Available online at www.sciencedirect.com
SciVerse ScienceDirect
Fuzzy Sets and Systems 235 (2014) 83–100
FUZZY sets and systems
www.elsevier.com/locate/fss

An interval-valued fuzzy controller for complex dynamical systems with application to a 3-PSP parallel robot

Hamid Reza Hassanzadeh[a,*], Mohammad-R. Akbarzadeh-T[a], Alireza Akbarzadeh[b], Amir Rezaei[b]

[a] *Department of Computer Engineering, Center of Excellence on Soft Computing and Intelligent Information Processing (SCIIP), Ferdowsi University of Mashhad, Mashhad, Iran*

[b] *Department of Mechanical Engineering, Center of Excellence on Soft Computing and Intelligent Information Processing (SCIIP), Ferdowsi University of Mashhad, Mashhad, Iran*

Received 23 September 2011; received in revised form 13 February 2013; accepted 15 February 2013

Available online 7 March 2013

Abstract

In this paper, we present a novel interval-valued fuzzy model-based controller for handling the effects of uncertainty in controlling a complex dynamical system. Theoretically, model-based controllers may be the ideal control mechanisms; however, they are highly sensitive to model uncertainties and lack robustness. These controllers are also computationally intensive, rendering them unusable for many real-world applications. In this work, we incorporate an interval fuzzy logic paradigm into a *computed-torque controller* for a 3-PSP parallel robot. This paradigm aims to handle the uncertainties in the robot model. The proposed approach benefits from algebraic operations on type-I fuzzy numbers to enhance its capability in dealing with uncertainty. The simulations prove the superiority of the proposed controller in the presence of uncertainty. Furthermore, comparisons with a competing type-I reduced controller as well as a PD controller show this superiority to be more pronounced especially when noise level is remarkably high. Moreover, the designed controller satisfies the computational complexity constraints for real-time implementation.

Keywords: Fuzzy control; Parallel manipulators; Robotics; Type-II fuzzy logic (T2FL); Interval-valued fuzzy logic; Computed-torque control method (CTC)

1. Introduction

The concept of *uncertainty* is posed in almost all complex systems including *parallel robots* (as an outstanding instance of dynamical robotics system). As suggested by the name, uncertainty is any missing information that is beyond the knowledge of human or is ignored due to excessive complexity; thus it must be handled in a way to minimize its side-effects through the control process.

While designing a type-I fuzzy logic system, we presume that we are almost certain about the fuzzy membership functions. This is not true in many cases. Consequently, T2FL as a more realistic approach for dealing with practical applications offers new interesting results. Contrasted with type-I fuzzy logic (T1FL), interval-valued fuzzy logic,

* Corresponding author. Tel.: +1 4047190810.
E-mail addresses: ha.hassanzadeh@ieee.org (H.R. Hassanzadeh), akbarzadeh@ieee.org (M.R. Akbarzadeh-T), ali_akbarzadeh@ferdowsi.um.ac.ir (A. Akbarzadeh), amirrezaei@stu.um.ac.ir (A. Rezaei).

0165-0114/$ - see front matter
http://dx.doi.org/10.1016/j.fss.2013.02.009

as a special case for T2FL, takes into account a higher level of uncertainty so that membership grades associated to T2 fuzzy variables are no longer crisp numbers but rather they are themselves T1 fuzzy intervals [1–6].

Parallel robots, on the other hand, are rather new sort of industrial and scientific benchmarks that are being used in many research and industrial communities. The most problematic issues that engineers and designers face when using such robots are the high computational burden of dynamical calculations as well as the uncertainty in the underlying robot structure and parameters. Maneuvering the aforementioned challenges properly may lead to tangible effects on the smoothness and amount of error during the control process.

