Robotica (2015) volume 33, pp. 1886–1898. © Cambridge University Press 2014
doi:10.1017/S0263574714001143

Human - robot collision detection and identification based on fuzzy and time series modelling

Fotios Dimeas†*, L. D. Avendaño-Valencia‡ and Nikos Aspragathos†

†*Department of Mechanical Engineering & Aeronautics, Robotics Group, University of Patras, Greece*
‡*Department of Mechanical Engineering & Aeronautics, Stochastic Mechanical Systems & Automation Lab, University of Patras, Greece*

(Accepted April 15, 2014. First published online: May 9, 2014)

SUMMARY
In this paper, two methods are proposed and implemented for collision detection between the robot and a human based on fuzzy identification and time series modelling. Both methods include a collision detection system for each joint of the robot that is trained to approximate the external torque. In addition, the proposed methods are able to detect the occurrence of a collision, the link that collided and to some extent the magnitude of the collision without using the explicit model of the robot. Since the speed of the detection is of critical importance for mitigating the danger, attention is paid to recognise a collision as soon as possible. Experimental results conducted with a KUKA LWR manipulator using two joints in planar motion, verify the validity on both methods.

KEYWORDS: Robot collision detection; Safe human-robot interaction; Fuzzy modelling; Time series modelling.

1. Introduction

When robots operate in unstructured environments and share the same workspace with humans, safety issues are of primary concern. Physical human-robot interaction (pHRI) can lead to unexpected collisions that can harm humans and to mitigate that risk, a method for avoidance or detection of the collision should be available. In the relevant literature, there is a variety of methods dealing with this problem either on pre-impact or during the impact phase. The performance of a collision detection system can be evaluated from the following criteria: fast and accurate detection of the collision force, use of a small number of sensors, ability to generalise, high success rate.

On the pre-impact phase, a collision can be foreseen and avoided during motion planning by having knowledge of the environment.[1,2] 3D vision techniques[3] that allow the monitoring of the human pose can cope with this problem, but they have high computational cost and blind spots can limit their reliability. Another approach is based on using proximity sensors on the robot body that are able to identify human tissue through capacitive sensing.[4] Although this method identifies the point of an imminent collision, a special treatment of the robot body is required for the sensor installation and the cost is quite high. During the impact, a contact of the robot with the environment can cause high reaction forces that influence significantly its the dynamic behaviour. Passive safety features like compliant joints, actuators and lightweight manipulators cannot always guarantee harmless collisions and as a result active safety systems are investigated.

Model-based approaches have been proposed,[5,6] where the measured joint torques are compared with those determined by the robot model and deviations are spotted. These methods either require a sufficient model of the system or make use of noisy acceleration measurements that need filtering and cause delays. Improvements to overcome the problem of imprecise parameters of a dynamic model have been proposed based on parameter estimation with adaptive control[7] or on neural network

* Corresponding author. E-mail: dimeasf@mech.upatras.gr

https://doi.org/10.1017/S0263574714001143 Published online by Cambridge University Press

approximation.[8] By using the generalised momentum of the robot, De Luca *et al.*[9] developed an efficient collision detection method that is applicable in manipulators with rigid and elastic links. Although this method uses decoupled properties of the manipulator, along with the rest model-based methods, it requires the explicit dynamic model which is not always available.

Fault detection methods have been proven sufficient in rotating machinery, automation or aircraft systems and can also be used in robotics. These methods are based on the following principle: small changes or faults in a structure can cause significant deviations in its dynamic behaviour. DeLuca and Mattone[10] detected faulty behaviours of the robot actuating system which implied collisions by using the joint position signal. Cho *et al.*[11] continued their research and proposed a collision detection methods based on the generalised momentum and by using joint torque measurements. Fuzzy logic has been proposed to construct fault detection observers that are robust to unknown disturbances, but they are not capable to perform under the high complexity of a robot manipulator.[12]

Statistical time series methods have also been used for fault detection in structures.[13] The main advantage of those methods lies on their ability to use the dynamic response signals to build a compact mathematical model representing the true dynamics of the structure. In this sense, parametric AutoRegressive (AR) or AutoRegressive Moving Average (ARMA) models are mainly used.[13] Nonetheless, this class of stationary parametric models are insuficient for the modelling of the dynamics of complex non-linear or time-dependent structures, like a robot manipulator. Instead, models capable of tracking the time-dependent dynamics of the structure are mandatory for this kind of application. In the Time-dependent ARMA (TARMA) class of models, the signal dynamics are modelled through a conventional ARMA model with parameters allowed to change with time.[14] The evolution of the time-dependent parameters is typically defined within deterministic paths, expanded on functional basis, in the so called Functional Series (FS-TARMA) model class. This class of models is very useful for the modelling of non-stationary dynamics following pre-determined trajectories. However, these models cannot predict the response signal for different system trajectories. A closely related class of time-dependent models are the Linear Parameter Varying (LPV-ARMA) model class, where the model parameters depend on the instant values of a scheduling variable which also varies with time[15] and determines the instantaneous system operating point.

In this paper, the collision detection problem of a robot with a human is investigated and two novel methods are proposed that do not require the explicit dynamic model of the robot. Instead, they approximate the coupled dynamics of the robot joints through a training procedure. With these methods, the occurrence and the magnitude of a collision on the link that collided is detected very quickly and before the reaction forces become dangerous for the human. More specifically, the first method consists of a fuzzy inference system (FIS) on each joint of the robot that approximates the collision torque using as inputs the position error, the velocity and the measured joint torque. The second method consists of modelling the joint torques of the manipulator under normal operating conditions (no collision) by means of LPV-ARMA models with parameters depending on the measured joint velocity and the measured joint torque.

In Section 2 the collision detection problem is analysed and the proposed methods are described. The performance of the proposed schemes is evaluated in a collision test-bed using two joints of a KUKA LWR robot in Section 3.

