We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

BOOK CITATION INDEX
CLARIVATE ANALYTICS
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Performance Evaluation of Fault-Tolerant Controllers in Robotic Manipulators

Claudio Urrea[1], John Kern[1,2] and Holman Ortiz[2]
[1]Departamento de Ingeniería Eléctrica, DIE,
Universidad de Santiago de Chile, USACH, Santiago
[2]Escuela de Ingeniería Electrónica y Computación,
Universidad Iberoamericana de Ciencias y Tecnología, UNICIT, Santiago
Chile

1. Introduction

Thanks to the incorporation of robotic systems, the development of industrial processes has generated a great increase in productivity, yield and product quality. Nevertheless, as far as technological advancement permits a greater automation level, system complexity also increases, with greater number of components, therefore rising the probability of failures or anomalous operation. This can result in operator's hazard, difficulties for users, economic losses, etc. Robotic automatic systems, even if helped in minimizing human operation in control and manual intervention tasks, haven't freed them from failure occurrences. Although such failures can´t be eliminated, they can be properly managed through an adequate control system, allowing to reduce degraded performance in industrial processes.
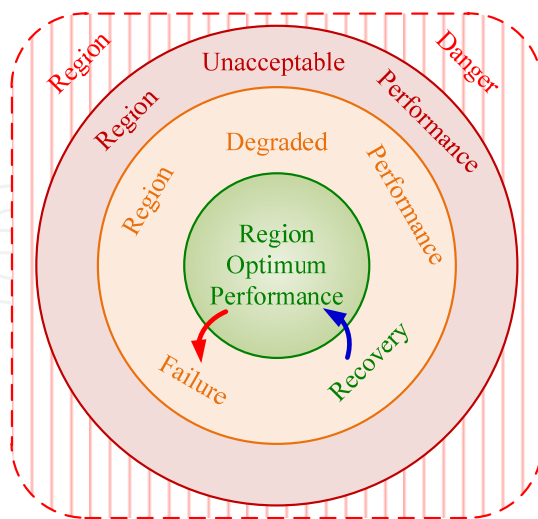

Fig. 1. Performance regions under failure occurrence

In figure 1 we see a scheme showing the different performance regions a given system can adopt when a failure occurs. If the system deviates to a degraded performance region in, presence of a failure, it can recover itself moving into an optimum performance region, or

near to it. These systems are called fault tolerant systems and have become increasingly important for robot manipulators, especially those performing tasks in remote or hazardous environments, like outer space, underwater or nuclear environments.

In this chapter we will address the concept of fault tolerance applied to a robotic manipulator. We will consider the first three degrees of freedom of a redundant SCARA-type robot, which is intended to follow a Cartesian test trajectory composed by a combination of linear segments. We developed three fault-tolerant controllers by using classic control laws: *hyperbolic sine-cosine*, calculated torque and adaptive inertia. The essays for such controllers will be run in a simulation environment developed through MatLab/Simulink software. As a performance requirement for those controllers, we considered the application of a failure consisting in blocking one of the manipulator's actuators during trajectory execution. Finally, we present a performance evaluation for each one of the above mentioned fault-tolerant controllers, through joint and Cartesian errors, by means of graphics and rms rates.

2. Fault tolerant control

The concept of fault tolerant control (Zhang & Jiang, 2003) comes first from airplane fault tolerant control; although at scientific level it appears later, as a basic aim in the first congress of IFAC SAFEPROCESS 1991, with an especially stronger development since the beginning of 21th century. Fault tolerant control can be considered both under an active or passive approach, as seen in figure 2a. Passive tolerant control is based on the ability of feedback systems to compensate perturbations, changes in system dynamics and even system failures (Puig, Quevedo, Escobet, Morcego, & C., 2004). Passive tolerant control considers a robust design of the feedback control system in order to immunize it from some specific failures (Patton, 1997). Active tolerant control is centered in on-line failure, that is, the ability to identify the failing component, determine the kind of damage, its magnitude and moment of appearance and, from this information, to activate some mechanism for
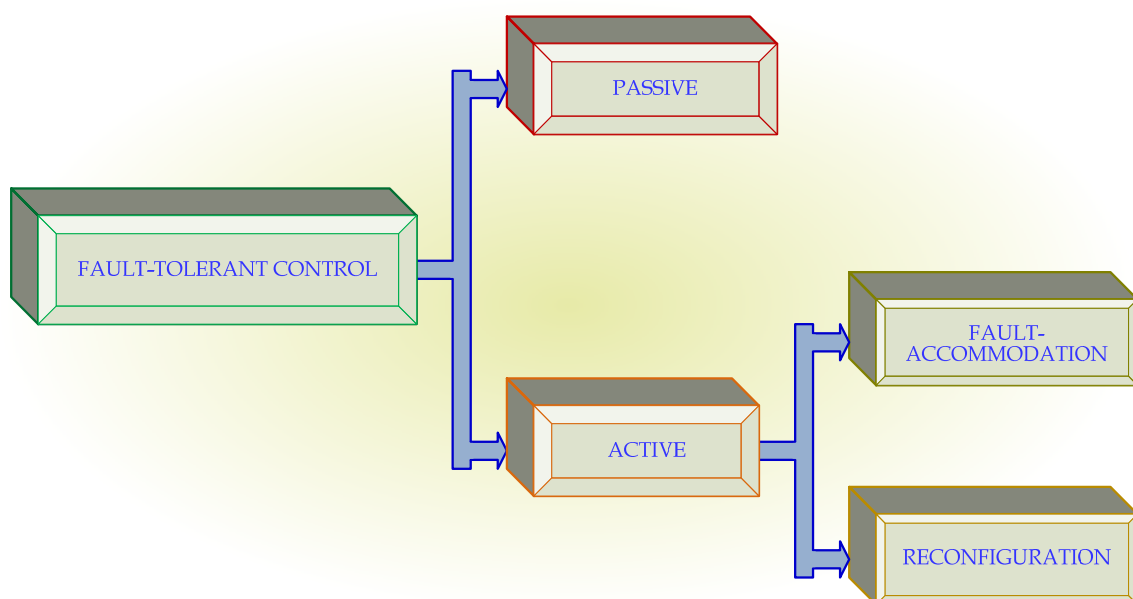

Fig. 2a. Types of fault tolerant control

rearrangement or control reconfiguration, even stopping the whole system, depending on the severity of the problem (Puig, Quevedo, Escobet, Morcego, & C., 2004).

Fault tolerant control systems (being of hybrid nature) consider the application of a series of techniques like: component and structure analysis; detection, isolation and quantification of failures; physical or virtual redundancy of sensors and/or actuators; integrated real-time supervision of all tasks performed by the fault tolerant control, as we can see in figure 2b (Blanke, Kinnaert, Lunze, & Staroswiecki, 2000).
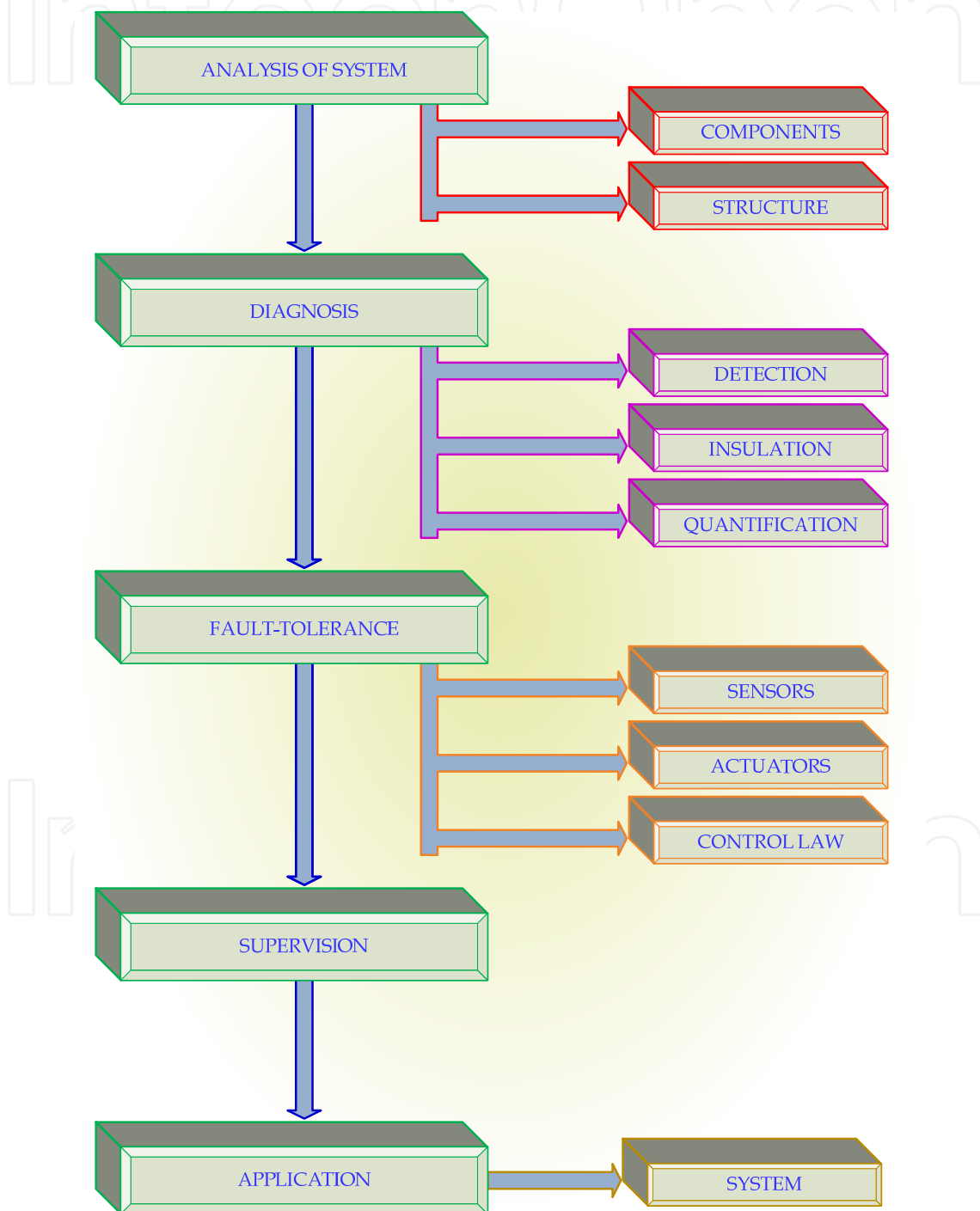