During the past 2 years, type-II fuzzy sets and systems have been used in different control applications due to their relatively higher robustness compared to traditional fuzzy controllers [7–10]. The recent increase in the corresponding number of publications is a compelling evidence for the success of such approaches. In [11], the authors conducted a concise overview of different industrial applications in the emerging realm of type-II fuzzy logic. Castillo et al. [12] performed a review on methods utilized in design of interval type-II fuzzy controllers. Their research has been focused on basic reasons for optimizing type-II fuzzy controllers in different applications. In addition, a rather thorough survey on industrial applications of fuzzy control is presented in [13]. One of the areas that type-II fuzzy systems yield promising results is control engineering where in the presence of uncertainty they offer significant improvements compared to other pioneering control methods. This improvement is more pronounced when the system under control becomes more and more complex as is the case in this research. In [14–17] control of complex nonlinear dynamical systems in industrial fields that are difficult to model analytically is considered. Li et al. [18] presented a novel impulsive control approach based on an interval-valued fuzzy model for nonlinear systems. In [19], an inverse controller based on model control design strategy is introduced and implemented in a pH neutralization experimental setup. Sepulveda et al. [20] embedded a high speed interval-valued fuzzy controller for a plant into a field programmable gate arrays (FPGA) chip where being real-time is a requirement. A robust adaptive fuzzy tracking control for a class of nonlinear systems based on the sliding mode control is proposed in [21]. In [22] a servo control system with time-varying and nonlinear load conditions using type-II TSK fuzzy neural system is investigated. The authors use a type-II fuzzy neuro system for controlling of a DC motor by means of a hybrid algorithm. They also make use of clustering and gradient techniques to learn the parameters of the fuzzy neuro system.

In control, manufacturing and prediction, on the other hand, many articles have come out in just recent years, including but not limited to, T2 fuzzy systems for buck and boost DC–DC converters [23], coiler entry temperature prediction [24], assessment of power transformer failure risk [25], control of a smart washing machine [26], prediction of turbine inlet temperature of gas turbines [27] and sliding-mode controller for linear and nonlinear systems [28].

In robotics, especially parallel robots, as far as the state of the art matters, type-II fuzzy logic is still in its infancy which urges more effort in this important field of science. Some recent works in this area include, the use of type-II fuzzy logic for modeling a robot hand [29], control of an autonomous wheeled mobile robot [30], developing hybrid tracking controller for the mobile robot [31], navigation of mobile robots [32] and real-time control of mobile robots [33,34]. As for parallel robots, in [35] the authors devise a controller for leveling the cable-driven parallel mechanism. Moreover, Linda [7] introduces a fuzzy controller robust to uncertainty for Delta parallel robot.

In this paper we introduce a model-based control architecture based on type-II fuzzy logic for complex dynamical systems that deals with both uncertainty and real-time implementation requirements. To better represent the uncertainty, which flows in the control loop, we take advantage of fuzzy arithmetic to represent state variables in a more realistic way. We show that the designed controller exhibits smoother control surface than the type-I counterpart. To the best of our knowledge, our original study is the first of its kind in designing a type-II fuzzy model-based architecture to control a complex dynamical system.

The rest of this paper is organized as follows. In Section 2 few preliminary topics are covered briefly. Section 3 presents the architecture of the newly designed T2FL based controller as well as its type-reduced counterpart for further comparison. We then show the simulation results in Sections 4 through the course of different scenarios. Section 5 concludes the paper, and finally in Section 6 few suggestions for extending the research is given.

2. Theoretical framework

To enhance understanding of the proposed approach in later sections, this section presents a few important topics in brief, and provides references for further details.