2. The Proposed Collision Detection Methods

In the development of a collision detection method, it is necessary to detect the external force as soon as possible, and before it builds up to the level that can harm humans or the robot itself. Conventional methods calculate the collision force numerically from the explicit dynamic model of the manipulator using proprioceptive (position, torque) and exteroceptive (acceleration) sensors on the robot. The dynamics of a robot arm that interacts physically with the environment during a collision include the contact forces applied to the colliding link that are translated to joint torques. The dynamics of an n-link robot during a collision can be expressed as:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \tau_{\mathbf{tot}} = \tau_{\mathbf{mot}} - \tau_{\mathbf{ext}} \tag{1}$$

where $\mathbf{q}$, $\dot{\mathbf{q}}$, $\ddot{\mathbf{q}}$ are the manipulator's joint positions and their corresponding time derivatives, $\mathbf{M}(\mathbf{q})$ is the inertia matrix, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ is a matrix containing the Coriolis and centrifugal terms, $\mathbf{g}(\mathbf{q})$ is the gravity vector, $\tau_{\mathbf{tot}}$ is the total torque vector measured on the joints, $\tau_{\mathbf{mot}}$ is the actuator torque vector and $\tau_{\mathbf{ext}}$

https://doi.org/10.1017/S0263574714001143 Published online by Cambridge University Press

is the external torque vector from the collision. Notice that under normal operating conditions $\tau_{\mathbf{tot}}$ equals $\tau_{\mathbf{mot}}$ (no collision). Otherwise, during a collision, the external torques $\tau_{\mathbf{ext}}$ can be calculated as:

$$\tau_{\mathbf{ext}} = \tau_{\mathbf{tot}} - \tau_{\mathbf{mot}} \tag{2}$$

where $\tau_{\mathbf{mot}}$ is the expected motor torque that is calculated from the dynamic model (1) in the absence of $\tau_{\mathbf{ext}}$. Assuming that the collision occurs on link k of an n degree-of-freedom manipulator, then the collision torque is related to the collision force F_{ext} as follows:

$$\tau_{\mathbf{ext}} = \mathbf{J_k}^T \mathbf{F_{ext}} \quad \tau_{\mathbf{ext}} = \left[\tau_{ext}^1 \; \tau_{ext}^2 \ldots \tau_{ext}^k\right] \tag{3}$$

where $\mathbf{J_k}^T$ is the $6 \times k$ Jacobian matrix that relates the velocity of the collision point with the joint velocity and $\tau_{\mathbf{ext}}$ is the vector of the external joint torques up to joint k. The external joint torques $[\tau_{ext}^{k+1} \ldots \tau_{ext}^n]$ after the k^{th} link are negligible. Although this appears to be a straightforward technique, in order to be reliable, it requires an accurate dynamic model of the system, which in practice is not easy to obtain. Parameters like the weight and inertia of each link and the friction and hysteresis effects in the joints are very difficult to model and identify their values. As a result, the external torques estimated with this class of methods are not very accurate.

Alternatively, it is possible to train an implicit model to produce an estimate of the manipulators behaviour under normal operating conditions, whose output can be compared with actual observations obtained from the manipulator, in order to detect whether or not there have been changes in the dynamics. In this sense, a change in the dynamics would indicate a possible collision. The signals used for training and evaluation of the provided collision detection systems should be available from the proprioceptive sensors of the robots in order to reduce the dependency from external sensors during operation. In collaborative robots apart from position encoders, joint torque sensors are also available. The requirement is that these signals must include substantial information that allow identification of the external torques $\tau_{\mathbf{ext}}$. At the same time, the number of signals should be kept as small as possible to yield a computationally efficient solution. Under those considerations, the signals selected to build the models in each joint include:

1. The position errors $\mathbf{q_e}$ of the position control system,
2. The total joint torques $\tau_{\mathbf{tot}}$ measured by the built-in sensor,
3. The commanded $\dot{\mathbf{q}}_{\mathbf{d}}$ and estimated joint velocities $\dot{\mathbf{q}}$ of the position control system. The later is derived from numeric differentiation of the measured joint position variables.

The position error and the total joint torque variables are selected because they are greatly influenced by the presence of a collision. However, large alterations in these variables are not always indicators of collision, as these changes can also be observed under normal operation, due to high inertial forces appearing on the links during changes of the velocity direction. Since this phenomenon is related to the joint velocity, the signal of commanded joint velocity $\dot{\mathbf{q}}_{\mathbf{d}}$ can also be used as an input for the training in order to distinguish the collision spikes. Although, the signals from the force and torque sensors include a small amount of electrical noise, no digital filters are being used in order to avoid delays.

Although the proposed methods are implemented on multiple joints of the robot, which is a coupled system, each collision detection system only uses the position or/and torque sensors of the joint that acts upon. As a result, the collision detection systems are decentralised. This choice reduces significantly the complexity, because otherwise each collision detection system should also include a quite larger number of inputs proportional to the number of joint that would result in a computationally inefficient overall method. On motion control strategies[16] the highly coupled non-linear dynamics of a robot are transformed into linear decoupled systems with the feedback linearisation technique which is a straightforward problem. In an effort to design uncoupled collision detection systems that rely on coupled variables ($\mathbf{q_e}$, $\tau_{\mathbf{tot}}$) the training process is selected accordingly in order to excite the coupled dynamics and include the coupling information on each of the uncoupled collision detection systems.That can be achieved with simultaneous excitation of all joints or of the joints that have coupled dynamics rather than identification of each joint at a time. In this paper, two joints of a robot are investigated with collision detection for proof of concept. The main advantage of uncoupled

https://doi.org/10.1017/S0263574714001143 Published online by Cambridge University Press

systems is their ability to be implemented in more joints with linear increase of computations rather than exponential.

In the following sections the two proposed approaches for collision detection are presented: the first one, is a fuzzy inference system that creates a mapping from the position error, total torque and joint velocity, to an estimate of the external torque; and the second one, is a Linear Parameter Varying AR model that predicts the total joint torque, based on previous measurements of the total measured torque and the current joint velocity.

2.1. Fuzzy inference system detection method

Fuzzy modelling is a powerful tool for non-linear function approximation.[17] A fuzzy system can be trained to identify complex functions by estimating the fuzzy parameters and rules. Fuzzy modelling is simple, flexible and can approximate the behaviour of complex systems even if inexact and incomplete data are available. For the problem of external torque detection, a fuzzy system in each joint is trained to approximate the input-output association between the external torque and the consequence on the robot joint that is measured by the proprioceptive sensors.

A Tagaki-Sugeno functional fuzzy system is selected, which is computationally efficient and can work well with non-linear problems. Each of the fuzzy systems has the following form:

$$R_i : IF\ q_e[t]\ is\ A^h_{q_e}\ and\ IF\ \dot{q}_d[t]\ is\ A^l_{\dot{q}_d} and\ IF\ \tau_{tot}[t]\ is\ A^m_{\tau_{tot}}\ THEN\ \tau'_{ext}[t] = g_i \tag{4}$$

where R_i represents the i^{th} rule, $q_e[t]$, $\dot{q}_d[t]$, $\tau_{tot}[t]$ are the linguistic variables that represent the inputs for the position error, commanded velocity and total measured torque at time t for joint k respectively (indexes are omitted), A are the linguistic values and τ'_{ext} is the estimated output of the fuzzy system for the external torque that is a linear function of g_i and has the following form:

$$g_i = a_{i,0} + a_{i,1}q_e + a_{i,2}\dot{q} + a_{i,3}\tau_{tot} \tag{5}$$

where $a_{i,j}$ are real number parameters of each rule.