Fig. 2b. Stages included in the design of a fault tolerant control system

In the evaluation of fault tolerant controllers it is assumed that a robotic manipulator where a failure has arisen in one or more actuators, can be considered as an underactuated system, that is, a system with less actuators than the number of joints (El-Salam, El-Haweet, & and Pertew, 2005). Those underactuated systems present a greater degree of complexity compared with the simplicity of conventional robot control, being not so profoundly studied yet (Rubí, 2002). The advantages of underactuated systems have been recognized mainly because they are lighter and cheaper, with less energy consumption. Therefore, a great deal of concern is being focused on those underactuated robots (Xiujuan & Zhen, 2007). In figure 3 it is shown a diagram displaying the first three degrees of freedom of a SCARA type redundant manipulator, upon which essays will be conducted considering a failure in the second actuator, making the robot become an underactuated system.

3. SCARA-type redundant manipulator

For the evaluation of fault-tolerant controllers, we consider the first three degrees of freedom of a redundant SCARA-type robotic manipulator, with a failure occurring in one of its actuators; such a system can be considered as an underactuated system, *i.e.*, with less actuators than the number of joints (Xiujuan & Zhen, 2007). Those underactuated systems have a greater complexity compared with the simplicity of conventional robots control, and they haven't been so deeply studied yet (Rubí, 2002). The advantages of underactuated systems have been remarked mainly because they are lighter and less expensive; also having less energy consumption, consequently an increasing level of attention is being paid to underactuated robots (Xiujuan & Zhen, 2007).

In figure 3 it is shown the scheme of a redundant SCARA-type robotic manipulator, and in figure 4 we can see a diagram showing the first three degrees of freedom of such manipulator, on which the essays will be carried on.
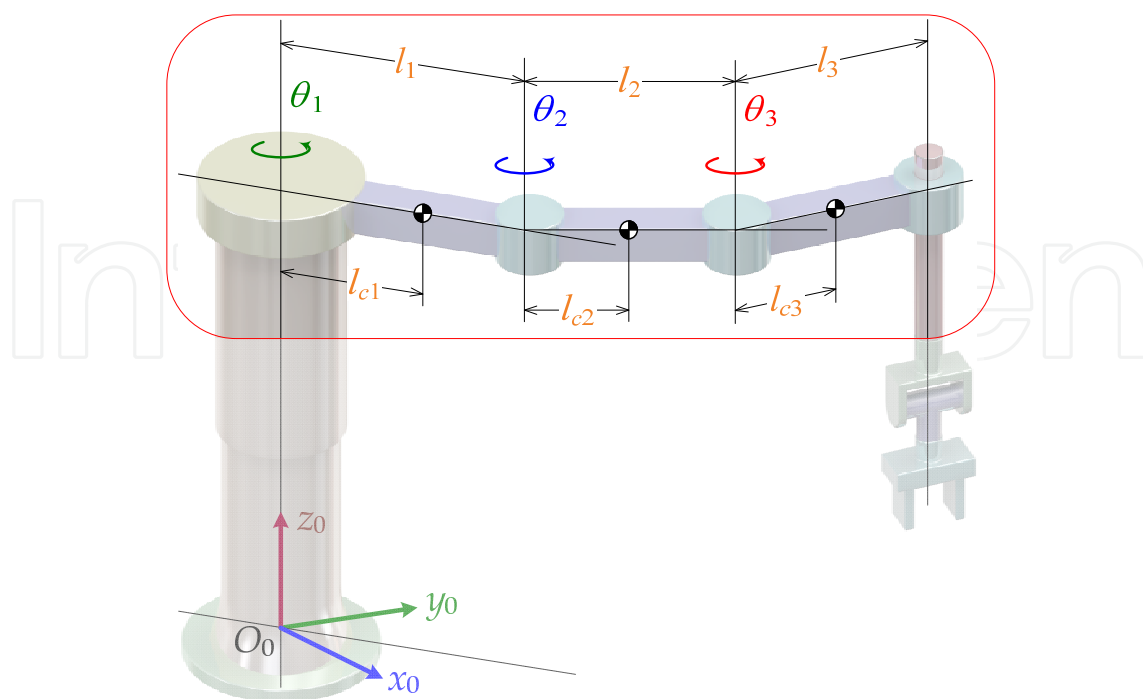

Fig. 3. Scheme of a SCARA-type redundant manipulator

The considered failure is the blocking of the second actuator, what makes this robot an underactuated system.

Having in mind the exposed manipulator, it is necessary to obtain its model; therefore we will consider that the dynamic model of a manipulator with *n* joints can be expressed through equation (1):

$$\boldsymbol{\tau} = \mathbf{M}(q)\ddot{\mathbf{q}} + \mathbf{C}(q,\dot{q}) + \mathbf{G}(q) + \mathbf{F}(\dot{q}) \tag{1}$$

where:

$\boldsymbol{\tau}$: Vector of generalized forces ($n\times1$ dimension).
$\mathbf{M}$: Inertia matrix ($n\times n$ dimension).
$\mathbf{C}$: Centrifugal and Coriolis forces vector ($n\times1$ dimension).
q : Components of joint position vector.
$\dot{q}$: Components of joint speed vector.
$\mathbf{G}$: Gravity force vector ($n\times1$ dimension).
$\ddot{\mathbf{q}}$: Joint acceleration vector ($n\times1$ dimension).
$\mathbf{F}$: Friction forces vector ($n\times1$ dimension).

Under failure conditions in actuator number 2, that is, it's blocking, the component 2 of equation (1) becomes a constant.
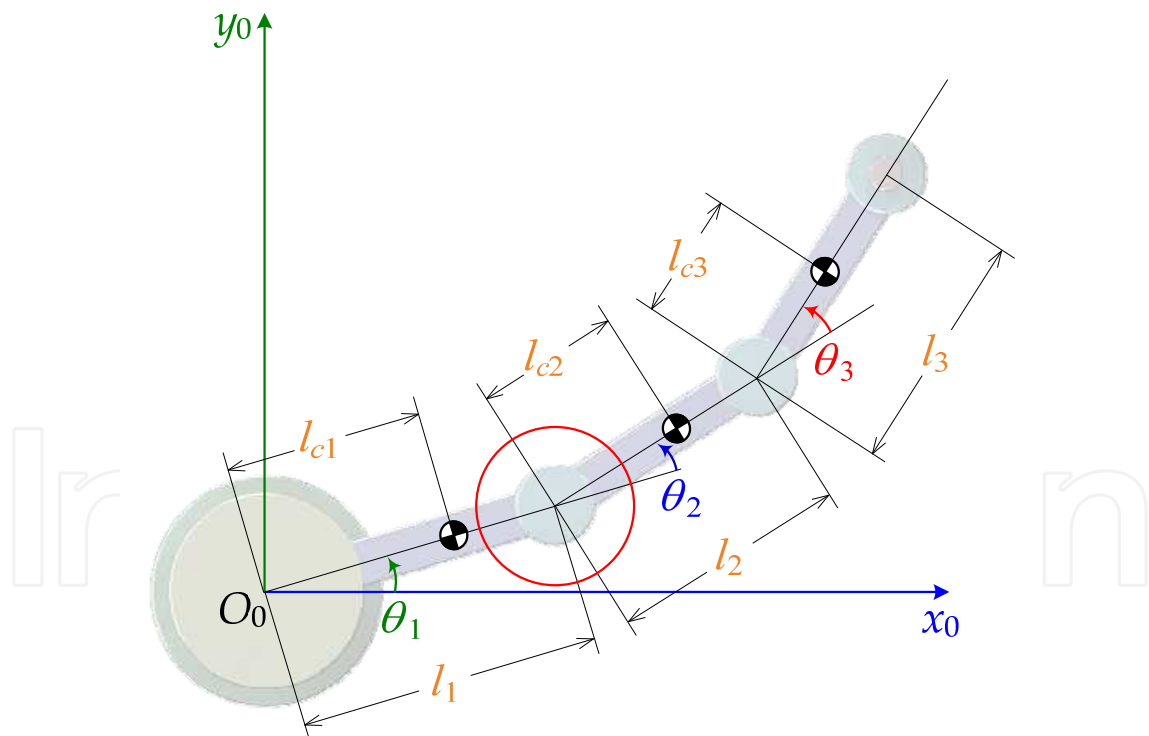

