Fault-tolerant robot manipulators based on output-feedback $H_\infty$ controllers

A.A.G. Siqueira$^a$,*, M.H. Terra$^b$, C. Buosi$^b$

$^a$ University of São Paulo at São Carlos, Mechanical Engineering Department, São Carlos, SP, 13566-590, Brazil
$^b$ University of São Paulo at São Carlos, Electrical Engineering Department, São Carlos, SP, 13566-590, Brazil

Received 22 February 2006; received in revised form 26 April 2007; accepted 28 April 2007
Available online 17 May 2007

Abstract

This paper develops two fault-tolerant control strategies for robot manipulators. The first is based on linear parameter-varying systems and the second on Markovian jump linear systems. Firstly, it is shown that with the LPV approach post-fault stability is guaranteed only if the robot stops completely after a fault detection. Then, with an underactuated configuration, the manipulator can be controlled appropriately. Secondly, it is shown that with the fault-tolerant system based on Markovian jump linear systems, stability is guaranteed after a fault is detected even with the robot still moving. This approach incorporates all manipulator configurations in a unified model. Both strategies have been implemented based on output-feedback $H_\infty$ controllers, which are the main focus of this paper. Experimental results illustrate the performance of each controller.

Keywords: Fault-tolerant system; $H_\infty$ control; Markovian jump linear system; Linear parameter-varying system; Robot manipulator

1. Introduction

Parametric uncertainties and exogenous disturbances increase the difficulty of reference tracking control for robot manipulators. $H_\infty$ control strategies for robot manipulators based on state-feedback control have been used to minimize the disturbance effects in system performance, [1]. However, the velocity signal, considered as state, generally is not available and can be obtained indirectly from a position measurement. This procedure can introduce noises and delays, which decrease tracking control efficiency. An output-feedback controller can be used in order to avoid these problems. Two design techniques for output-feedback gain-scheduling controllers with a guaranteed $H_\infty$ performance are proposed in [2] for linear parameter-varying (LPV) systems. In this paper, the second design technique, named as Projected Characterization, is applied to an actual robot manipulator in its Quasi-LPV representation, which means the parameters matrix of the model depends on the state.

Fault-tolerant systems for robot manipulators have been developed by several authors; see, for instance, [3–5] and references therein. Free torque failures in robot manipulators, where the torque supply in the motor breaks down suddenly, can make these systems uncontrollable. Furthermore, if the robot is working in hazardous or unstructured environments, where repairs are not allowed, the requested movement must be completed according to the fault configuration. When a free torque failure occurs, the fully actuated manipulator changes to an underactuated configuration. However, when the manipulator changes, after a fault occurrence, from a fully actuated to an underactuated configuration, the system stability is not guaranteed with the deterministic output-feedback controllers proposed in [2]. To use these controllers in a fault-tolerant robot system, it is necessary to stop completely the movement of all joints after the fault detection, restarting it from zero velocity.

To avoid the necessity of stopping the robot when a fault occurs, Markov theory is used in this paper to characterize abrupt changes in the operation points of the robotic manipulator. A model is developed based on linear systems subject to abrupt variations, namely, Markovian jump linear systems (MJLS) [6,7]. In order to formulate this model, the manipulator dynamic is linearized around operation points, and a Markovian model is developed to encompass the changes of the operation points and the transition rate between fault configurations [1,8]. With the proposed model, that represents
all manipulator configurations in a unified way, the output-feedback $\mathcal{H}_\infty$ controller for MJLS proposed in [9] is used to guarantee stability after the occurrence of a sequence of faults.

This paper is organized as follows: in Section 2, the Quasi-LPV representations of fully actuated and underactuated robot manipulators are presented, with experimental results, using a deterministic output-feedback $\mathcal{H}_\infty$ controller; in Section 3, the fault-tolerant manipulator model and the control system based on output-feedback $\mathcal{H}_\infty$ controller for MJLS are presented, and two fault sequences for the UArm II robot are evaluated to demonstrate the effectiveness of this approach.

2. Quasi-LPV representation of the manipulator

2.1. Fully actuated manipulator

The dynamic equations of a robot manipulator can be found by Lagrange theory as

$$\tau = M(q)\ddot{q} + C(q, \dot{q})\dot{q} + F + G(q),$$  \hspace{1cm} (1)

where $q \in \mathbb{R}^n$ is the joint position vector, $M(q) \in \mathbb{R}^{n \times n}$ is the symmetric positive definite inertia matrix, $C(q, \dot{q}) \in \mathbb{R}^{n \times n}$ is the Coriolis and centripetal matrix, $F \in \mathbb{R}^{n \times 1}$ is the diagonal matrix of frictional torque coefficients, $G(q) \in \mathbb{R}^n$ is the gravitational torque vector, and $\tau \in \mathbb{R}^n$ is the applied torque vector. A parametric uncertainty can be introduced dividing the matrix $M(q)$, $C(q, \dot{q})$, $F$, and $G(q)$ into a nominal and a perturbed part, where $M_0(q)$, $C_0(q, \dot{q})$, $F_0$, and $G_0(q)$ are the nominal matrices, and $\Delta M(q)$, $\Delta C(q, \dot{q})$, $\Delta F$, and $\Delta G(q)$ are the parametric uncertainties. A finite energy exogenous disturbance, $\tau_d \in \mathbb{R}^n$, can also be introduced resulting in

$$\tau + \delta(q, \dot{q}, \ddot{q}, \tau_d) = M_0(q)\ddot{q} + C_0(q, \dot{q})\dot{q} + F_0\dot{q} + G_0(q),$$  \hspace{1cm} (2)

with

$$\delta(q, \dot{q}, \ddot{q}, \tau_d) = -(\Delta M(q)\ddot{q} + \Delta C(q, \dot{q})\dot{q}$$
$$+ \Delta F\dot{q} + \Delta G(q) - \tau_d).$$