Given the function

$$\tau_{ext} : X \to Y$$

where $X \subset \Re^3$ and $Y \subset \Re$, we wish to construct a fuzzy system

$$\tau'_{ext} : X \to Y$$

so that

$$\tau'_{ext}(x) = \tau_{ext}(x) + e(x) \tag{6}$$

for all $x = [q_e\ \dot{q}_d\ \tau_{tot}]^\intercal \in X$, where the approximation error $e(x)$ should be as small as possible.

Each fuzzy system acts on a single joint and their total number equals to the number of joints that are implemented with collision detection. In this paper, collision detection on two joints is investigated for proof of concept and consequently two fuzzy systems are developed. As it is discussed in next sections, the method can be implemented in more joints however further investigation has to be conducted in order to acquire the coupling between the robot link dynamics. The fuzzy system on each joint, is a multi input - single output system that has three inputs $q_e, \dot{q}, \tau_{tot}$ and one output τ'_{ext}. For the training procedure the information for the actual external torque τ^i_{ext} is provided by the external force sensor (see Fig. 1). The actual collision force F_{ext} is measured by an external force sensor and is translated into joint torques $\tau_{\mathbf{ext}}$ using Eq. (3).

Considering the amount of data that have to be processed, a fuzzy c-means clustering technique is selected to divide the data into classes and reduce the number of the membership functions that results in a computationally efficient overall system. The clustering process determines the number of rules and membership functions for the premise and consequence. The training of each FIS for tuning the consequence function parameters, is accomplished with the least-squares method.

The operating principal of the fuzzy system as a collision detector is that the fuzzy is trained to create a mapping between the collision torques $\tau_{\mathbf{ext}}$ and the resulted disturbances that can be

https://doi.org/10.1017/S0263574714001143 Published online by Cambridge University Press

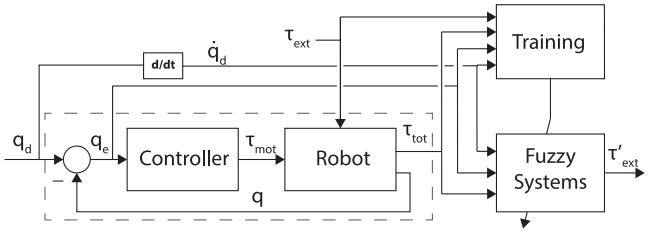


Fig. 1. Diagram of the fuzzy inference system for the training process. In normal operation for collision detection the "Training" block is not active. The dashed rectangle represents the internal robot controller.

measured from the proprioceptive sensors. After the mapping is complete, the inverse function can be accomplished and estimate the external torques when a collision occurs without using the external force sensor.

2.2. Linear parameter varying autoregressive detection method

A statistical time series method is proposed for collision detection that identifies the robot dynamics in collision-free operation and detects collisions as faults in the structure of the robot. A Linear Parameter Varying AR (LPV-AR) model is used with parameters depending on the measured joint velocity. Unlike the fuzzy system, the LPV-AR method models the total torque in each joint and does not require an external force sensor. In this form, the LPV-AR model is capable to describe the non-linear time-dependent nature of the total joint torque with the joint velocity. The total torque in each joint is modelled as follows:

$$\tau_{tot}[t] = -\sum_{i=1}^{n_a} a_i(\dot{q}[t]) \cdot \tau_{tot}[t-i] + w[t], \quad w[t] \sim NID\,(0, \sigma_w^2(\dot{q}[t])) \tag{7}$$

where $t = 1, 2, \ldots, N$ is the normalized discrete time, $\tau_{tot}[t]$ is the joint total torque signal to be modelled, $a_i(\dot{q}[t])$ is the scheduling variable, n_a the model order, and $w[t]$ is a Normally Independently Distributed (NID) random process with mean zero and variance $\sigma_w^2(\dot{q}[t])$. The proposed method differs from conventional AR models, because the model parameters $a_i(\dot{q})$ are a function of the external scheduling variable $\dot{q}$, which is the joint velocity for the current application.

The LPV-AR model is completed by formulating explicitly the dependency of the model parameters $a_i(\dot{q}[t])$ and innovations variance $\sigma_w^2(\dot{q}[t])$ on the instantaneous value of the scheduling variable $\dot{q}$, in the form of a functional basis expansion

$$a_i(\dot{q}[t]) = \sum_{k=0}^{p_a} a_{i,k} G_{b_{a(k)}}(\dot{q}[t]), \quad \sigma_w^2(\dot{q}[t]) = \sum_{k=0}^{p_s} s_k G_{b_{s(k)}}(\dot{q}[t]) \tag{8}$$

where $\dot{q} \in \mathbb{R}(-1, 1)$ is the normalized velocity (where ± 1 indicates the maximum value during normal operation), $a_{i,k}$ and s_k are the AR and innovations variance projection parameters, with corresponding orders p_a and p_s. The functional basis expansion of the AR parameters and innovations variance is made based on a properly selected functional subspace

$$\mathcal{F}_{AR} = \left\{G_{b_{a(1)}}(\dot{q}), G_{b_{a(2)}}(\dot{q}), \ldots, G_{b_{a(p_a)}}(\dot{q})\right\}, \quad \mathcal{F}_{\sigma_w^2} = \left\{G_{b_{s(1)}}(\dot{q}), G_{b_{s(2)}}(\dot{q}), \ldots, G_{b_{s(p_s)}}(\dot{q})\right\} \tag{9}$$

where $\mathcal{F}$ designates a functional space of the indicated quantity and $G_k(\dot{q})$ a set of orthogonal basis functions selected from a suitable family. Within this application, the family of Chebyshev polynomials are used, which are defined as follows

$$G_0(\dot{q}) = 1, \quad G_1(\dot{q}) = \dot{q}, \quad G_{k+1}(\dot{q}) = 2\dot{q}G_k(\dot{q}) - G_{k-1}(\dot{q}) \tag{10}$$

The identification of the LPV-AR model consists on the selection of the model and basis orders, n_a, p_a and p_s, respectively, and the estimation of the projection parameters $a_{i,k}$ and s_k. The estimation

https://doi.org/10.1017/S0263574714001143 Published online by Cambridge University Press

of the projection parameters, based on a single scheduling trajectory $\dot{q}[t]$, is carried out by Maximum Likelihood method, which consists on the minimization of the negative log-likelihood of the model, defined as follows:

$$\hat{\theta}_{ML} = \arg\min_{\theta} \left(-\ln \mathcal{L}(\theta|\tau_{tot}[t], \dot{q}[t])\right) = \arg\min_{\theta} \sum_{t=1}^{N} \left(\ln \sigma_w^2(\dot{q}[t]) + \frac{e^2[t]}{\sigma_w^2(\dot{q}[t])}\right)$$

$$e[t] = \tau_{ext}[t] + \sum_{i=1}^{n_a} a_i(\dot{q}[t]) \cdot \tau_{ext}[t-i] \tag{11}$$

where $\hat{\theta}_{ML} = \left[a_{1,0}\ a_{2,0} \cdots a_{n_a,p_a}\ s_1 \cdots s_{p_s}\right]^T$ is the Maximum Likelihood estimated parameter vector and $e[t]$ is the one-step ahead prediction error of the LPV-AR model. This optimization problem is non-quadratic and must be solved by non-linear optimization techniques, which require accurate initial values in order to avoid potential local minima. To do so, initial values are found using other techniques, like the Multi-Stage Weighted Least Squares method. The model order selection is carried out heuristically by examining models with different orders and selecting the one that provides the best performance with the least number of parameters. To do so, performance measures such as the Residual Sum of Squares (RSS) or the Bayesian Information Criterion (BIC) are used.