Fig. 4. Scheme of the three first DOF of a redundant SCARA-type robotic manipulator

4. Considered controllers

Considering the hybrid nature of fault tolerant control, it is proposed an active fault tolerant control having a different control law according to the status of the robotic manipulator, *i.e.*

normal or failing, with on-line sensing of possible failures and, in correspondence with this, reconfiguring the controller by selecting the most adequate control law (changing inputs and outputs).

Next, we will present a summary of the controllers considered for performance evaluation when a failure occurs in the second actuator of the previously described manipulator.

5. Fault tolerant controller: *hyperbolic sine* and *cosine*

This controller is based on the classic controller *hyperbolic sine-cosine* presented in (Barahona, Espinosa, & L., 2002), composed by a proportional part based on sine and *hyperbolic cosine* functions, a derivative part based on hyperbolic sine and gravity compensation, as shown in equation (2). The proposed fault tolerant control law includes two classic *hyperbolic sine-cosine* controllers that are "switched" to reconfigure the fault tolerant controller.

$$\boldsymbol{\tau} = \mathbf{K_p}\sinh(\mathbf{q_e})\cosh(\mathbf{q_e}) \text{ - } \mathbf{K_v}\sinh(\dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}) \tag{2}$$

$$\mathbf{q_e} = \mathbf{q_d} \text{ - } \mathbf{q} \tag{3}$$

According to equations (2) and (3):

$\mathbf{K_p}$: Proportional gain, diagonal definite positive matrix (n×n dimension).

$\mathbf{K_v}$: Derivative gain, diagonal definite positive matrix (*n*×*n* dimension).

$\mathbf{q_e}$: Joint position error vector (*n*×1 dimension).

$\mathbf{q_d}$: Desired joint position vector (*n*×1 dimension).

In (Barahona, Espinosa, & L., 2002) it is established that robotic manipulator's joint position error will tend asymptotically to zero as long as time approaches to infinite:

$$\lim_{t\to\infty} \mathbf{q_e} \to 0 \tag{4}$$

This behavior is proved analyzing equation (5) and pointing that the only equilibrium point for the system is the origin (0,0).

$$\frac{d}{dt}\begin{bmatrix} \mathbf{q_e} \\ \dot{\mathbf{q}} \end{bmatrix} = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}$$

$$a_1 = \text{-}\dot{\mathbf{q}} \tag{5}$$

$$a_2 = \mathbf{M}(\mathbf{q})^{\text{-}1}\left(\mathbf{K_p}\sinh(\mathbf{q_e})\cosh(\mathbf{q_e})\text{-}\mathbf{K_v}\sinh(\dot{\mathbf{q}})\text{-}\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}\right)$$

6. Fault tolerant controller: Computed torque

Another active fault tolerant controller analyzed here uses a control law by computed torque, consisting in the application of a torque in order to compensate the centrifugal, Coriolis, gravity and friction effects, as shown in equation (6).

$$\boldsymbol{\tau} = \hat{\mathbf{M}}(q)\left(\ddot{\mathbf{q}}_\mathbf{d} + \mathbf{K_v}\dot{\mathbf{q}}_\mathbf{e} + \mathbf{K_p}\mathbf{q_e}\right) + \hat{\mathbf{C}}(q,\dot{q}) + \hat{\mathbf{G}}(q) + \hat{\mathbf{F}}(\dot{q}) \tag{6}$$

where:

$\hat{\mathbf{M}}$: Estimation of inertia matrix (*n*×*n* dimension).

$\hat{\mathbf{C}}$: Estimation of centrifugal and Coriolis forces vector ($n\times1$ dimension).
$\hat{\mathbf{G}}$: Estimation of gravity force vector ($n\times1$ dimension).
$\hat{\mathbf{F}}$: Estimation of friction forces vector ($n\times1$ dimension).

$$\mathbf{K}_{v} = \begin{bmatrix} \mathrm{K}_{v1} & & & \\ & \mathrm{K}_{v2} & & \\ & & \ddots & \\ & & & \mathrm{K}_{vn} \end{bmatrix} \tag{7}$$

$\mathbf{K_v}$: Diagonal definite positive matrix ($n\times n$ dimension).

$$\mathbf{K}_{p} = \begin{bmatrix} \mathrm{K}_{p1} & & & \\ & \mathrm{K}_{p2} & & \\ & & \ddots & \\ & & & \mathrm{K}_{pn} \end{bmatrix} \tag{8}$$

$\mathbf{K_p}$: Diagonal definite positive matrix ($n\times n$ dimension).
$\ddot{\mathbf{q}}_{\mathbf{d}}$: Desired joint acceleration vector ($n\times1$ dimension).

$$\dot{\mathbf{q}}_{\mathbf{e}} = \dot{\mathbf{q}}_{\mathbf{d}} - \dot{\mathbf{q}} \tag{9}$$

$\dot{\mathbf{q}}_{\mathbf{e}}$: Joint speed error vector ($n\times1$ dimension).
If estimation errors are little, joint errors near to a linear equation, as shown in equation (10).

$$\ddot{\mathbf{q}}_{\mathbf{e}} + \mathbf{K}_{v}\dot{\mathbf{q}}_{\mathbf{e}} + \mathbf{K}_{p}\mathbf{q}_{\mathbf{e}} \approx 0 \tag{10}$$

7. Fault tolerant controller: Adaptive inertia

The fault tolerant control under examination is based on an adaptive control law, namely: adaptive inertia (Lewis, Dawson, & Abdallah, 2004), (Siciliano & Khatib, 2008), for what it is necessary to consider the manipulator dynamic model in the form expressed in equation (11). The term corresponding to centrifugal and Coriolis forces is expressed through a matrix $\mathbf{V_m}$.

$$\tau = \mathbf{M}(q)\ddot{\mathbf{q}} + \mathbf{V}_{\mathbf{m}}(q,\dot{q})\dot{\mathbf{q}} + \mathbf{G}(q) + \mathbf{F}(\dot{q}) \tag{11}$$

In this case, we define an auxiliary error signal $\mathbf{r}$ and its derivative $\dot{\mathbf{r}}$, as shown in equations (12) and (13), respectively:

$$\mathbf{r} = \Lambda\mathbf{q}_{\mathbf{e}} + \dot{\mathbf{q}}_{\mathbf{e}} \tag{12}$$

$$\dot{\mathbf{r}} = \Lambda\dot{\mathbf{q}}_{\mathbf{e}} + \ddot{\mathbf{q}}_{\mathbf{e}} \tag{13}$$

where:
Λ : Diagonal definite positive matrix ($n\times n$ dimension).

$$\Lambda = \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{bmatrix} \tag{14}$$

When replacing equations (3), (9), (12) and (13) into expression (11), we obtain:

$$\tau = \mathbf{M}(q)(\ddot{\mathbf{q}}_d + \Lambda\dot{\mathbf{q}}_e) + \mathbf{V_m}(q,\dot{q})(\dot{\mathbf{q}}_d + \Lambda\mathbf{q}_e) + \mathbf{G}(q) + \mathbf{F}(\dot{q}) - \mathbf{M}(q)\dot{\mathbf{r}} - \mathbf{V_m}(q,\dot{q})\mathbf{r} \tag{15}$$

And making the following matching:

$$\mathbf{Y}(\bullet)\boldsymbol{\varphi} = \mathbf{M}(\mathbf{q})(\ddot{\mathbf{q}}_d + \Lambda\dot{\mathbf{q}}_e) + \mathbf{V_m}(q,\dot{q})(\dot{\mathbf{q}}_d + \Lambda\mathbf{q}_e) + \mathbf{G}(q) + \mathbf{F}(\dot{q}) \tag{16}$$

where:

$$\mathbf{Y}(q,\dot{q},q_d,\dot{q}_d,\ddot{q}_d) = \begin{bmatrix} Y_{11} & Y_{12} & \cdots & Y_{1n} \\ Y_{21} & Y_{22} & \cdots & Y_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ Y_{n1} & Y_{n2} & \cdots & Y_{nn} \end{bmatrix} \tag{17}$$