The state is defined as $x = [\dot{q}^T \dot{q}^T]^T$, where $q$ and $\dot{q}$ are the positions and the velocities of the manipulator joints, respectively. The Quasi-LPV representation of a fully actuated manipulator is given by

$$\dot{x} = A(q, \dot{q})x + B(q)u + B(q)\delta(q, \dot{q}, \ddot{q}, \tau_d),$$  \hspace{1cm} (3)

with

$$A(q, \dot{q}) = \begin{bmatrix} -M_0^{-1}(q) (C_0(q, \dot{q}) + F_0) & 0 \\ I_{n \times n} \\ 0 \\ 0 \end{bmatrix},$$
$$B(q) = \begin{bmatrix} M_0^{-1}(q) \\ 0 \\ 0 \end{bmatrix},$$
$$u = \tau - G_0(q).$$

2.2. Underactuated manipulator

Underactuated robot manipulators are mechanical systems with fewer actuators than degrees of freedom. For this reason, the control of passive joints is made considering the dynamic coupling between them and the active joints. Here, the manipulator is considered with $n$ joints, in which $n_p$ are passive and $n_a$ are active joints. From [10], no more than $n_a$ joints of the manipulator can be controlled at every instant when breaks are used in the passive joints. Let $n_a$ be the number of passive joints that have not already reached their set point in a given instant. If $n_a \geq n_a$, $n_a$ passive joints are controlled and grouped in the vector $q_a \in \mathbb{R}^{n_a}$, the remaining passive joints, if any, are kept locked, and the active joints are grouped in the vector $q_a \in \mathbb{R}^{n_a}$. If $n_a < n_a$, the $n_a$ passive joints are controlled applying torques in $n_a$ active joints. In this case, $q_a \in \mathbb{R}^{n_a}$ and $q_a \in \mathbb{R}^{n_a}$. The strategy is to control all passive joints until they reach the desired position, considering the conditions exposed above, and then turn on the brakes. After that, all the active joints are controlled by themselves as a fully actuated manipulator. The dynamic Eq. (2) can be partitioned as

$$\begin{bmatrix} \tau_a \\ 0 \end{bmatrix} + \begin{bmatrix} \delta_a \\ 0 \end{bmatrix} = \begin{bmatrix} M_{aa} & M_{au} \\ M_{au} & M_{uu} \end{bmatrix} \begin{bmatrix} \dot{q}_a \\ \dot{q}_u \end{bmatrix} + \begin{bmatrix} C_{aa} & C_{au} \\ C_{au} & C_{uu} \end{bmatrix} \begin{bmatrix} q_a \\ q_u \end{bmatrix} + \begin{bmatrix} F_{aa} & 0 \\ 0 & F_{au} \end{bmatrix} \begin{bmatrix} \dot{q}_a \\ \dot{q}_u \end{bmatrix} + \begin{bmatrix} G_a \\ G_u \end{bmatrix},$$  \hspace{1cm} (4)

where the indices $a$ and $u$ represent the active and free (breaks not actuated) passive joints, respectively. Factoring out the vector $\tilde{q}_a$ in the second line of (4) and substituting in the first one, results in

$$\begin{bmatrix} \tau_a + \delta(q, \dot{q}, \ddot{q}, \tau_d) \end{bmatrix} = \begin{bmatrix} \tilde{M}_0(q) \tilde{q}_a + \tilde{C}_0(q, \dot{q}) \tilde{q}_a + \tilde{F}_0(q) \tilde{q}_u \\ \tilde{D}_0(q, \dot{q}) \tilde{q}_a + \tilde{G}_0(q) \end{bmatrix},$$  \hspace{1cm} (5)

where $\tilde{M}_0(q) = M_{aa} - M_{aa} M^{-1}_{aa} M_{au}$, $\tilde{C}_0(q, \dot{q}) = C_{aa} - C_{aa} M^{-1}_{aa} C_{au} + F_{aa}$, $\tilde{F}_0(q) = -M_{aa} M^{-1}_{aa} F_{au}$, $\tilde{G}_0(q) = G_a - M_{aa} M^{-1}_{aa} G_u$, $\delta(q, \dot{q}, \ddot{q}, \tau_d) = \delta_a - M_{aa} M^{-1}_{aa} \delta_a$, with matrices and vectors have appropriate dimensions, depending on the numbers of active, $n_a$, and free passive joints, $n_u$. The state is defined as $x_a = [\tilde{q}_a^T \tilde{q}_a^T]^T$. Hence, a Quasi-LPV representation of the underactuated manipulator can be defined as follows

$$\dot{x}_a = A(q, \dot{q})x_a + B(q)u + B(q)\delta(q, \dot{q}, \ddot{q}, \tau_d),$$  \hspace{1cm} (6)

with

$$A(q, \dot{q}) = \begin{bmatrix} -\tilde{M}_0^{-1}(q) (\tilde{C}_0(q, \dot{q}) + \tilde{F}_0(q)) & 0 \\ I \\ 0 \end{bmatrix},$$
$$B(q) = \begin{bmatrix} \tilde{M}_0^{-1}(q) \end{bmatrix},$$
$$u = \tau_a - \tilde{D}_0(q, \dot{q}) \tilde{q}_a - \tilde{G}_0(q).$$
2.3. Output-feedback \( H_\infty \) LPV control