The collision detection is performed by assessing the value of the prediction error of the LPV-AR model. The idea behind the method consists on assuming that if there has been a change in dynamic system, the estimated LPV-AR model will not predict accurately the torque in the joints, and as a result, the prediction error will increase. In that sense, the following hypothesis test is developed[18]

$$\begin{aligned} H_o &: \sigma_{w(u)}^2[t] \leq \sigma_w^2(\dot{q}[t]), && \text{null hypothesis - no collision} \\ H_1 &: \sigma_{w(u)}^2[t] > \sigma_w^2(\dot{q}[t]), && \text{alternative hypothesis - collision} \end{aligned} \tag{12}$$

where $\dot{q}[t]$ is the joint velocity of the scheduling variable during the test, $\sigma_{w(u)}^2(\dot{q}[t])$ is the estimated innovations variance, and $\sigma_w^2(\dot{q}[t])$ is the innovations variance predicted by the model, both of them defined as follows

$$\sigma_{w(u)}^2[t] = \frac{1}{N_w} \sum_{j=1}^{N_w} e_u^2[t-j], \quad \sigma_w^2(\dot{q}[t]) = \sum_{k=0}^{p_s} s_k G_{b_{s(k)}}(\dot{q}[t]) \tag{13}$$

and where $e_u[t]$ are the prediction errors obtained by the model based on the test values of the joint torque $\tau_{tot(u)}[t]$ and corresponding velocity $\dot{q}[t]$. Being the prediction errors $e_u[t]$ Gaussian distributed, and assuming that the LPV-AR model predicted innovations variance is deterministic, then the following statistic[19]

$$\chi_{N_w}^2[t] = \frac{N_w \cdot \sigma_{w(u)}^2[t]}{\sigma_w^2(\dot{q}[t])} \tag{14}$$

follows a chi-square distribution with N_w degrees of freedom. Therefore, the following collision detection test is defined at the α (false alarm) risk level

$$\begin{aligned} \chi_{N_w}^2[t] \leq \chi_{N_w}^2(1-\alpha) \qquad & \text{Accept } H_o \\ \text{Otherwise} \qquad & \text{Accept } H_1 \end{aligned} \tag{15}$$

where $\chi_{N_w}^2(1-\alpha)$ is the chi square probability distribution function at the $1-\alpha$ probability level.

Summarizing, the LPV-AR based collision detection method is performed as follows: (i) In the training phase, an LPV-AR model is identified for each of the two joint of the manipulator that are investigated in collision-free operating conditions by providing the total measured torques and the measured joint velocities. The model identification includes selection of the AR and basis orders,

https://doi.org/10.1017/S0263574714001143 Published online by Cambridge University Press

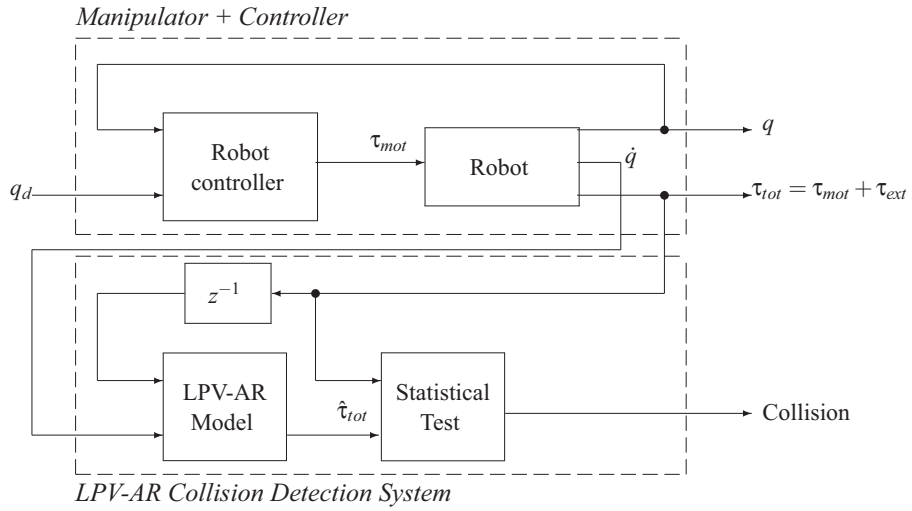


Fig. 2. Graphical description of the collision detection system hole configuration of the robot manipulator with the LPV-AR collision detection method.

plus estimation of the projection parameters. (ii) In the test phase, the estimated LPV-AR models are used to predict the corresponding total joint torques using Eqs. (7) and (8), based on the estimated projection parameters and the measured joint torques at $t-1,\ldots,t-n_a$. Afterwards, based on the prediction error $e[t]$, proceed with the test in Eq. (15). Figure 2 shows a graphical description of the collision detection system for a single joint with the trained LPV-AR method.

3. Experimental Validation

The evaluation of the collision detection methods is conducted using data from a KUKA LWR IV manipulator commanded to perform planar movements in the 4th and 6th joint which are mentioned as joint 1 and 2 respectively (Fig. 3). Both of the proposed methods are implemented on a desktop PC communicating with the robot controller at a frequency of $f_s = 500Hz$. The external force sensor is connected to the desktop PC and measures the actual force at the same frequency f_s. The training of the proposed fuzzy and LPV-AR models requires the provisioning of sufficient input-output associations that can describe accurately the manipulator dynamics. So, in order to assure that the models provide minimum error for a large span of operational conditions, the commanded trajectory should excite as much of the dynamics of the system as possible. With this purpose in mind, the selected excitation