$\mathbf{Y}(\bullet)$: Regression matrix ($n{\times}n$ dimension).
$\boldsymbol{\varphi}$: Parameter vector ($n{\times}1$ dimension).
With these relationships, expression (15) can be rewritten in the following way:

$$\tau = \mathbf{Y}(\bullet)\boldsymbol{\varphi} - \mathbf{M}(q)\dot{\mathbf{r}} - \mathbf{V_m}(q,\dot{q})\mathbf{r}\,\mathbf{Y}(\bullet) \tag{18}$$

And the control torque is expressed through equation (19):

$$\tau = \mathbf{Y}(\bullet)\hat{\boldsymbol{\varphi}} + \mathbf{K_v}\mathbf{r} \tag{19}$$

where:
$\hat{\boldsymbol{\varphi}}$: Parameter estimation vector ($n{\times}1$ dimension).
$\mathbf{K_v}$: Diagonal definite positive matrix ($n{\times}n$ dimension).
The adaptive control updating rule can be expressed by:

$$\dot{\hat{\boldsymbol{\varphi}}} = -\dot{\tilde{\boldsymbol{\varphi}}} = \boldsymbol{\Gamma}\mathbf{Y}^T(\cdot)\mathbf{r} \tag{20}$$

where:
$\boldsymbol{\Gamma}$: Diagonal definite positive matrix ($n{\times}n$ dimension).

8. Fault tolerant control simulator

The three above mentioned control laws, along with the dynamic model of the redundant SCARA-type manipulator considering the first three degrees of freedom (Addendum A), are run under the simulation structure shown in figure 5, where we can see the hybrid nature of this kind of controller.

In Addendum B we show the set of parameter values employed in the manipulator dynamic model, and the gains considered for each kind of fault tolerant controller.
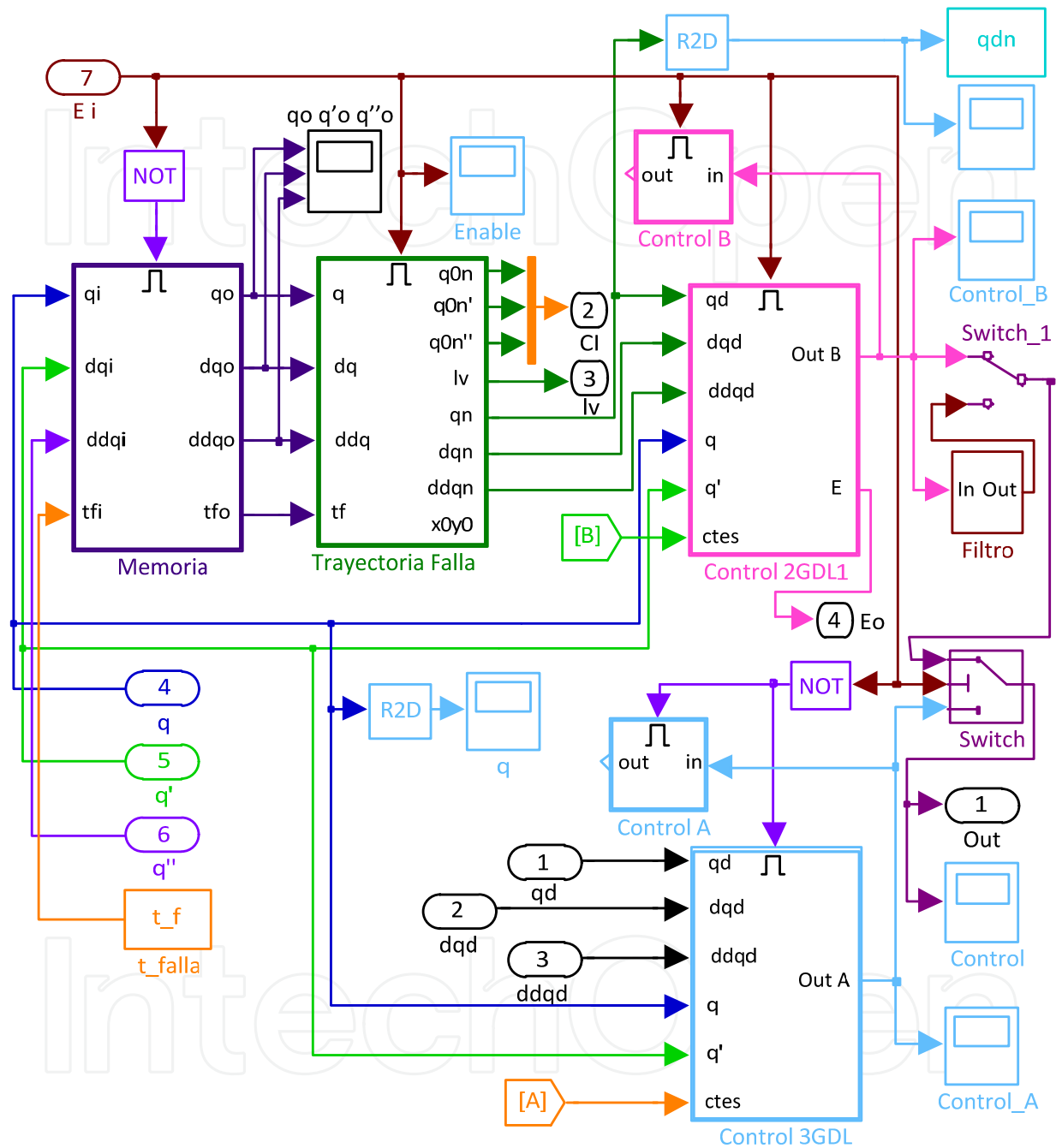

Fig. 5. Block diagram of the structure of the fault tolerant controller used to test the above mentioned control laws

9. Results

After establishing the control laws being utilized, we determine the trajectory to be entered in the control system to carry out the corresponding performance tests of fault tolerant control algorithms. This trajectory is displayed in figure 6.
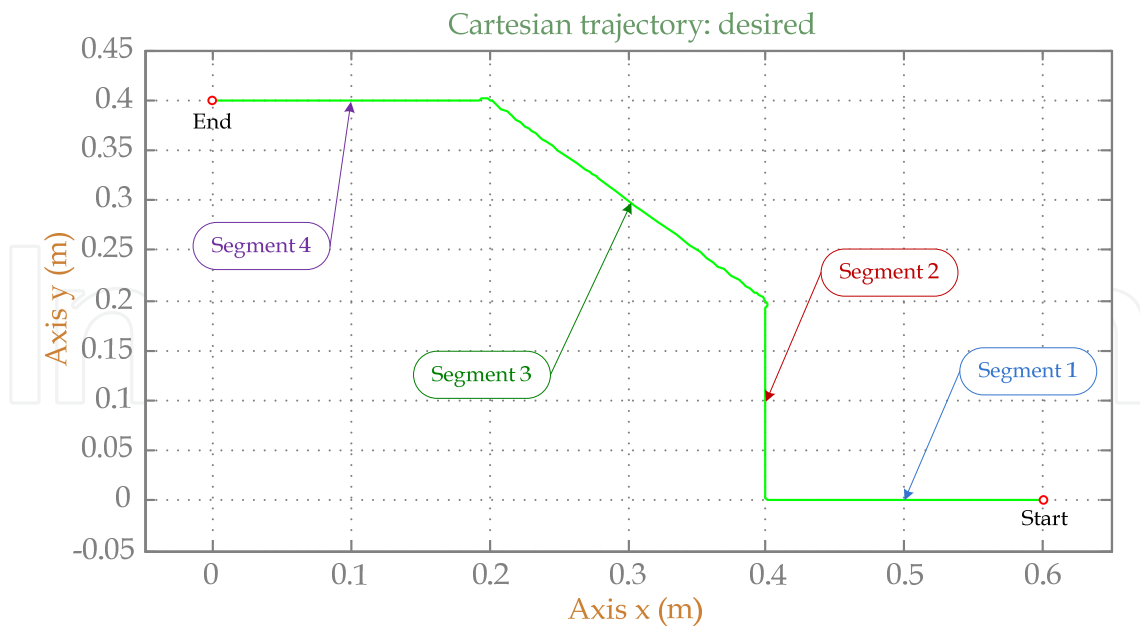
Fig. 6. Cartesian test trajectory

Figures 7 and 8a show the curves corresponding to the differences between desired and real joint trajectories, and between desired and real Cartesian trajectories, respectively, all this under *hyperbolic sine-cosine* fault tolerant control when there is a failure in actuator 2 at 0.5 sec from initiating movement.