To apply the control techniques presented in [2], the robot manipulator needs to be represented according to equation
\[
\begin{align*}
\dot{x} &= A(\theta)x + B_1(\theta)w + B_2(\theta)u, \\
z &= C_1(\theta)x + D_{11}(\theta)w + D_{12}(\theta)u, \\
y &= C_2(\theta)x + D_{21}(\theta)w, \\
\end{align*}
\]
where \( \theta = [\rho_1(t), \ldots, \rho_N(t)]^T \) belongs to a convex space \( \mathcal{P} \), and \( \rho_i(t), i = 1, \ldots, N \), are the time-varying parameters satisfying \( |\dot{\rho}_i(t)| \leq v_i \) with \( v_i \geq 0, i = 1, \ldots, N \), the bounds of the parameter variation rates. Consider as system disturbances, the desired position, \( q^d \), and the combined torque disturbance, \( \delta \), that is: \( w = [\delta^T(q^d)^T]^T \). The system outputs, \( z \), are the position error, \( q^d - q \), and the control input, \( u \). The control output is the position error, \( y = q^d - q \), since we only have the position measured directly. Note that for the underactuated case, instead of using \( \dot{q} \) as the state, one must use \( \dot{\hat{q}}_u, \dot{q}_u \).

Hence, the robot system can be described by (7) with
\[
\begin{align*}
A(\theta) &= A(q, \dot{q}), \\
B_1(\theta) &= [B(q) \ 0], \\
B_2(\theta) &= B(q), \\
C_1(\theta) &= \begin{bmatrix} 0 & -I \\ 0 & 0 \end{bmatrix}, \\
C_2(\theta) &= [0 \ -I]^T, \\
D_{11}(\theta) &= \begin{bmatrix} 0 & 0 \end{bmatrix}, \\
D_{12}(\theta) &= \begin{bmatrix} 0 & I \end{bmatrix}, \\
D_{21}(\theta) &= \begin{bmatrix} 0 & 1 \end{bmatrix}, \\
D_{22}(\theta) &= 0,
\end{align*}
\]
where matrices \( A(q, \dot{q}) \) and \( B(q) \) are obtained from (3), for the fully actuated case, and (6), for the underactuated case. In [2], two \( H_\infty \) control techniques for LPV systems are presented. The one named as Projected Characterization, which uses the projection lemma [11] to reduce the number of unknown variables, was applied to the manipulator in its Quasi-LPV representation (7). The controller dynamics is defined as
\[
\begin{align*}
\dot{x}_K &= \begin{bmatrix} A_K(\theta, \dot{\hat{q}}) & B_K(\theta, \dot{\hat{q}}) & D_K(\theta, \dot{\hat{q}}) \end{bmatrix} x_K, \\
y &= \begin{bmatrix} 0 \\ I \\ 0 \end{bmatrix} X.
\end{align*}
\]
To obtain the controller, one must solve the following set of linear matrix inequalities (LMI) for \( X(\theta) \) and \( Y(\theta) \) minimizing \( \gamma \).

\[
\begin{align*}
\begin{bmatrix} N_X \ 0 \ 0 \end{bmatrix}^T \begin{bmatrix} X + XA + A^T X & XB_1 \\
B_1^T X & -\gamma I & D_{11}^T \\
B_1 & D_{11} & -\gamma I \end{bmatrix} \begin{bmatrix} C_1^T \\
D_{12}^T \\
D_{21}^T \end{bmatrix} < 0,
\end{align*}
\]
and
\[
\begin{align*}
\begin{bmatrix} N_Y \ 0 \ 0 \end{bmatrix}^T \begin{bmatrix} -\dot{Y} + YAT + AY & YC_1^T \\
C_1 Y & -\gamma I & D_{11}^T \\
B_1^T & D_{11} & -\gamma I \end{bmatrix} \begin{bmatrix} B_1 \\
D_{12}^T \\
D_{21}^T \end{bmatrix} < 0,
\end{align*}
\]
where \( N_X \) and \( N_Y \) designate any bases of the null spaces of \([C_2 \ D_{21}]\) and \([B_2^T \ D_{12}]\), respectively. Note that the matrices depend on \( \theta \); this dependency was omitted for convenience. After finding \( X \) and \( Y \), the LPV controller can be designed by the following sequential scheme
- Compute \( D_K \), solution to
\[
\begin{align*}
\sigma_{\max}(D_{11} + D_{12} D_K D_{21}) < \gamma,
\end{align*}
\]
and set \( D_{12} := D_{11} + D_{12} D_K D_{21} \).
- Compute \( \hat{B}_K \) and \( \hat{C}_K \), solutions to the linear matrix equations
\[
\begin{align*}
\begin{bmatrix} 0 & D_{21} & 0 \\
D_{12} & -\gamma I & D_{11} \\
0 & D_{11} & -\gamma I \end{bmatrix} \begin{bmatrix} \hat{B}_K \\
\hat{C}_K \end{bmatrix} &= -\begin{bmatrix} C_2 \\
D_{12} \\
D_{21} \end{bmatrix} \\
\begin{bmatrix} C_1 + D_{12} D_K C_2 \\
C_1 Y + D_{12} \hat{C}_K \\
(\hat{B}_K + B_2 D_K D_{21})^T \end{bmatrix},
\end{align*}
\]
- Compute \( \hat{A}_K = -(A + B_2 D_K C_2)^T + [XB_1 + \hat{B}_K D_{21}(C_1 + D_{12} D_K C_2)] + D_{12} D_K C_2)^T [D_{11} - \gamma I]^{-1} \times [C_1 Y + D_{12} \hat{C}_K].
\]
- Solve for \( N, M \), the factorization problem
\[
I - XY = NM^T,
\]
- Finally, compute \( A_K, B_K, \) and \( C_K \) as
\[
A_K = N^{-1}(XY + \hat{A}_K - X(A + B_2 D_K C_2)Y - \hat{B}_K C_2 Y - X B_2 \hat{C}_K) M^{-T},
\]
\[
B_K = N^{-1}(\hat{B}_K - X B_2 D_K),
\]
\[
C_K = (\hat{C}_K - D_K C_2) M^{-T}.
\]