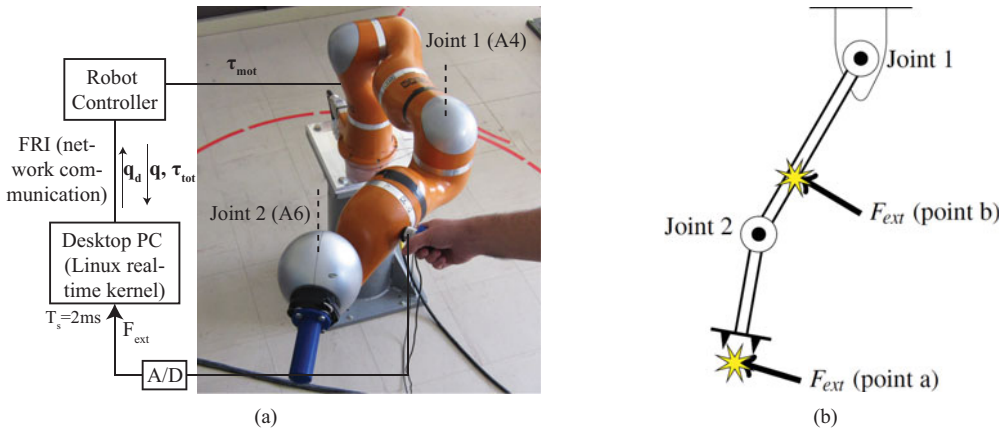


Fig. 3. a) Experimental setup with the KUKA LWR moving joints A4 & A6 and an external force sensor held by the operator. b) The external forces F_{ext} are applied at predefined locations (one collision point each time) for training and are translated into joint torques τ_{ext} using Eq. (3). The collision detection systems estimate that external torques $\tau'_{\mathbf{ext}}$ and the link that collided.

https://doi.org/10.1017/S0263574714001143 Published online by Cambridge University Press

signal $\mathbf{q_d}$ is designed to follow a sinusoidal profile with linearly increasing frequency between 0.05Hz and 0.3Hz at a slightly different rate for each joint. This frequency produces angular velocities up to 2 rad/s and linear velocity at the end-effector up to 1.5 m/s. These velocities approach the usual operating values for industrial robots and far exceed the velocity of 0.2 m/s that is the current maximum value allowed for human-robot interaction.[20]

The training data is divided in two sets: in the first set the robot joints perform the commanded sinusoidal motion without any external forces applied to the robot body, and in the second set the same motion is performed, with the user provoking random collisions with his hand. The forces are applied in predefined positions, perpendicular to the motion of the corresponding link, and are measured with an external force sensor held by the operator. The collision forces are then translated into torques in each joint according to the configuration of the robot in order to be used for the evaluation of the methods. During the experiments the robot is commanded to move with position control mode active and no reaction strategy implemented. The deliberately applied collisions are performed momentarily on random time, and for safety reasons the experiments are conducted under supervision and with enabling hand-held switches.

3.1. Fuzzy system training

Using only the data that include collisions, the fuzzy systems are created and trained as it was described in Section 2.1. A total number of 23.000 input-output pairs are used for each of the two system, collected at a frequency f_s. The input membership functions calculated for each fuzzy system from the c-means clustering procedure are shown in Fig. 4. The clusters between the two joints share some similarities on the corresponding inputs but they are not identical, since each system approximates different dynamics. The membership functions of inputs q_e and $\dot{q}$ present a symmetry approximately around zero because of the sinusoid signal exciting the joint dynamics in both directions. However, in the membership functions for τ_{tot} in joint 1 there is no symmetry since the training data contain collision occurrences in certain directions. This affects only the accuracy of the estimation and not the detection because the fuzzy has the ability to generalise. By providing more collision data to the training process, this abnormality can be avoided. In comparison with the

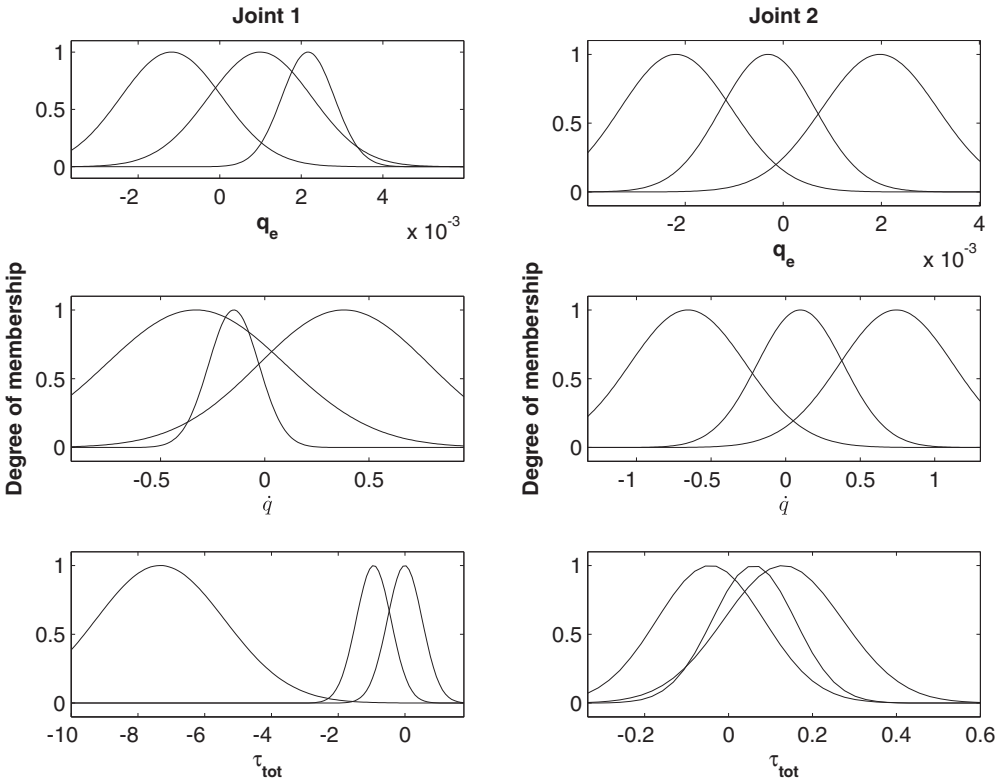


Fig. 4. Input membership functions of the trained fuzzy systems for two joints.

https://doi.org/10.1017/S0263574714001143 Published online by Cambridge University Press

τ_{tot} in joint 2 it can be seen that joint 1 is affected much more because of the higher level-arm. The parameters of the output functions given by Eq. (5) are trained with the least squares method and the method converges with a single epoch of the training data.

After the training is completed, the fuzzy system is evaluated with the same training data in order to get an insight about the approximation. The absolute error between the collision torque $\tau_{\mathbf{ext}}$ measured by the external sensor and the estimated output $\tau'_{\mathbf{ext}}$ of the fuzzy system are illustrated in Fig. 5(a). The average errors for joints 1 and 2, are 0.03 Nm and 0.01 Nm respectively. The corresponding maximum error values are 1.62 Nm and 0.47 Nm, that are observed during collisions. These large errors appear because of the clustering process of the fuzzy system, which reduces the number of rules and the computational complexity of the system. The accuracy of the evaluation of the collision force could be improved by selecting even partitioning of membership functions.