2.1. Algebraic operations on fuzzy numbers

By definition, a fuzzy number is a fuzzy set which is *normal* and *convex* [36–38]. Based on the *extension principle* [4], any operation that is defined between two crisp numbers can be generalized to apply to two fuzzy numbers, as well. Accordingly, let $\odot$ denote a binary operation defined on $\mathcal{R}$ and F and G be two fuzzy numbers with membership functions f and g, respectively. Then the fuzzy number $H = F \odot G$ is defined, as follows:

$$H(z) = \sup_{z=v\odot w} \min(f(v), g(w)), \quad (v, w, z) \in \mathcal{R}^3. \tag{1}$$

Eq. (1) is the basis to define many useful operations on fuzzy numbers including multiplication of a fuzzy number by a crisp number, addition of two fuzzy numbers and consequently affine combination of several fuzzy numbers. As will be demonstrated in later sections, our proposed controller needs to compute the affine combination of some intervals (or general fuzzy numbers if general type-II fuzzy sets are used) which are the unprecisiated outputs of the interval-valued approximators. Therefore, as a corollary to (1), the following holds true for affine combination of interval sets [2].

Corollary 1. *Let $F_1, \ldots, F_n$ be n intervals with mid-points $m_1, \ldots, m_n$ and radius $s_1, \ldots, s_n$. For constants $\alpha_1, \ldots, \alpha_n$, β, $\sum_{i=1}^n \alpha_i \times F_i + \beta$ is an interval with mean $m = \sum_{i=1}^n \alpha_i m_i + \beta$ and spread $s = \sum_{i=1}^n |\alpha_i| s_i$.*

2.2. Type-II fuzzy logic

Compared to T1FL, T2FL is a more recent approach to fuzzy sets that has been successfully employed in many control applications where dealing with uncertainty is a key challenge [39,31,32,40,7,29,34]. Type-II fuzzy sets (T2FSs), which serve as the cornerstone for such systems, build on their predecessor by adding a second layer of fuzziness to them. Therefore, up to a restatement and sometimes extension to operations on traditional fuzzy sets, the core procedure to design a T2FLS remains the same. Accordingly, to communicate effectively through the rest of this paper, concise definitions of the key concepts are presented next and further references to external resources are provided where deemed necessary.

2.2.1. Type-II fuzzy sets

Consider an ordinary fuzzy set depicted in Fig. 1(a). An interval-valued fuzzy set can be acquired by shifting (not necessarily evenly) the points on the former membership function up and down. Fig. 1(b) depicts such a modified fuzzy set. For a point, x', in the domain of this fuzzy set there is not any attributed unique membership grade, rather, it corresponds to an interval as its membership grade. Furthermore, if we add a third dimension to account for the membership grades for each point on the aforementioned interval, what we get is called a general *type-II fuzzy set* (T2FS). These auxiliary membership functions are called *secondary* membership functions. Accordingly, an interval-valued fuzzy set is mathematically defined as follows:

$$\tilde{A} = \{(x, u), \mu_{\tilde{A}}(x, u) = 1 | \forall x \in X, \forall u \in J_x \subseteq [0, 1]\} \tag{2}$$

where J_x, the support of the secondary membership function, is an interval. In simpler terms, an interval-valued fuzzy set can be defined by a lower and an upper membership function and the area between them is called the *footprint of uncertainty* (FOU for short). The FOU corresponding to a general type-II fuzzy set can be found by projecting its 3D fuzzy set into the $x - u$ plane.

Due to significant reduction in the amount of computations that are required to compute the outputs of interval-valued based fuzzy systems, they have been readily used in a variety of control applications. Similarly, in this research we make use of interval-valued fuzzy sets, however, we should emphasize that the developed control architecture in this study does not restrict us to the use of this kind of fuzzy sets. In case general T2FSs used, we will have to manipulate fuzzy numbers instead of intervals in the last stage of our control system (cf. Section 3).

2.2.2. Type-II fuzzy logic systems

The knowledge to build a fuzzy logic system is itself uncertain. This uncertainty may arise from different sources [1] including the presence of noise and inexactness of measuring systems. Hence, the output of a control system should somehow reflect the available uncertainty in a proper way. This is one of the most meaningful reasons why ordinary