Where:

e_{q1} :   Joint trajectory error, joint 1.
e_{q2} :   Joint trajectory error, joint 2.
e_{q3} :   Joint trajectory error, joint 3.
e_x :   Cartesian trajectory error, x axis.
e_y :   Cartesian trajectory error, y axis.
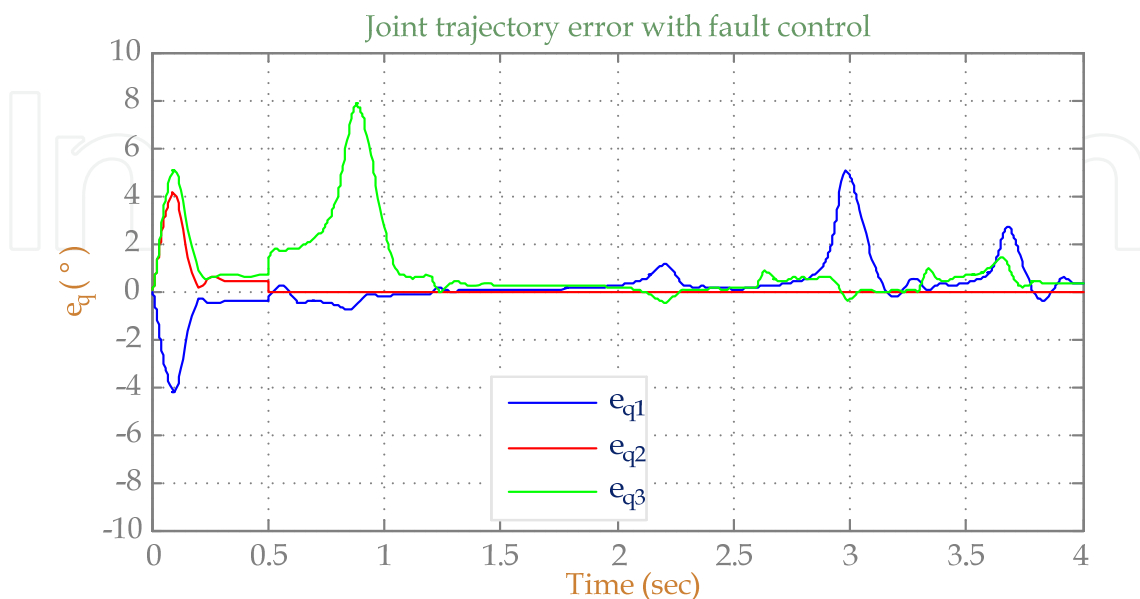
Fig. 7. Joint trajectory error with fault control using *hyperbolic sine-cosine* controller
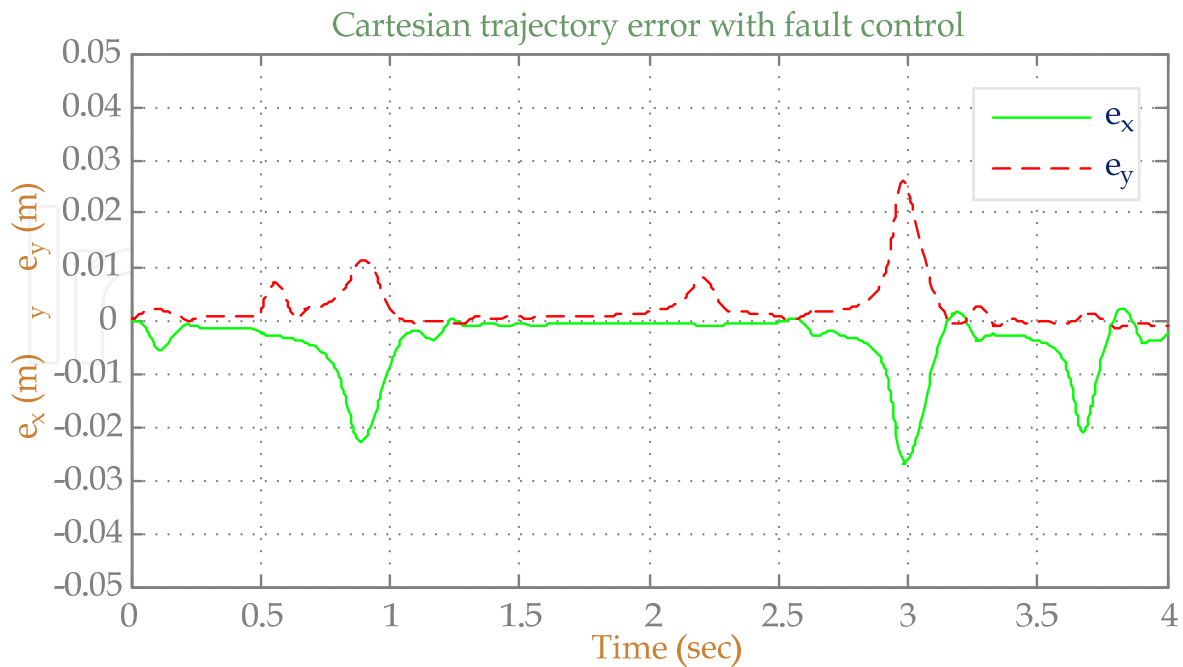
Fig. 8a. Cartesian trajectory error with fault control using *hyperbolic sine-cosine* controller

The performance of fault tolerant controller by computed torque is shown in figures 8b and 9, displaying the curves for joint and Cartesian errors under the same failure conditions than the previous case.
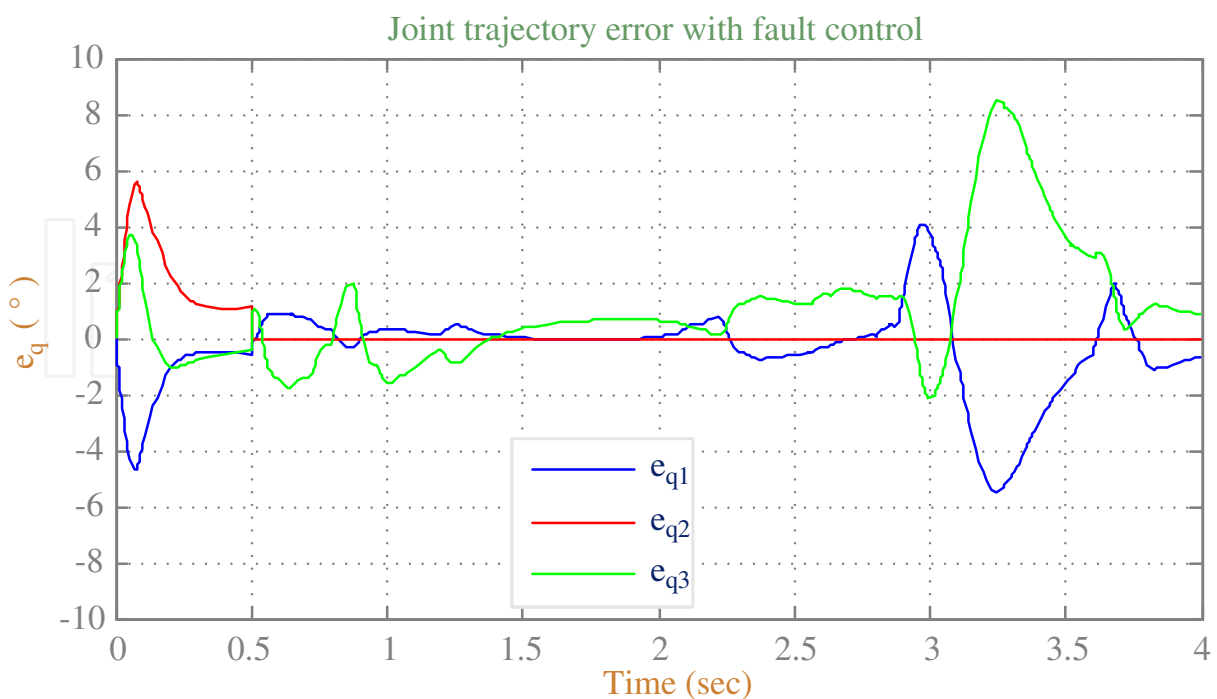