The LMI problem defined by (9)–(11) is infinite-dimensional, since the parameter vector \( \theta \) varies continuously. To solve this problem, one can divide the parameter space, \( \mathcal{P} \), in several points. The variables \( X(\theta) \) and \( Y(\theta) \) will be a solution if both satisfy the LMIs on all points. There is no systematic rule that defines how \( X(\theta) \) and \( Y(\theta) \) vary upon \( \theta \), although this problem is usually solved using basis functions to describe the unknown matrices, which lead them to be written as
\[
Z(\theta) = \sum_{i=1}^{M} f_i(\theta) Z_i,
\]
where \( \{f_i(\theta)\}_{i=1}^{M} \) are differentiable functions of \( \theta \). See [2] for more details on the controller synthesis.
Remark 1. Note that to obtain the best performance of this controller, the choice of $v_2$ (in order to guarantee $|\dot{\theta}(t)| \leq v_2$) should be checked a posteriori.

2.4. Experimental results

The output-feedback $H_\infty$ LPV controller described in the previous section was applied to the experimental underactuated manipulator UArm II (Underactuated Arm II), Fig. 1, which is a special-purpose planar manipulator whose joints can be configured to act as passive (P) or active (A). More information on the manipulator can be found in [12].

2.4.1. Fully actuated control

For the fully actuated control design, the selected parameters, which are part of the state vector, are $\theta(x) = [q_1, q_3]^T$. The parameter space, $\mathcal{P}$, is defined as $\theta \in [40 \ -40]^\circ \times [40 \ -40]^\circ$. The parameter variation rate is bounded by $|\dot{\theta}| \leq [90 90]^\circ/s$. $X(\theta)$ and $Y(\theta)$ were defined as follows

$$X(\theta) := X_0 f_1(\theta),$$

$$Y(\theta) := Y_0 f_1(\theta) + Y_1 f_2(\theta) + Y_2 f_3(\theta),$$

where $f_1(\theta) = 1$, $f_2(\theta) = \sin(q_2) + \cos(q_2)$, $f_3(\theta) = \sin(q_3) + \cos(q_3)$. The parameter space was divided in $L = 5$ for each parameter, and the best level of attenuation found was $\gamma = 2.49$.

2.4.2. Underactuated control

The underactuated configuration used in the experiment presented in the next section is the APA configuration, i.e. joint 2 is passive and joints 1 and 3 are active. For this configuration, two control phases, hence two controllers, are necessary to control all joints to the set point. In the first control phase, the passive joint 2 is controlled by the dynamic coupling with the active joint 1, that is, $q_2 = q_1$, and joint 3 is kept locked; in the second one, the active joints 1 and 3 are controlled. For the first control phase of the experiment, the parameter $\theta$ selected is the state representing the position of joint 2, $\theta(x) = [q_2]$. The parameter space, $\mathcal{P}$, is defined as $\theta \in [30 \ -30]^\circ$. The parameter variation rate is bounded by $|\dot{\theta}| \leq [90 90]^\circ/s$. $X(\theta)$ and $Y(\theta)$ are defined as in (16) with, $f_1(\theta) = 1$, $f_2(\theta) = \sin(q_2) + \cos(q_2)$. The parameter space was divided in $L = 5$, and the best level of attenuation found was $\gamma = 1.87$. For the second control phase, the selected parameters, that are part of the state vector, were $\theta(x) = [q_1, q_3]^T$. The parameter space, $\mathcal{P}$, is defined as $\theta \in [40 \ -40]^\circ \times [40 \ -40]^\circ$. The parameter variation rate is bounded by $|\dot{\theta}| \leq [90 90]^\circ/s$. $X(\theta)$ and $Y(\theta)$ are the same as in (16) with $f_1(\theta) = 1$, $f_2(\theta) = \sin(q_1) + \cos(q_1)$, $f_3(\theta) = \sin(q_3) + \cos(q_3)$. The parameter space was divided in $L = 5$ for each parameter, and the best level of attenuation found was $\gamma = 2.36$.

2.4.3. AAA–APA fault sequence

The controllers designed in Sections 2.4.1 and 2.4.2 do not guarantee that the joints will reach the set point if a free torque fault occur in the second joint, changing suddenly the configuration from AAA to APA. To verify this behavior, one experiment was performed considering initially the manipulator in the fully actuated configuration AAA, with initial position $q(0) = [0^\circ \ 0^\circ \ 0^\circ]^T$ and desired final position $q(T) = [20^\circ \ 20^\circ \ 20^\circ]^T$.

When the joint positions reached approximately $15^\circ$ for all joints, at $t_f = 2.5$ s, a free torque fault was introduced in the second joint. Hence, the controller changes from the fully actuated configuration to the underactuated one maintaining the manipulator movement.