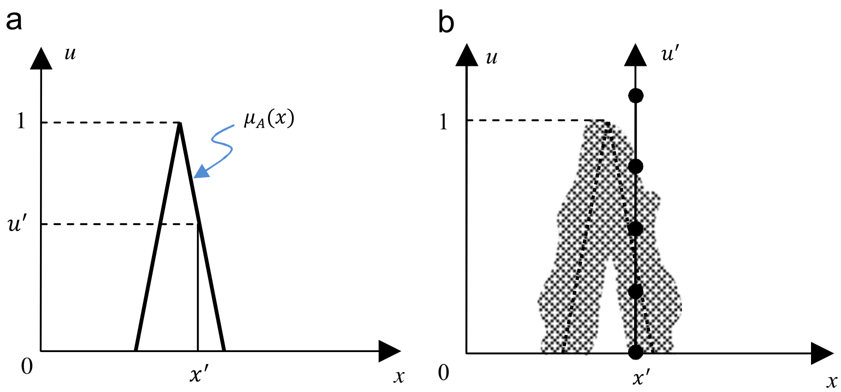

Fig. 1. Ordinary vs. perturbed fuzzy set. (a) An ordinary fuzzy set. (b) Perturbed fuzzy set.

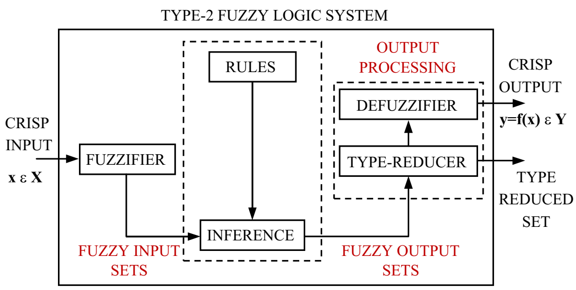

Fig. 2. Architecture of a T2FLS.

fuzzy logic systems (henceforth called T1FLSs) have given way to type-II fuzzy logic systems (T2FLSs). In other words, increased fuzziness in a description requires increased ability to handle inexact information in a reasonably proper way [41].

Fig. 2 shows the architecture of a type-II fuzzy logic system. As easily understood from this block diagram, it resembles the structure of a T1FLS with the only exception that here a new unit called *type reducer* which operates on the inference unit's outcome, right behind the defuzzification (precisiation in case of interval-valued FLSs) block. The aforementioned unit reduces type order of the incoming fuzzy set by one.

The inference unit in a T2FLS maps a T2FS into another T2FS much the same way a T1FLS maps a T1FS into another T1FS. The only difference is that, in case of a type-II fuzzy system, the operators that are involved in the inference procedure, namely *meet* ($\sqcap$), *join* ($\sqcup$), and *extended sup-star* composition ($\circ$), are generalized versions of their type-I counterparts which are respectively *t-norm*, *s-norm*, and *sup-star* composition and are defined as follows:

$$Z(z) = X \sqcap Y = \sup_{z=\min(x,y)} \min(X(x), Y(y)) \tag{3}$$

$$Z(z) = X \sqcup Y = \sup_{z=\max(x,y)} \min(X(x), Y(y)) \tag{4}$$

$$\mu_{\tilde{A}\circ\tilde{R}}(v) = \sqcup_{u}[\mu_{\tilde{A}}(u) \sqcap \mu_{\tilde{R}}(u, v)], \tag{5}$$

where X, Y, Z are type-I fuzzy sets, $\tilde{A}$ is a type-II fuzzy set and, $\tilde{R}$ is a type-II fuzzy relation. It can be shown that under minimum t-norm and maximum t-conorm, for interval sets $F_1, \ldots, F_n$ with domains $[l_1, r_1], \ldots, [l_n, r_n]$, the meet and join operations respectively reduce to [2]

$$\mathop{\sqcap}_{i=1}^{n} F_i = \left[\min_{i=1,\ldots,n} l_i \quad \min_{i=1,\ldots,n} r_i \right] \tag{6}$$

$$\bigsqcup_{i=1}^{n} F_i = \left[\max_{i=1,\dots,n} l_i \quad \max_{i=1,\dots,n} r_i \right], \tag{7}$$

where the first equation requires l_i and r_i to be non-negative, a condition that is true in our case since each F_i is a primary membership for an interval-valued fuzzy set.