The collision threshold value $\tau_{\mathbf{th}}$ for each joint is calculated as the maximum approximation error of contact free motion. By providing the first dataset without any collisions to the fuzzy systems, the resulted threshold values are $\tau_{\mathbf{th}} = \begin{bmatrix} 0.078 & 0.026 \end{bmatrix} Nm$ for joints 1 and 2 respectively.

3.2. LPV-AR model training

The LPV-AR models are trained to estimate the total torque on each joint from the measured joint velocity signals. As presented in Section 2.2, the model identification requires the selection of the model and basis orders, and the estimation of the parameter vector $\theta_{\mathbf{ML}}$. The parameter vector is estimated using the Multi-Stage Weighted Least Squares methods, with the following conditions: maximum number of iterations 1000, tolerance of the objective function change 10^{-8}, tolerance of the norm of the change of the parameter vector 10^{-8}.

The process of model structure selection (model and basis order selection) for the signal in normal operating conditions consists on two steps: (i) LPV-AR models are estimated for increasing AR orders in the range $n_a = \{1, \ldots, 20\}$ with $p_a = p_s = 5$, the model order is selected as the one minimizing the BIC; (ii) LPV-AR models are estimated with the value of n_a found in the previous step, and with increasing values of p_a and p_s, where $p_a = p_s = \{1, \ldots, 20\}$. After the model identification process the following structures are found:

- For the first joint (joint 4): LPV-AR with $n_a = 12$, $p_a = p_s = 15$.
- For the second joint (joint 6): LPV-AR with $n_a = 15$, $p_a = p_s = 15$.

Figure 5b shows the estimation error of the model for the data-set of contact-free motion. Figure 6 shows the obtained values of the collision detection statistic $\chi^2_{N_w}[t]$ for the model obtained in the first joint using a window length $N_w = 100$ samples, and a threshold $\chi^2_{N_w}(1-\alpha) = 250$ corresponding

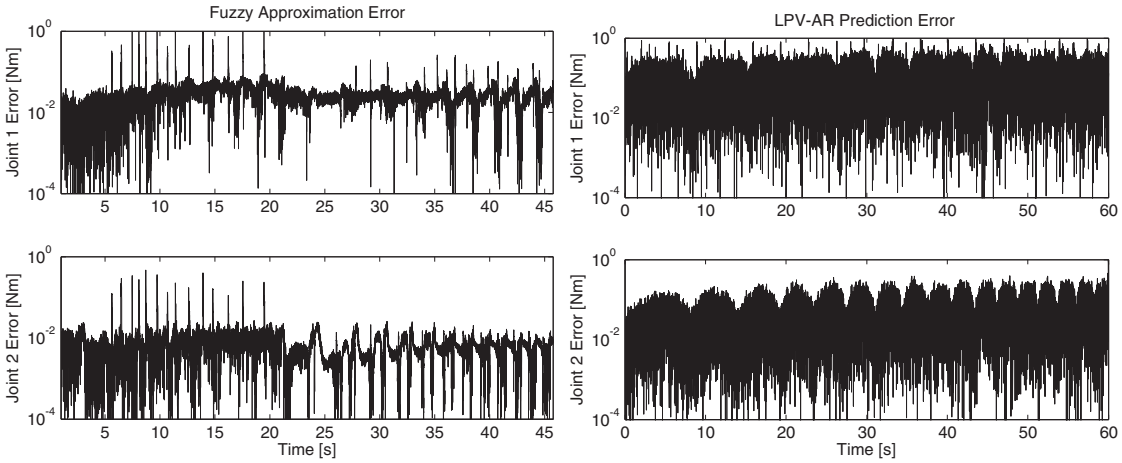


Fig. 5. Error analysis at each one of the two joints on the training data. (a) Approximation error of the FIS on the training data with collision occurrences. (b) Prediction error obtained by the LPV-AR(12)$_{15}$ and LPV-AR(15)$_{15}$ models of the joint torques. Top: Error for joint 1. Bottom: Error for joint 2.

https://doi.org/10.1017/S0263574714001143 Published online by Cambridge University Press

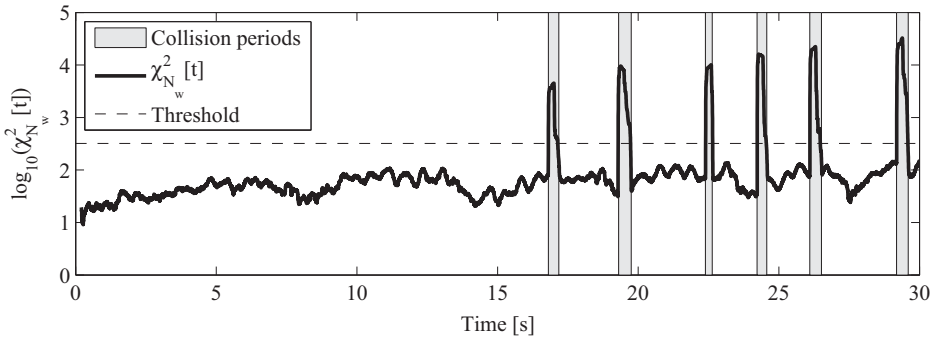


Fig. 6. Values of the collision detection statistic $\chi^2_{N_w}[t]$ for the LPV-AR model obtained in the first joint, during a test with collisions in the periods indicated by shaded areas.

to $\alpha = 0.01$. It can be noticed that the collision statistic value clearly identifies the collisions and provides early alerts.

3.3. Evaluation and comparison

The proposed methods are evaluated in different collision scenarios, generated while the robot performs joint motion profiles different than the ones used for training, in order to test the generalization properties of the methods. During the tests, the collisions are deliberately produced when the operator touches the manipulator in one of the links with his hand. The actual collision forces are measured with a force sensor held by the operator.

