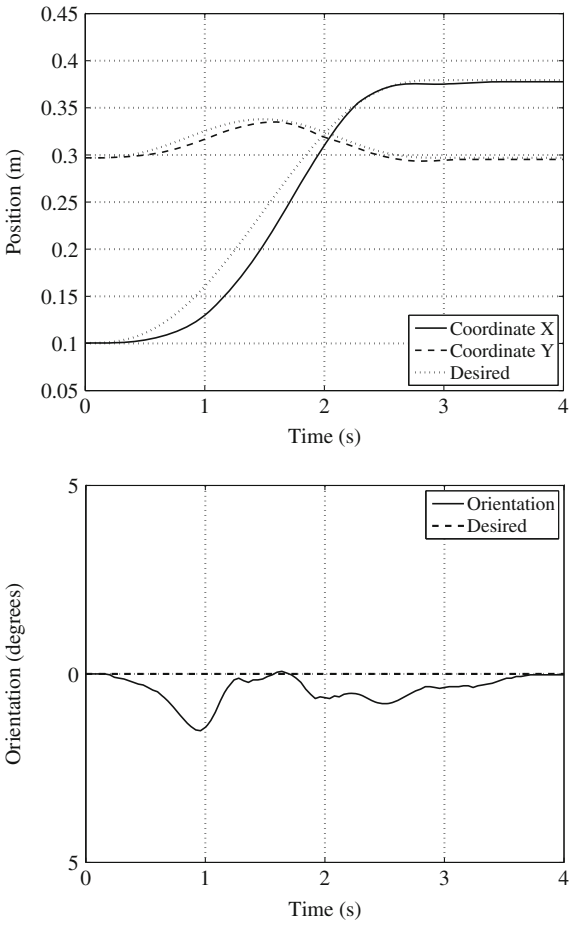


Fig. 9.11 Nonlinear H_1 neural network-based adaptive control of a rigid load by a system of two cooperative underactuated manipulators

fully-actuated cooperative manipulator system following the desired trajectory of Fig. 9.9. Note that the neural network-based H_1 controller presented signi cantly better performance both in terms of trajectory tracking error and energy usage, while its performance in terms of squeeze forces is equivalent to the other two.

When applying the neural network-based adaptive controller to the underactuated con guration, we again assumed that joint 1 of arm A is passive and the desired values for the squeeze forces are $f_{sc}^d = [0\ 0]^T$. The gains are $K = \mathrm{diag}[3.42 I_3, 0.38 I_6]$,

Table 9.6 Performance indexes, underactuated con guration

Controller formulation	L_2-norm	$E\{\tau\}$ (N m s)	$E\{h_{os}\}$ (N s)
Neural network-based	0.0462	2.36	1.56
Quasi-LPV	0.0561	2.70	1.61
Game theory	0.0574	2.83	1.65

$\Lambda = 1.5I_3$, $\Lambda_f = 0.4I_3$, $\rho = 0.2$, $\eta = 1$, and $S = 50$. The experimental results are shown in Fig. 9.11. The comparative performance indexes are presented in Table 9.6. Note that, once more, the neural network-based H_∞ controller presents the best performance in terms of trajectory tracking error and energy usage, while presenting similar performance in terms of squeeze force control.

References

1. Chang YC (2000) Neural network-based H_∞ tracking control for robotic systems. IEE Proc Control Theory Appl 147(3):303–311
2. Chen BS, Lee TS, Feng JH (1994) A nonlinear H_∞ control design in robotic systems under parameter perturbation and external disturbance. Int J Control 59(2):439–461
3. Khalil HK (1996) Adaptive output feedback control of nonlinear systems represented by input-output models. IEEE Trans Autom Control 41(2):177–188
4. Lian KY, Chiu CS, Liu P (2002) Semi-descentralized adaptive fuzzy control for cooperative multirobot systems with H_∞ motion/internal force tracking performance. IEEE Trans Syst, Man Cybern—Part B: Cybernetics 32(3):269–280
5. McClamroch NH, Wang D (1988) Feedback stabilization and tracking of constrained robots. IEEE Trans Autom Control 33(5):419–426
6. Siqueira AAG, Terra MH (2007) Neural network-based H_∞ control for fully actuated and underactuated cooperative manipulators. In: Proceedings of the American Control Conference, New York, USA pp 3259–3264
7. Tinós R, Terra MH (2006) Motion and force control of cooperative robotic manipulators with passive joints. IEEE Trans Control Syst Technol 14(4):725–734
8. Wen T, Kreutz-Delgado K (1992) Motion and force control for multiple robotic manipulators. Automatica 28(4):729–743
9. Wu F, Yang XH, Packard A, Becker G (1996) Induced L_2-norm control for LPV systems with bounded parameter variation rates. Int J Robust Nonlinear Control 6(9–10):983–998