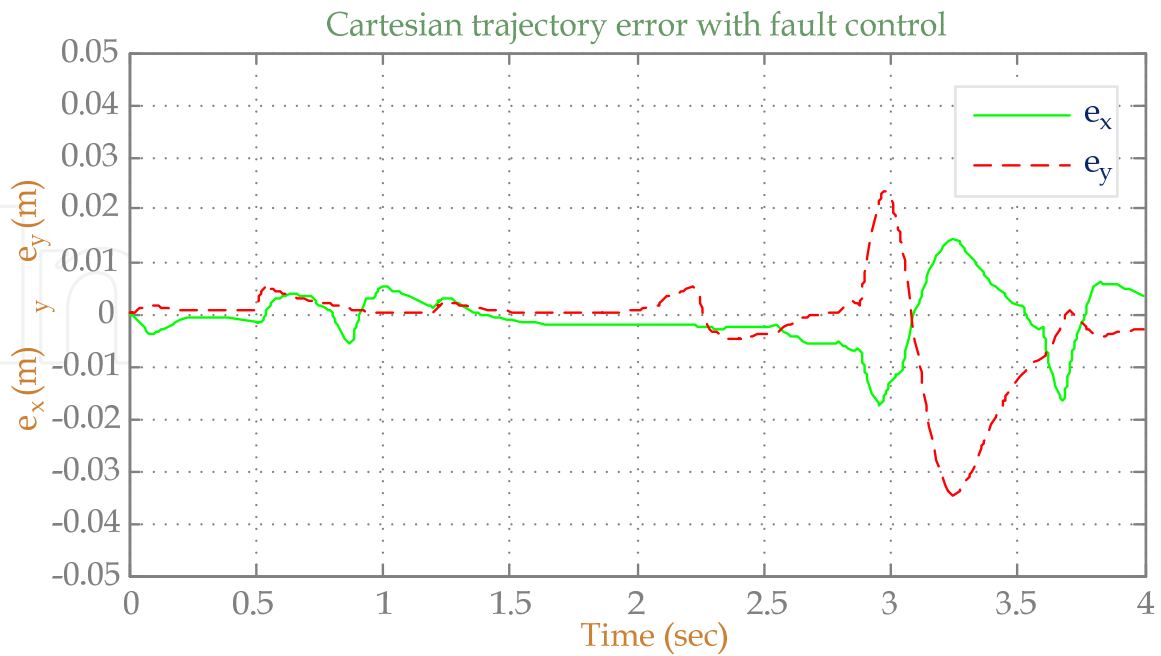Fig. 8b. Joint trajectory error with fault control using computed torque controller

Fig. 9. Cartesian trajectory error with fault control using computed torque controller

In figures 10 and 11 we can see charts displaying respectively joint and Cartesian errors corresponding to the performance of fault tolerant controller by adaptive inertia, under the same failure conditions imposed to the previous controllers.
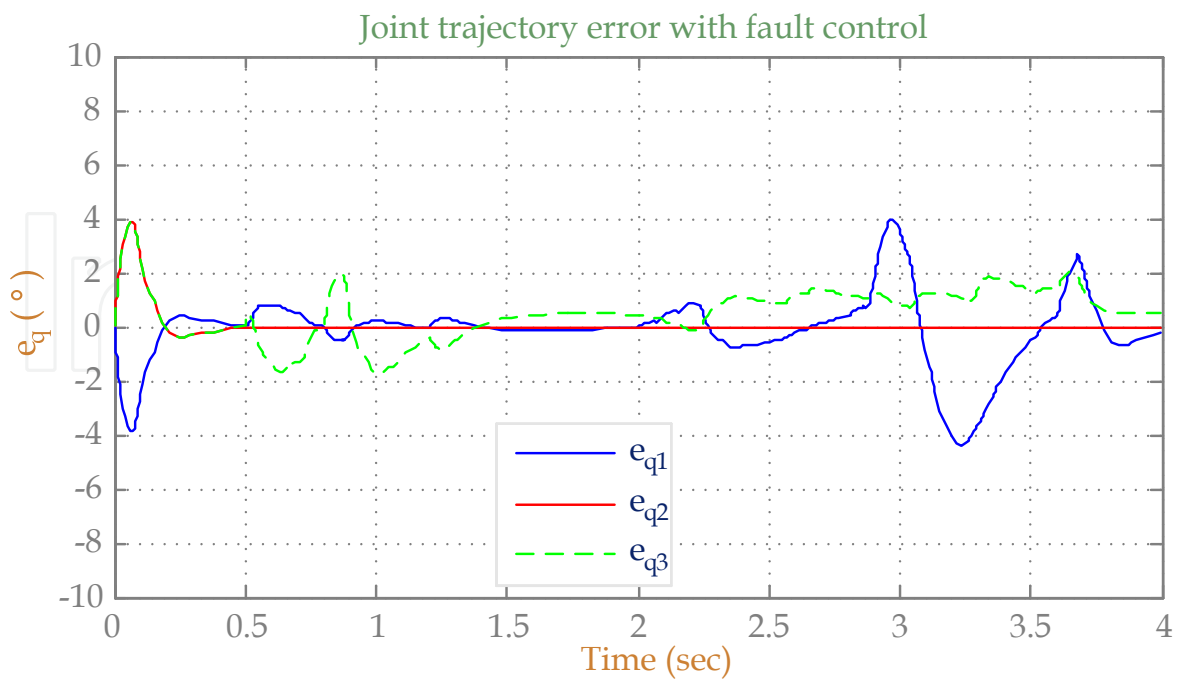

Fig. 10. Joint trajectory error with fault control using adaptive inertia controller
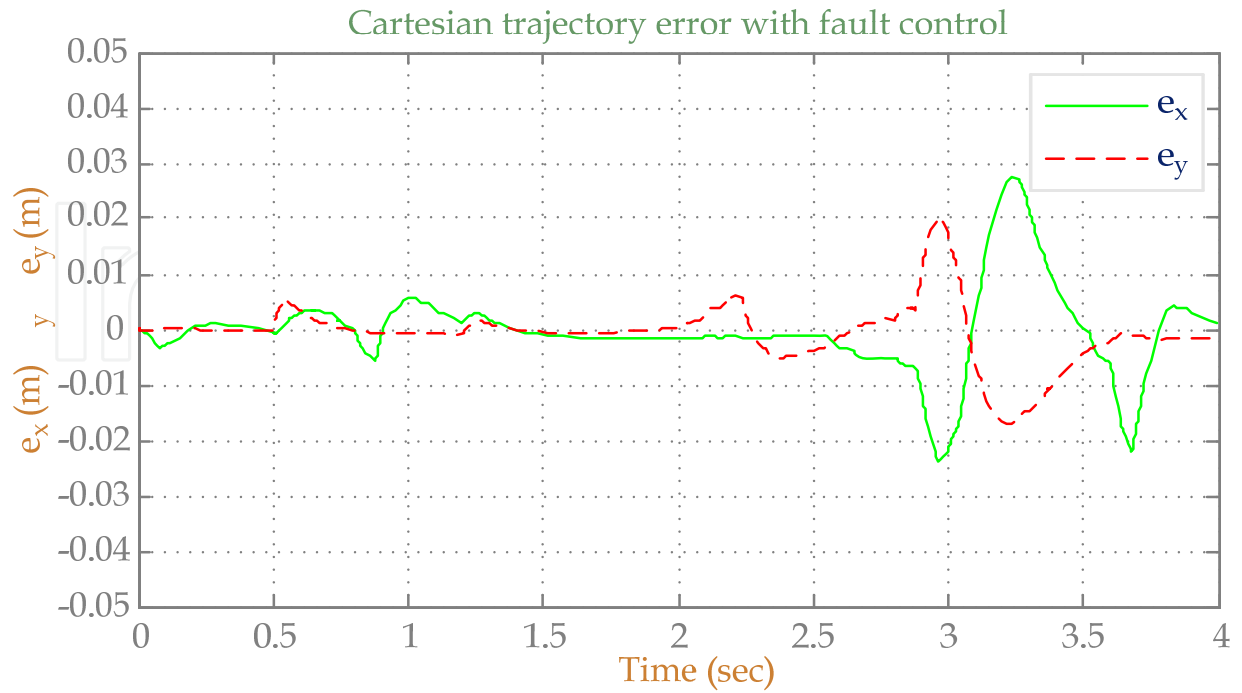
Fig. 11. Cartesian trajectory error with fault control using adaptive inertia controller
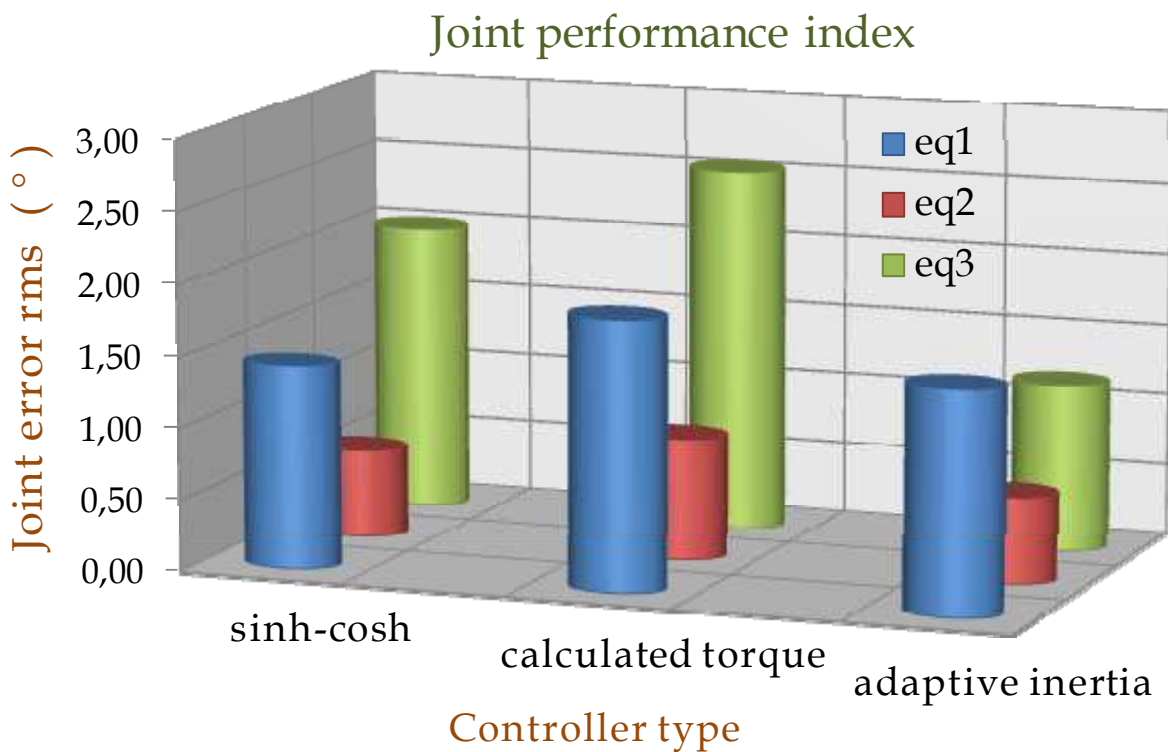