The evaluation of the collision detection is not implemented to the robot controller, since we are interested on identifying collisions and we are not investigating a reaction strategy. Consequently, we collect the data from the evaluation velocity profiles experiment and we provide them to the trained systems. It should be noted that both methods can be easily implemented in the robot controller for real-time collision detection. In order to identify a collision, a threshold value has to be exceeded, that is selected equal to the maximum training error from contact-free motion training. The detection time is calculated as the elapsed time from the detection of a collision by the external force sensor to the moment when the estimated force exceeds the threshold. To compare the proposed trained methods with an analytical model-based approach, the estimation of the external torque was obtained by the build-in model of the KUKA robot that uses a state feedback controller.[21]

The proposed trained methods are first evaluated with a steady velocity profiles equal to 1.0rad/s on both joints with two cases of collisions (see Fig. 3(b)). In Fig. 7 is illustrated how each one of the methods approximates the external torques during a collision on the end-effector (point a), expecting to affect both joints, and in between the two joints (point b), in order to investigate the coupling between them. The evaluated collision torques from the estimation of the fuzzy systems (Fig. 7(a)) and the LPV-AR models (Fig. 7(b)) are illustrated along with the actual contact torques measured by the external force sensor and a model-based estimation of the external torque as it is provided from the robot controller.

The contact torques estimated by the FIS are very close to the actual measured torques, which demonstrates that the FIS model is capable of modelling with very good accuracy the collision torques. In comparison with the results provided by the analytical model, the FIS approximation successfully compensates for any unmodelled parameters which is necessary for choosing the reaction strategy. When the collision occurs between the two joints, the external torque evaluated on the second joint correctly remains below the threshold value because the collision force does not apply wrench to it. On the other hand, the estimates obtained with the LPV-AR model are less accurate, meaning that they have larger threshold values, but appear to be more sensitive to the collision. Table I provides a summary of the thresholds used for collision detection and the detection time provided by each one of the methods. The LPV-AR collision detection method provides the earlier detection times, followed by the FIS method. The lower threshold of the FIS method is product of a more accurate approximation of the external torques and consequently of the training procedure. The accuracy of

https://doi.org/10.1017/S0263574714001143 Published online by Cambridge University Press

Table I. Comparison of the collision detection thresholds and detection times obtained for the proposed collision detection methods on each joint and constant velocity of 1.0rad/s.

	Threshold [Nm]		Detection time [ms]	
Detection method	Joint 1	Joint 2	Joint 1	Joint 2
FIS method	0.078	0.026	10	5
LPV-AR method	1.000	1.000	4	4
Model-based method	1.040	0.240	16	10

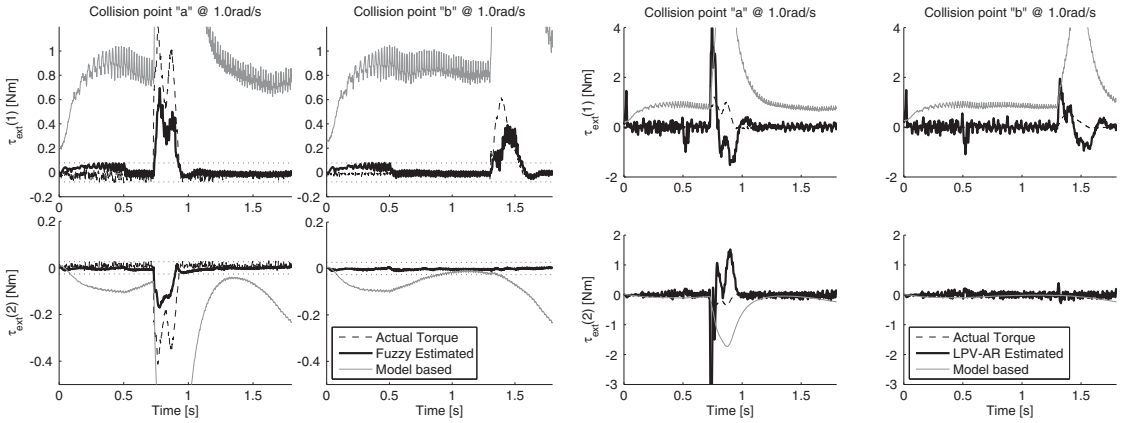


Fig. 7. The actual collision torque compared to the fuzzy system and model-based method (Fig. a) and with the time series method (Fig. b) for constant velocity 1.0rad/s on both joints. Left columns represent collisions at the end effector (point a) and the right columns collision between the two joints (point b). Each row on the figure represents the corresponding joint.

the evaluation can be used to select an appropriate reaction strategy and the early detection time for triggering the reaction as soon as possible.

To confirm the validity of the proposed methods under a wide range of operating conditions and acquire a performance measure, another 61 trials of collisions are evaluated involving various magnitudes, points of collision, directions and joint velocities. Table II provides the performance in terms of: the number of collisions accurately detected (ACC); the number of false negatives (FN), which is the number of collisions not detected by the methods; and the number of false positives (FP), which is the number of collision alerts provided by the methods when there is not an actual collision. The performance of the methods are separated into two categories according to the link of the collision. It is demonstrated that the collisions occurred at the end-effector are acknowledged by both of the proposed systems with approximately the same success percentage (fuzzy: 85%, LPV-AR: 95%). The collisions occurred on the link between the two joints are acknowledged successfully both with the fuzzy and the LPV-AR method on joint 1 and not by joint 2 but with a lower success rate (fuzzy: 50%, LPV-AR: 35%). That happens because the collision force is applied closer to joint 1 and has a smaller level-arm distance that results in smaller torque. A number of false positive collisions is presented with the LPV-AR method (11%), mainly because less variables are used for the estimation than the fuzzy system. As a result, the LPV-AR method is more sensitive to external disturbances and unmodelled parameters unlike the fuzzy system that does not evaluate any false positive collisions and is more robust.

Even though the collision detection systems in each joint are not coupled, both of the proposed systems can successfully estimate the coupled dynamics. That can be clearly seen in Tab. II, by the number of the collisions acknowledged on joint 2 during the collision between the two-joints. With both methods this number is zero and it is implied that the collision detection systems in joint 2

https://doi.org/10.1017/S0263574714001143 Published online by Cambridge University Press

Table II. Summary of the performance of the collision detection methods obtained for different collision scenarios and angular speeds of the joints.

Collision scenario		NC	FIS method			LPV-AR method		
			ACC	FN	FP	ACC	FN	FP
Collision at the	Joint 1	27	23	4	0	23	4	5
end-effector	Joint 2	27	23	4	0	27	0	0
Collision between	Joint 1	34	17	17	0	12	22	4
the two joints	Joint 2	0	0	0	0	0	0	1
Average Percentage			72%	28%	0%	70%	30%	11%

Conventions: NC: number of collisions, ACC: Number of collisions Accurately detected, FN: Number of False Negatives, FP: Number of False Positives.

are not affected by the disturbances occurred in previous joints (towards base). Since the proposed methods can approximate the coupled dynamics of a robot using uncoupled systems on each joint, this collision detection scheme can be implemented in more than two joints of a robot, but with further consideration to the training procedure in order to include substantial information for the robot dynamics and the coupling between the joints.