Assuming that there are M rules available with the lth rule having the form

$$\text{Rule}^l: \text{ if } x_1 \text{ is } \tilde{F}_1^l \text{ and } x_2 \text{ is } \tilde{F}_2^l \text{ and } \cdots \text{ and } x_p \text{ is } \tilde{F}_p^l \text{ then } y \text{ is } \tilde{G}^l, \tag{8}$$

we can derive the corresponding *implication* membership functions as follows:

$$\mu_{\tilde{R}^l}(\mathbf{x}, y) = [\tilde{F}_1^l(x_1) \sqcap \tilde{F}_2^l(x_2) \sqcap \cdots \sqcap \tilde{F}_p^l(x_p)] \sqcap \tilde{G}^l(y_p). \tag{9}$$

Once the crisp inputs $x_1', \dots, x_p'$ delivered to the fuzzifiers, the fuzzy sets $\tilde{X}_1, \dots, \tilde{X}_p$ are determined and the *output sets* corresponding to each rule can be computed as

$$\mu_{\tilde{B}^l}(y) = \mu_{\tilde{A}_{\mathbf{x}'} \circ \tilde{R}^l}(y) = \bigsqcup_{\mathbf{x} \in \mathbf{X}} [\mu_{\tilde{A}_{\mathbf{x}'}(\mathbf{x})} \sqcap \mu_{\tilde{R}^l}(\mathbf{x}, y)], \tag{10}$$

where $\mu_{\tilde{A}_{\mathbf{x}'}}(\mathbf{x}) = \sqcap_{i=1}^{p} \mu_{\tilde{X}_i}(x_i)$. The output set can further simplified to $\mu_{\tilde{B}^l}(y) = \mu_{\tilde{R}^l}(\mathbf{x}', y) = \mu_{\tilde{G}^l}(y) \sqcap [\sqcap_{i=1}^{p} \mu_{\tilde{F}_i^l}(x_i')]$ if precise values are used for inputs.

Finally, using the *center of sets* type reducer [42], the crisp output is computed as follows:

$$Y_{gc}(y) = \sup_{z_1,\dots,z_n,w_1,\dots,w_n,y=\sum_{l=1}^{N} w_l z_l} \min[\mu_{Z_1}(z_1), \dots, \mu_{Z_1}(z_1), \mu_{W_1}(w_1), \dots, \mu_{W_n}(w_n)], \tag{11}$$

where Z_l are the *centroid* type-reduced sets of individual consequent sets with supports $[c_l - s_l, c_l + s_l]$ and W_l are the *firing sets* with supports $[h_l - \Delta_l, h_l + \Delta_l]$.

2.3. Structural description for the 3-PSP parallel manipulator

In this sub-section, a brief explanation of a 3-PSP parallel manipulator is given. Later on, we take advantage of this platform to evaluate the performance of the introduced controller. Fig. 3 depicts the solid model of the underlying parallel platform. According to the figure, the manipulator is a fully parallel mechanism with three degrees of freedom. Its structure is composed of a star-like moving platform as well as two fixed plates. The moving platform is designed such that different tools can be attached to its center (also known as moving star) depending on relevant targeted tasks. The moving star and the fixed plates are connected with three parallel legs having identical serial kinematic chains. Each of aforementioned legs consists of an active prismatic joint (P-joint), a passive spherical joint (S-joint) followed by a second passive prismatic joint. Therefore, the moving star is connected to the base via three identical serial PSP linkages (see [43] for details). Fig. 4 depicts the geometry of the 3-PSP parallel robot. Appendix A demonstrates structural parameters of the manipulator.