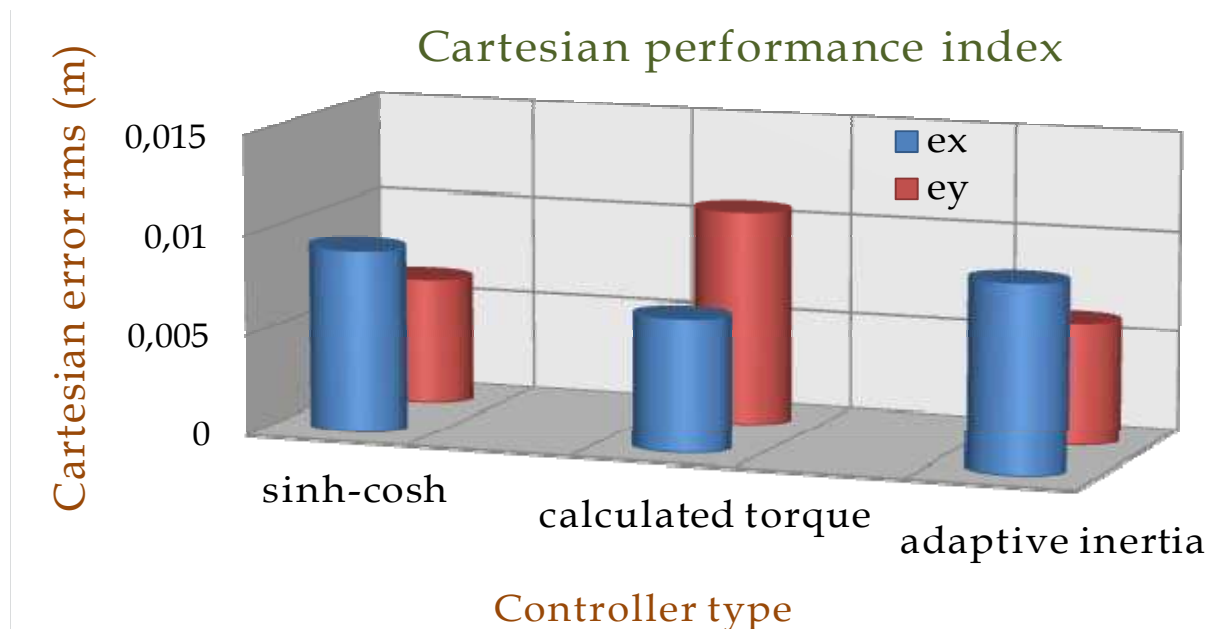Fig. 12. Performance index corresponding to joint trajectory

Fig. 13. Performance index corresponding to Cartesian trajectory

Finally, in figures 12 and 13 it is shown a performance summarization of the analyzed fault tolerant controllers in terms of joint and Cartesian *mean square root* errors, accordingly to equation (21)

$$rms \;=\; \sqrt{\frac{1}{n}\sum_{i=1}^{n} e_i^{\,2}} \qquad\qquad (21)$$

Where e_i represents articular trajectory as well as Cartesian errors.

10. Conclusions

In this work we presented a performance evaluation of three fault tolerant controllers based on classic control techniques: hyperbolic sine-cosine, calculated torque and adaptive inertia. Those fault tolerant controllers were applied on the first three degrees of freedom of a redundant SCARA-type robotic manipulator. The different system stages were implemented in a simulator developed using MatLab/Simulink *software*, allowing to represent the robotic manipulator behavior following a desired trajectory, when blocking of one of its actuators occurs. In this way we obtained the corresponding simulation curves. From the obtained results, we observed that the adaptive inertia fault tolerant controller have errors with less severe maximums than the other controllers, resulting in more homogeneous manipulator movements. We noticed that greater errors were produced with the calculated torque fault tolerant controller, both for maximums and *rms*. Consequently, the best performance is obtained when using the adaptive inertia controller, as shown in figures 14 and 15. It is also remarkable that the hyperbolic sine-cosine fault tolerant controller have a lesser implementation complexity, since it does not require the second derivative of joint position. This can be a decisive factor in the case of not having high performance processors.

11. Further developments

Thanks to the development of this work, from the implemented simulation tools and the obtained results, fault tolerant control systems essays are being currently carried out, in order to apply them to actual robotic systems, with and without link redundancy, like the SCARA-type robots shown in figure 14 and figure 15, respectively.
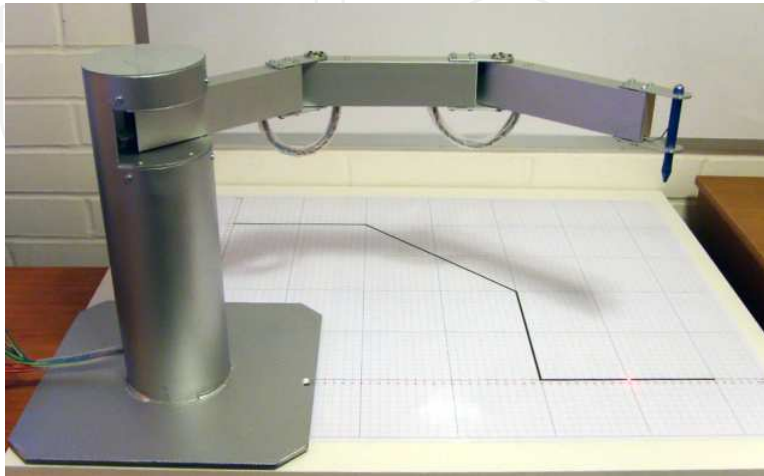

Fig. 14. SCARA-type redundant robot, DIE-USACH

Fig. 15. SCARA-type robot, DIE-USACH

12. Addendum A: Manipulator's dynamic model

The manipulator's dynamic model is given by equations a1 to a14.

$$\mathbf{M} = \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix} \tag{a1}$$

$$M_{11} = I_{1zz} + I_{2zz} + I_{3zz} + m_1 l_{c1}^2 + m_2\left(l_1^2 + l_{c2}^2\right) + m_2 2 l_1 l_{c2}\cos\theta_2 + \ldots$$
$$m_3\left(l_1^2 + l_2^2 + l_{c3}^2 + 2 l_1 l_2\cos\theta_2 + 2 l_2 l_{c3}\cos\theta_3 + 2 l_1 l_{c3}\cos\left(\theta_2 + \theta_3\right)\right) \tag{a2}$$

$$M_{21} = M_{12} = I_{2zz} + I_{3zz} + m_2\left(l_{c2}^2 + l_1 l_{c2}\cos\theta_2\right) + \ldots$$
$$m_3\left(l_2^2 + l_{c3}^2 + l_1 l_2\cos\theta_2 + 2 l_2 l_{c3}\cos\theta_3 + l_1 l_{c3}\cos\left(\theta_2 + \theta_3\right)\right) \tag{a3}$$

$$M_{31} = M_{13} = I_{3zz} + m_3\left(l_{c3}^2 + l_2 l_{c3}\cos\theta_3\right) + m_3 l_1 l_{c3}\cos\left(\theta_2 + \theta_3\right) \tag{a4}$$

$$M_{22} = I_{2zz} + I_{3zz} + m_2 l_{c2}^2 + m_3\left(l_2^2 + l_{c3}^2 + 2 l_2 l_{c3}\cos\theta_3\right) \tag{a5}$$

$$M_{32} = M_{23} = I_{3zz} + m_3\left(l_{c3}^2 + l_2 l_{c3}\cos\theta_3\right) \tag{a6}$$