4. Conclusions

In this paper, two novel methods for robot collision detection are proposed based on fuzzy logic and times series modelling. Unlike previous methods that require the explicit robot model and suffer from unmodelled parameters or low estimating accuracy, the proposed methods are trained to the robot dynamics and can estimate the occurrence, the link that collided and in some extent the magnitude of a collision with a high percentage.

The proposed methods are trained using data collected from the proprioceptive sensors of a KUKA LWR robot on a two joint, proof of concept experiment. The fuzzy inference systems are designed to evaluate directly the collision joint torques by the disturbances occurred in the measurements. To enable better sensitivity of the fuzzy systems on detecting collisions, the training data are derived from contacts of the robot body by a human hand using an additional external force sensor. In that way, a richer signal is obtained and the train error becomes very small. On the other hand, the proposed statistical time series method (LPV-AR) is trained only with collision-free data and estimates the expected total torque on each of the investigated joints. Collisions are detected statistically by observing large deviations between the actual and expected torque. This method is more efficient since it does not require an external sensor.

The evaluation of the proposed methods shows that both of them can detect collisions very fast and they have inherited information about the coupled robot dynamics by the training process even though the systems in each joint are decoupled. This is succeeded by the appropriate selection of inputs and training trajectory for sufficient dynamic excitation. The fuzzy system is very accurate on the magnitude of the estimation and has the lowest threshold value. The LPV-AR method, although it has a larger threshold and overestimates the magnitude, it is more sensitive and thus faster on the detection time. The average percentage of acknowledged collisions is 71% and for the false negative collisions is 29%. The LPV-AR method being more sensitive, demonstrates false positive collisions that occur due to unmodelled parameters, in contrast with the fuzzy system which is more robust.

The results of this paper is a preliminary analysis that requires further investigation and experiments in order to implement a collision detection system considering motion of all joints of a robot. Moreover, the accurate estimation of the collision magnitude can be used to develop an appropriate collision reaction strategy that mitigates the reaction forces, which can also benefit a lot by having identified the link of the robot that collided. For future work, a collision detection scheme will be investigated that distinguishes between unexpected collisions and expected contacts of the robot with the environment (human or objects).

https://doi.org/10.1017/S0263574714001143 Published online by Cambridge University Press

References

1. C.-J. Wu, "A modeling method for collision detection and motion planning of robots," *Robotica* **11**, 217–226 5 (1993).
2. J. S. Choi, Y. Yoon, M. H. Choi and B. H. Lee, "Parameterized collision region for centralized motion planning of multiagents along specified paths," *Robotica* **29**, 1059–1073 (12 2011).
3. F. Flacco, T. Kroger, A. De Luca and O. Khatib, "A Depth Space Approach to Human-Robot Collision Avoidance," *Proceedings of the 2012 IEEE International Conference on Robotics and Automation*, IEEE (May 2012) pp. 338–345.
4. T. L. Lam, H. W. Yip, H. Qian and Y. Xu, "Collision Avoidance of Industrial Robot Arms Using an Invisible Sensitive Skin," *Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE (Oct. 2012) pp. 4542–4543.
5. W. Chen, Y. Sun and Y. Huang, " A Collision Detection System for an Assistive Robotic," **In**: *Communications in Computer and Information Science* (K. Li, X. Li, S. Ma and G. W. Irwin, eds.) (Springer Berlin Heidelberg, 2010) pp. 117–123.
6. Y. Yamada, Y. Hirasawa, S. Huang, Y. Umetani and K. Suita, "Human-robot contact in the safeguarding space," *IEEE/ASME Trans. Mechatronics* **2**(4), 230–236 (1997).
7. T. Matsumoto and K. Kosuge, "Collision Detection of Manipulator Based on Adaptive Control Law," *Proceedings of the 2001 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (Cat. No.01TH8556)*, volume 1. IEEE (2001) pp. 177–182.
8. S. Lu, J. H. Chung and S. A. Velinsky, "Human-robot interaction detection: a wrist and base force/torque sensors approach," *Robotica* **24**(04), 419 (Feb. 2006).
9. A. De Luca, A. Albu-Schaffer, S. Haddadin and G. Hirzinger, "Collision Detection and Safe Reaction with the DLR-III Lightweight Manipulator Arm," *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems* (Oct. 2006) pp. 1623–1630.
10. A. De Luca and R. Mattone, "Sensorless Robot Collision Detection and Hybrid Force/Motion Control," *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. IEEE (2005) pp. 999–1004.
11. C.-N. Cho, J.-H. Kim, S.-D. Lee and J.-B. Song, "Collision detection and reaction on 7 DOF service robot arm using residual observer," *J. Mech. Sci. Technol.* **26**(4), 1197–1203 (Apr. 2012).
12. M. Bouattour, M. Chadli, M. Chaabane and A. Hajjaji, "Design of robust fault detection observer for Takagi-Sugeno models using the descriptor approach," *Int. J. Control Autom. Syst.* **9**(5), 973–979 (Oct. 2011).
13. J. S. Sakellariou and S. D. Fassois, "Vibration based fault detection and identification in an aircraft skeleton structure via a stochastic functional model based method," *Mech. Syst. Signal Process.* **22**(3), 557–573 (2008).
14. A. G. Poulimenos and S. D. Fassois, "Parametric time-domain methods for non-stationary random vibration modelling and analysis: A critical survey and comparison," *Mech. Syst. Signal Process.* **20**(4), 763–816 (2006).
15. L. Guiarre, D. Bauso, P. Falugi and B. Bamieh, "LPV model identification for gain scheduling control: An application to rotating stall and surge control problem," *Control Eng. Pract.* **14**, 351–361 (2006).
16. W. Chung, L.-C. Fu and S.-H. Hsu, "Motion Control," **In**: *Springer Handbook of Robotics* (B. Siciliano and O. Khatib, eds.) (Berlin Heidelberg, 2008) pp. 133–159.
17. K. M. Passino and S. Yurkovich, *Fuzzy Control* (Addison Wesley Publishing Company, California, 1997).
18. S. D. Fassois and F. P. Kopsaftopoulos, "Statistical Time Series Methods for Vibration Based Structural Health Monitoring," **In**: *New Trends in Structural Health Monitoring* (W. Ostachowicz and J. A. Güemes, eds.) (2013).
19. L. Ljung, *System Identification: Theory for the User*, 2nd edn. (Prentice Hall PTR, New Jersey, 1999).
20. International Standard. ISO 10182-1 INTERNATIONAL STANDARD. 2011 (2011).
21. A. Albu-Schäffer, S. Haddadin, Ch. Ott, A. Stemmer, T. Wimböck and G. Hirzinger, "The DLR lightweight robot: design and control concepts for robots in human environments," *Ind. Robot. Int. J.* **34**(5), 376–385 (Aug. 2007).

https://doi.org/10.1017/S0263574714001143 Published online by Cambridge University Press