Here, we assume that a fault detection system indicates the fault instantaneously. As can be seen in Fig. 2, the LPV controllers were not able to react efficiently to the fault occurrence, resulting in an unstable system.

An alternative procedure is to use brakes during the control reconfiguration, which is equivalent to using each controller considering their design specifications. For the UArm II, when the brakes are turned on, with the joints still in movement, there are some oscillations in the joint positions which take at least 1 s to vanish; see Fig. 3. In this case, all joints are locked for $t_f = 1$ s between the fault detection and the beginning of the APA configuration control phase. One disadvantage of this procedure is that some components can be damaged, mainly when the robot is performing high-speed motions. The results of this experiment can be seen in Fig. 3. The next objective is to design a control strategy that eliminates the necessity of stopping the
joints between the fully actuated and underactuated control phases.

### 3. Manipulator robots as Markovian model

In this section, MJLS are adopted to model free-joint failures of robot manipulators. The dynamic model of an underactuated manipulator (5) can be represented as

\[
\tau_a = M_0(q)\ddot{q} + \bar{b}_0(q, \dot{q}) + \bar{\delta}(q, \dot{q}, \ddot{q}),
\]

with \( \bar{b}_0(q, \dot{q}) = C_0(q, \dot{q})\dot{q} + F_0(q)\dot{q} + D_0(q, \dot{q})\dot{q} + G_0(q) \). The fully actuated manipulator (2) can be represented by (17), with \( q_a = q, M_0(q) = M_0(q), \bar{b}_0(q, \dot{q}) = b_0(q, \dot{q}) = C_0(q, \dot{q})\dot{q} + F_0(q) + G_0(q), \) and \( \bar{\delta}(q, \dot{q}, \ddot{q}) = \delta(q, \dot{q}, \ddot{q}) \). A proportional controller can be introduced in the form \( \tau_a = [K_P \ 0]x + u \), in order to pre-compensate model imprecisions. The linearization of (17) around an operation point with position \( \dot{q}_0 \) and velocity \( \ddot{q}_0 \), is given by

\[
\begin{align*}
\dot{x} &= Ax + Ew + Bu, \\
z &= C_1x + D_1u, \\
y &= C_2x + D_2w,
\end{align*}
\]

with the equations given in Box I, where \( q^d \) is the desired trajectory, \( \alpha \) and \( \beta \) are constants defined by the designer, and used to adjust the controllers, and \( w = \ddot{\delta} \).

The model presented in this section describes the changes between the linearization points of the plant (18), and the probability of a fault occurrence for the three-link manipulator.

The workspace of each joint is divided into two sectors of 10° each. For each sector a linearization point is defined, 5° for the first sector and 15° for the second one. All the possible combinations to position the three joints, \( q_1, q_2, q_3 \), in these two points are used to map the manipulator workspace. Then, eight linearization points, with the velocities set to zero, are found. For a three-link manipulator robot, seven possible fault configurations can occur: AAP, APA, PAA, APP, PAP, and PPP, where A represents active joints and P represents passive joints. Here, it is considered that two or more failures cannot occur simultaneously. The fault configurations AAP, APA, and PAA have \( n_a = 2 \), then two control phases are necessary to control all joints to the set point. The first control phase is denoted by the configuration name followed by the subscript \( u_1 \); the second control phase is followed by the subscript \( u_2 \); and the third control phase is followed by the subscript \( l \). The Markovian states are the manipulator dynamic model linearized properly according to (18) in the eight points for all control phases of all configurations. Fig. 4.

**Remark 2.** Here, the transition rate matrix \( \Lambda \) is used instead of the well-known transition probability matrix \( P \) adopted in the fault-tolerant model based on state-feedback control for MJLS presented in [1,8]. The difference between them is that, while the sum of the row elements of \( P \) is one, the sum of the row elements of \( \Lambda \) is zero. \( \Lambda \) describes the transition rate among the Markovian states, which is positive when the system jumps to a different state and negative when it remains in the same state; see [13] for more details.

The transition rate matrices \( \Lambda_f, \Lambda_s, A_0, \) and \( A_{100} \) describe, respectively, the rates of a fault occurrence, the passive joint being controlled to reach the set point, the defective joint to be repaired, and the manipulator to stay in the configuration PPP.

**Remark 3.** The generalization of the Markovian model developed in this section for free-joint failures can be performed for \( n \)-link robot manipulators. The total number of
Markovian states can be computed by the following formula

\[ T_{MS} = 1 + n_l \times \left( \sum_{i=1}^{n-1} (n_{cp_i} \times n_{f_c_i}) + 1 \right), \]

where \( n_{f_c_i} = \left\lceil \frac{n}{(n-1)!} \right\rceil \) is the number of possible fault configurations for \( i \) faults, \( n_{cp_i} = \text{ceil} \left( \frac{n}{(n-r)} \right) \) is the number of control phases for a configuration with \( i \) faults (ceil(x) rounds \( x \) to the nearest integer towards infinity) and \( n_l = (n_r)^n \) is the number of linearization points (\( n_r \) is the number of sectors where the linearizations will be performed).