$$M_{33} = I_{3zz} + m_3 l_{c3}^2 \tag{a7}$$

$$\mathbf{C} = \begin{bmatrix} C_{11} & C_{21} & C_{31} \end{bmatrix}^T \tag{a8}$$

$$C_{11} = -2 l_1\left(m_2 l_{c2}\sin\theta_2 + m_3 l_2\sin\theta_2\right)\dot{\theta}_1\dot{\theta}_2 - 2 l_1 m_3 l_{c3}\sin\left(\theta_2 + \theta_3\right)\dot{\theta}_1\dot{\theta}_2 + \ldots$$
$$- m_2 l_1 l_{c2}\sin\theta_2 \cdot \dot{\theta}_2^2 + m_3\left(l_1 l_2\sin\theta_2 + l_1 l_{c3}\sin\left(\theta_2 + \theta_3\right)\right)\dot{\theta}_2^2 + \ldots$$
$$- 2 l_{c3} m_3\left(l_2\sin\theta_3 + l_1\sin\left(\theta_2 + \theta_3\right)\right)\dot{\theta}_1\dot{\theta}_3 + \ldots$$
$$- 2 m_3 l_{c3}\left(l_2\sin\theta_3 + l_1\sin\left(\theta_2 + \theta_3\right)\right)\dot{\theta}_2\dot{\theta}_3 + \ldots$$
$$m_3\left(-l_2 l_{c3}\sin\theta_3 - l_1 l_{c3}\sin\left(\theta_2 + \theta_3\right)\right)\dot{\theta}_3^2 \tag{a9}$$

$$C_{21} = m_3\left(l_1 l_2\sin\theta_2 + l_1 l_{c3}\sin\left(\theta_2 + \theta_3\right)\right)\dot{\theta}_1^2 + m_2 l_1 l_{c2}\sin\theta_2 \cdot \dot{\theta}_1^2 + \ldots$$
$$- 2 m_3 l_2 l_{c3}\sin\theta_3 \cdot \dot{\theta}_1\dot{\theta}_3 - 2 m_3 l_2 l_{c3}\sin\theta_3 \cdot \dot{\theta}_2\dot{\theta}_3 - m_3 l_2 l_{c3}\sin\theta_3 \cdot \dot{\theta}_3^2 \tag{a10}$$

$$C_{31} = m_3\left(l_2 l_{c3}\sin\theta_3 + l_1 l_{c3}\sin\left(\theta_2 + \theta_3\right)\right)\dot{\theta}_1^2 + \ldots$$
$$2 m_3 l_2 l_{c3}\sin\theta_3 \cdot \dot{\theta}_1\dot{\theta}_2 + m_3 l_2 l_{c3}\sin\theta_3 \cdot \dot{\theta}_2^2 \tag{a11}$$

$$C_{31} = m_3\left(l_2 l_{c3}\sin\theta_3 + l_1 l_{c3}\sin\left(\theta_2 + \theta_3\right)\right)\dot{\theta}_1^2 + 2 m_3 l_2 l_{c3}\sin\theta_3 \cdot \dot{\theta}_1\dot{\theta}_2 + \ldots$$
$$m_3 l_2 l_{c3}\sin\theta_3 \cdot \dot{\theta}_2^2 \tag{a12}$$

$$\mathbf{G} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T \tag{a13}$$

$$\mathbf{F} = \begin{bmatrix} F_{11} & F_{21} & F_{31} \end{bmatrix}^T \tag{a14}$$

where:
m_1 : First link mass.
m_2 : Second link mass.

m_3 : Third link mass.
l_1 : First link length.
l_2 : Second link length.
l_3 : Third link length.
l_{c1} : Length from 1st link origin to its centroid.
l_{c2} : Length from 2nd link origin to its centroid.
l_{c3} : Length from 3rd link origin to its centroid.
I_{1zz} : 1st link inertial momentum with respect to the first z axis of its joint.
I_{2zz} : 2nd link inertial momentum with respect to the first z axis of its joint.
I_{3zz} : 3rd link inertial momentum with respect to the first z axis of its joint.

13. Addendum B: Considered parameter values

Parameter values considered for the manipulator as well as controller gains values are shown in tables B1 and B2, respectively.

Link 1		Link 2		Link 3		Units
l_1	= 0.2	l_2	= 0.2	l_3	= 0.2	$[\text{m}]$
l_{c1}	= 0.0229	l_{c2}	= 0.0229	l_{c3}	= 0.0983	$[\text{m}]$
m_1	= 2.0458	m_2	= 2.0458	m_3	= 6.5225	$[\text{kg}]$
I_{1zz}	= 0.0116	I_{2zz}	= 0.0116	I_{3zz}	= 0.1213	$\left[\text{kg·m}^2\right]$
F_{v1}	= 0.025	F_{v2}	= 0.025	F_{v3}	= 0.025	$\left[\dfrac{\text{N·m·s}}{\text{rad}}\right]$
F_{eca1}	= 0.05	F_{eca2}	= 0.05	F_{eca3}	= 0.05	$[\text{N·m}]$
F_{ecb1}	= -0.05	F_{ecb2}	= -0.05	F_{ecb3}	= -0.05	$[\text{N·m}]$

Table B1. Considered parameters for the manipulator

Constants	Controller Type		
	Hyperbolic Sine-Cosine	Computed Torque	Adaptive Inertia
K_{p1}, K_{p2}, K_{p3}	400, 300, 200	800, 800, 800	—
K_{v1}, K_{v2}, K_{v3}	3, 2, 1	140, 140, 140	20, 20, 20
$\lambda_1, \lambda_2, \lambda_3$	—	—	8, 8, 8
$\gamma_1, \gamma_2, \gamma_3$	—	—	0.1, 0.1, 0.1

Table B2. Controller gains

14. References

Barahona, J., Espinosa & L., C.F., 2002. Evaluación Experimental de Controladores de Posición tipo Saturados para Robot Manipuladores. In *Congreso Nacional de Electrónica, Centro de Convenciones William o Jenkins*. Puebla. México, 2002.

Blanke, M., Kinnaert, M., Lunze, J. & Staroswiecki, M., 2000. What is Fault-Tolerant Control. In *IFAC Symposium on Fault Detection, Supervision and Safety for Technical Process - SAFEPROCESS 2000*. Budapest, 2000. Springer-Verlag Berlin Heidelberg.

El-Salam, A., El-Haweet, W. & and Pertew, A., 2005. Fault Tolerant Kinematic Controller Design for Underactuated Robot Manipulators. In *The Automatic Control and Systems Engineering Conference*. Cairo, 2005.

Lewis, F., Dawson, D. & Abdallah, C., 2004. *Robot Manipulator Control Theory and Practice*. New York: Marcel Dekker, Inc.

Patton, R.J., 1997. Fault-Tolerant Control: The 1997 Situation. In *Proc. IFAC Symposium Safeprocess*. Kingston Upon Hull, 1997.

Puig, V. et al., 2004. Control Tolerante a Fallos (Parte I): Fundamentos y Diagnóstico de Fallos. *Revista Iberoamericana de Automática e Informática Industrial*, 1(1), pp.15-31.

Rubí, J., 2002. *Cinemática, Dinámica y Control de Robots Redundantes y Robots Subactuados*. Tesis Doctoral. San Sebastián, España: Universidad de Navarra.

Siciliano, B. & Khatib, O., 2008. *Handbook of Robotics*. Berlin, Heidelberg: Springer-Verlag.

Xiujuan, D. & Zhen, L., 2007. *Underactuated Robot Dynamic Modeling and Control Based on Embedding Model*. In *12th IFToMM World Congress*. Besançon. France, 2007.

Zhang, Y. & Jiang, J., 2003. Bibliographical Review on Reconfigurable Fault-Tolerant Control Systems. In *Proceedings IFAC SAFEPROCESS*. Washington, D.C., USA, 2003.
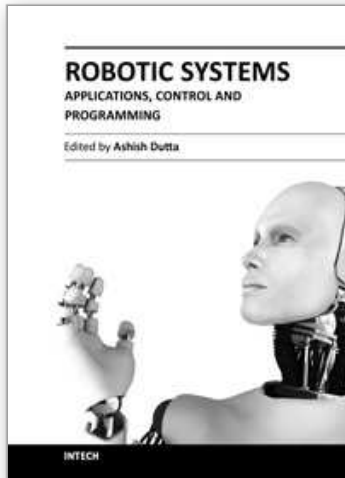
Robotic Systems - Applications, Control and Programming

Edited by Dr. Ashish Dutta

This book brings together some of the latest research in robot applications, control, modeling, sensors and algorithms. Consisting of three main sections, the first section of the book has a focus on robotic surgery, rehabilitation, self-assembly, while the second section offers an insight into the area of control with discussions on exoskeleton control and robot learning among others. The third section is on vision and ultrasonic sensors which is followed by a series of chapters which include a focus on the programming of intelligent service robots and systems adaptations.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

INTECH
open science | open minds