### 3.1. Output-feedback \( H_\infty \) control for MJLS

The output-feedback \( H_\infty \) control for MJLS briefly presented in this section was originally presented in [9]. Consider the collections of real matrices, \( A = (A_1, \ldots, A_N), \dim(A_1) = n \times n, E = (E_1, \ldots, E_N), \dim(E_1) = n \times m, B = (B_1, \ldots, B_N), \dim(B_1) = n \times r, C_1 = (C_{11}, \ldots, C_{1N}), \dim(C_{1i}) = p \times n, D_1 = (D_{11}, \ldots, D_{1N}), \dim(D_{1i}) = p \times r, C_2 = (C_{21}, \ldots, C_{2N}), \dim(C_{2i}) = q \times n, \) and \( D_2 = (D_{21}, \ldots, D_{2N}), \dim(D_{2i}) = q \times m, i = 1, \ldots, N. \)

Let us consider a continuous-time homogeneous Markov chain, \( \theta = \{ \theta(t) : t > 0 \} \), with transition probability \( Pr(\theta(t+\Delta t) = j \mid \theta(t) = i) \) defined as

\[
Pr(\theta(t+\Delta t) = j \mid \theta(t) = i) = \begin{cases} 
\lambda_{ij}(t)\Delta + o(\Delta) & \text{if } i \neq j \\
1 + \lambda_{ii}(t)\Delta + o(\Delta) & \text{if } i = j,
\end{cases}
\]

where \( \Delta > 0, \) and \( \lambda_{ij}(t) \geq 0 \) is the transition rate of the Markovian state \( i \) to \( j \) (\( i \neq j \)), and

\[ \lambda_{ii}(t) = - \sum_{j=1, j \neq i}^{N} \lambda_{ij}(t). \]

The probability distribution of the Markov chain at the initial time is given by \( \mu = (\mu_1, \ldots, \mu_N) \) in such a way that \( Pr(\theta(0) = i) = \mu_i. \)

The dynamic controller is given by

\[
\dot{x}(t) = A_{q(i)}x(t) + E_{q(i)}w(t) + B_{q(i)}u(t),
\]

\[
z(t) = C_{1q(i)}x(t) + D_{1q(i)}u(t), \quad y(t) = C_{2q(i)}x(t) + D_{2q(i)}u(t), \quad t \geq 0,
\]

with \( w \in \mathcal{L}_2(0,T), E(|x_0|^2) < \infty, \theta(0) \sim \mu, \) where \( x = [x(t), t \geq 0], z = [z(t), t \geq 0], e = [y(t), t \geq 0], e \) and \( y \) are respectively, the state, the controlled output, and the measured output of (20). Thus, whenever \( \theta(t) = i \in S, \) one has \( A_{q(i)} = A_i, E_{q(i)} = E_i, B_{q(i)} = B_i, C_{1q(i)} = C_{1i}, D_{1q(i)} = D_{1i}, C_{2q(i)} = C_{2i}, \) and \( D_{2q(i)} = D_{2i}. \)

The dynamic controller is given by

\[
\dot{v}(t) = A_{c(i)}v(t) + B_{c(i)}y(t),
\]

\[
u(t) = C_{c(i)}v(t), \quad t \geq 0,
\]

where \( A_c = (A_{c1}, \ldots, A_{cN}), \dim(A_{c1}) = n \times n, B_c = (B_{c1}, \ldots, B_{cN}), \dim(B_{c1}) = n \times q, \) and \( C_c = (C_{c1}, \ldots, C_{cN}), \dim(C_{c1}) = p \times n. \) The output-feedback \( H_\infty \) problem for MJLS is to find a controller \( (A_c, B_c, C_c) \) such that the \( H_\infty \) norm of the
closed-loop system is smaller than $\gamma$. To find this controller the LMI system given in Box II must be solved with

$$
\begin{bmatrix}
A_i^T X_i + X_i A_i + L_i C_{2i} + C_{2i}^T L_i + C_{1i}^T C_{1i} + \sum_{j=1}^{N} \lambda_{ij} X_j & X_i E_i + L_i D_{2i} \\
E_i^T X_i + D_{2i}^T L_i & -\gamma^{-2} I
\end{bmatrix} < 0,
$$

$$
\begin{bmatrix}
A_i Y_i + Y_i A_i^T + B_i F_i + F_i^T B_i^T + \lambda_{ii} Y_i + \gamma^{-2} E_i E_i^T & Y_i C_{1i}^T + F_i^T D_{1i}^T & R_i(Y) \\
C_{1i} Y_i + D_{1i} F_i & -I & 0 \\
R_i^T(Y) & 0 & S_i(Y)
\end{bmatrix} < 0,
$$

where

$$
Y_i = \frac{1}{2} \begin{bmatrix}
1 & I \\
I & X_i
\end{bmatrix} > 0.
$$

Box II.

Table 1

| AAA–APA Markovian states and linearization points |
|-----------------|-----------------|-----------------|
| AAA  | APAa  | APAb  | APAc  | APAa  | APAb  | APAc  | APAa  | APAb  | APAc  |
| 1    | 19    | 5     | 5     | 5     | 0     | 0     | 0     | 0     | 0     |
| 2    | 10    | 18    | 15    | 5     | 5     | 0     | 0     | 0     | 0     |
| 3    | 11    | 19    | 5     | 5     | 15    | 0     | 0     | 0     | 0     |
| 4    | 12    | 20    | 15    | 5     | 5     | 0     | 0     | 0     | 0     |
| 5    | 13    | 21    | 5     | 5     | 15    | 0     | 0     | 0     | 0     |
| 6    | 14    | 22    | 15    | 5     | 5     | 0     | 0     | 0     | 0     |
| 7    | 15    | 23    | 5     | 15    | 15    | 0     | 0     | 0     | 0     |
| 8    | 16    | 24    | 15    | 15    | 15    | 0     | 0     | 0     | 0     |

3.2. Experimental results

3.2.1. AAA–APA fault sequence

For the experimental implementation, it is considered the fault sequence where a free-joint fault occurs in joint 2, named AAA–APA fault sequence, represented in the fault-tolerant model by the numbers 1, 2, and 3; see Fig. 4. The vector of controlled joints, $q_c$, is chosen as $q_c = [q_2 \ q_3]^T$ for the control phase APAa, and $q_c = [q_1 \ q_3]^T$ for APAi. There exist 24 Markovian states for this fault sequence; see Table 1. Following the approach presented in [13], it is necessary to group the transition rates between the Markovian states in a transition rate matrix $A$ of dimension $24 \times 24$. The matrix $A$ is partitioned in nine submatrices of dimension $8 \times 8$

$$
A = \begin{bmatrix}
A_{AAA} & A_f & A_0 \\
A_0 & A_{APAa} & A_j \\
A_j & A_{APA} & A_{APAa}
\end{bmatrix}.
$$

The submatrix $A_{AAA}$ shows the relations between linearization points of configuration AAA, and the diagonal submatrix $A_f$ determines the probabilities of a fault occurring. After the fault occurrence, the system changes to the second line of $A$, where $A_{APAa}$ defines the relations between the linearization points in the control phase APAa, $A_0$ shows that the defective joint cannot be repaired, and the matrix $A_j$ represents how the transition rate of the system goes to the control phase APA. In the third line of $A$, $A_{APA}$ defines the relations between the linearization points in the set APAa, $A_j$ represents the possibility of the system returning to the control phase APAa, and $A_0$ represents, again, the impossibility of the defective joint being repaired. The transition rate matrix $A$ is selected as (23) with

$$
A_{AAA} = 0.09, \quad A_{APAa} = 0.08, \quad A_{APA} = 0.08,
$$

for $i \neq j$

$$
A_{AAA} = -0.73, \quad A_{APAa} = -0.76, \quad A_{APA} = -0.76,
$$

for $i = j$,

$$
A_f = 0.1I_8, \quad A_j = 0.2I_8, \quad A_0 = 0.
$$

The output-feedback $H_\infty$ control presented in [9] is implemented, according to the proposed fault-tolerant model aforementioned, in the planar three-link robot manipulator UArm II. The experiments were performed for an initial position $q(0) = [0^\circ \ 0^\circ \ 0^\circ]^T$ and for a desired final position $q(T) = [20^\circ \ 20^\circ \ 20^\circ]^T$. The initial configuration is AAA with a linearization point starting in 1; see Table 1.

To validate the fault-tolerant control proposed, a fault was introduced at $t_f = 1.5$ s. The fault-detection system proposed in [14] was implemented, considering the filtered torque prediction error approach. When the fault is detected at $t_f = 1.54$ s the Markov chain changes from the configuration AAA to the control phase APAa, keeping the related linearization point. Torque disturbances were also
introduced to verify the robustness of the controllers. For these experiments, the proportional controllers were selected heuristically as

\[
K_{P\text{AAA}} = \begin{bmatrix} 2.25 & 0 & 0 \\ 0 & 2.0 & 0 \\ 0 & 0 & 1.8 \end{bmatrix}, \quad K_{P\text{PAA}} = \begin{bmatrix} -1.8 & 0 \\ 0 & 0.5 \end{bmatrix}, \quad K_{P\text{APA}} = \begin{bmatrix} 2.0 & 0.3 \\ 0.1 & 0.7 \end{bmatrix}.
\]

The controllers were computed considering \( \alpha = 50 \) and \( \beta = 100 \) for all configurations; see (18). The best value of \( \gamma \) found was 1.5. The experimental results, joint positions, and Markovian states for the output-feedback \( \mathcal{H}_\infty \) controller are shown in Fig. 5. Even after the fault the system kept the stability with the manipulator in movement during the control reconfiguration.

### 3.2.2. AAA–PAA–PAP fault sequence

A more complete experiment was implemented considering the AAA–PAA–PAP fault sequence, represented in the fault-tolerant model by the numbers 1, 4, 5, 6, 7, and 8; see Fig. 4. With this second experiment, the guidelines to apply this procedure to manipulators with \( n \)-links and \( m \)-faults is better understood. The vector of controlled joints, \( q_c \), is chosen as \( q_c = [q_1 \ q_3]^T \) for the control phase PAA, \( q_c = [q_2 \ q_3]^T \) for PAP, \( q_c = q_1 \) for PAP\(_{u1}, q_c = q_3 \) for PAP\(_{u2} \) and \( q_c = q_2 \) for PAP\(_{u3} \). There exist 48 Markovian states for this fault sequence; see Table 2.

The matrix \( A \) is partitioned in 36 submatrices of dimension \( 8 \times 8 \)

\[
A = \begin{bmatrix}
A_{\text{AAA}} & A_f & A_0 & A_f & A_0 & A_0 \\
A_f & A_{\text{PAA}} & A_f & A_0 & A_0 & A_0 \\
A_0 & A_f & A_{\text{PAA}} & A_f & A_0 & A_0 \\
A_0 & A_0 & A_0 & A_{\text{PAP}_u} & A_f & A_0 \\
A_0 & A_0 & A_0 & A_0 & A_f & A_f \\
A_0 & A_0 & A_0 & A_0 & A_f & A_f \\
A_0 & A_0 & A_0 & A_0 & A_f & A_f \\
A_0 & A_0 & A_0 & A_0 & A_f & A_f \\
\end{bmatrix}.
\]

The matrix \( A \) is defined following the same arguments presented in Section 3.2.1 as

\[
A_{\text{AAA}(i,j)} = 0.08, \quad A_{\text{PAA}(i,j)} = 0.07, \quad A_{\text{PAP}(i,j)} = 0.07, \quad A_{\text{APA}(i,j)} = 0.08, \quad A_{\text{PAP}_u(i,j)} = 0.06, \quad A_{\text{PAP}_u(i,j)} = 0.06
\]

for \( i \neq j \)

\[
A_{\text{AAA}(i,j)} = -0.76, \quad A_{\text{PAA}(i,j)} = -0.79, \quad A_{\text{PAP}(i,j)} = -0.79, \quad A_{\text{APA}(i,j)} = -0.76, \quad A_{\text{PAP}_u(i,j)} = -0.82, \quad A_{\text{PAP}_u(i,j)} = -0.82
\]

for \( i = j \)

\[
A_f = 0.1I_8, \quad A_3 = 0.2I_8, \quad A_0 = 0.
\]

This experiment was performed for an initial position \( q(0) = [0^\circ \ 0^\circ \ 0^\circ]^T \) and for a desired final position \( q(T) = [20^\circ \ -20^\circ \ 20^\circ]^T \). The initial configuration is AAA, with the linearization point starting in 1; see Table 2. The fault-detection system used in the AAA–APA fault sequence was also adopted here to determine the fault occurrence. The first fault was introduced at \( t_f = 2.5 \) s and detected at \( t_{d1} = 2.58 \) s changing the Markovian state from AAA to PAA, maintaining its linearization point. The second fault was introduced at \( t_f = 2.5 \) s and detected at \( t_{d2} = 3.35 \) s, before joint 1 has reached its set point, resulting in the Markovian state jumping from PAA\(_u\) to PAP\(_{u1}\). After joint 1 reached its set point, at \( t_{r1} = 4.73 \) s, the Markovian state changed to PAP\(_{u2}\), and finally at \( t_{r3} = 6.75 \) s, the state changed to PAP\(_{u3}\).

Torque disturbances and an additional payload of 0.5 kg were introduced in order to check the controller robustness. The proportional controllers selected were

\[
K_{P\text{AAA}} = \begin{bmatrix} 2.25 & 0 & 0 \\ 0 & 2.0 & 0 \\ 0 & 0 & 1.8 \end{bmatrix}, \quad K_{P\text{PAA}} = \begin{bmatrix} -1.8 & 0 \\ 0 & 0.5 \end{bmatrix}, \quad K_{P\text{APA}} = -5, \quad K_{P\text{PAP}_u} = -30, \quad K_{P\text{PAP}_u} = 20.
\]

The controllers were computed considering \( \alpha = 10 \) and \( \beta = 10 \) for all configurations; see (18). The best value of \( \gamma \) found was 10. The experimental results, joint positions, and Markovian states for the output-feedback \( \mathcal{H}_\infty \) controller are shown in Fig. 6. The system kept the stability in a more critical situation than the one presented in Section 3.2.1. The use of a brake while the system operates in the Markovian states PAP\(_{u1}\)
4. Conclusion

The fault-tolerant control strategies developed in this paper, based on output feedback controllers, are complementary to the strategies presented in [1]. In this case, it is not necessary to measure the velocities of the manipulator; only the position information is required to implement these control strategies. The first one, based on an LPV output feedback controller, is subbed for use when the robot does not operate at high velocities. The second one, based on an output feedback controller for MJLS, can be applied in a more complex situation because it is not necessary to stop the robot when it is subjected to faults. By virtue of the MJLS approach being designed as depending on a set of linear models, three extra variables were introduced \( (K_p, \alpha, \beta) \) in order to pre-compensate the effects of the MJLS controller. Despite pre-compensations being common in linear model-based controllers, this represents a weak point of this controller. For the LPV controller, it is necessary to choose basic functions in order to solve the LMI and the designer can start with functions related to the robot model, usual for this kind of problem. In this paper, only free torque failures are considered. Fault tolerant systems for different fault types and concurrent faults are under investigation and will be presented in future works.

Acknowledgement

This work is supported by FAPESP (Fundação de Amparo à Pesquisa do Estado de São Paulo-Brazil) under grants 01/12943-0 and 05/005208-2.

References


Adriano Almeida Gonçalves Siqueira is Assistant Professor at the Department of Mechanical Engineering at the University of São Paulo, São Carlos, Brazil. He received a Bachelor’s degree in Mechanical Engineering (1999) and a Ph.D. degree in Electrical Engineering (2004) from the University of São Paulo. He was finalist of the Best Student Paper Award of the IEEE Conference on Decision and Control 2002 and finalist of the Best Student Paper Award of the IEEE Conference on Control Applications, 2004. His research interests are underactuated robots, cooperative robots, robust control, nonlinear control, and exoskeletons.

Marco Henrique Terra is Associate Professor at the Department of Electrical Engineering at the University of São Paulo, São Carlos, Brazil. He received his Ph.D. in Electrical Engineering in 1995 from USP. His research interests cover filtering, estimation and control theories, fault detection and isolation problems, and robotics.

Cleber Buosi received his Bachelor’s degree in Electrical Engineering in 2003 from the University of São Paulo at São Carlos. He has worked with Chemtech, a Siemens company in Brazil. His interests are in advanced control applied in industrial processes, multivariable control, process information management systems, and manufacturing execution systems.