2.4. Dynamical modeling for the underlying 3-PSP parallel manipulator

In this sub-section a brief discussion of a dynamical model for the manipulator is presented. As far as multi-body systems such as parallel manipulators are considered, knowledge of their dynamics as well as the way their corresponding differential equations are manipulated are matters of concern. As a first step in model-based control of a system, one must infer its dynamic model. Dynamical model of a complex system serves several purposes including facilitating design of robot's structure, simulation and control. The dynamical model once derived leads to ability to control the system in a proper way. To date, a variety of methods have been introduced for analysis of robots' dynamics including Newton–Euler method, Lagrangian formulation, principle of virtual work and the natural orthogonal complement (NOC). Using each of the foregoing techniques, the system's dynamic equations obtained in closed form as follows:

$$\tau_m = M(q)\ddot{q}^{ac} + C(q, \dot{q})q^{ac} + G(q) \tag{12}$$

where $\dot{q}^{ac} = [\dot{q}_1^{ac} \ \dot{q}_2^{ac} \ \dot{q}_3^{ac}]^T$ and $\ddot{q}^{ac} = [\ddot{q}_1^{ac} \ \ddot{q}_2^{ac} \ \ddot{q}_3^{ac}]^T$ are actuated P-joints' velocity and acceleration vectors, respectively, and $\tau_m = [\tau_{m1} \ \tau_{m2} \ \tau_{m3}]^T$ is a vector containing actuators' forces/torques. M and C are 3×3 symmetric

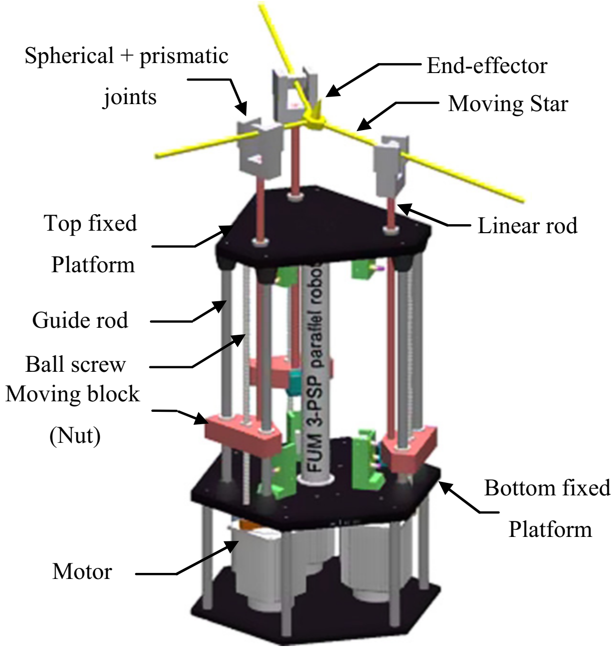

Fig. 3. Model of the manipulator in solid works.

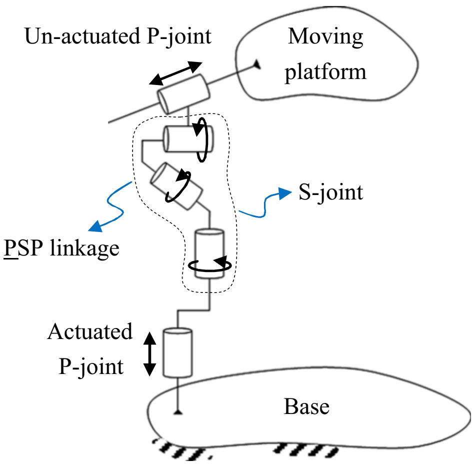

Fig. 4. The robot's geometry of one kinematic chain.

matrices for inertia and Coriolis-centrifugal forces, respectively and G is a 3×1 vector pertaining to gravity. In general, regardless of the selected approach to obtain system's dynamics, the last three matrices (M, C, G) are obtained. Note that the procedure leading to the calculation of these matrices are computationally intensive and considered as the most significant bottleneck from a computational perspective. In [44,45] a dynamical model of our robot is obtained using the NOC method. Fig. 5 depicts the concise sequence of actions followed by NOC to find the systems dynamic equations.

When modeling a dynamical system using (12), two different problems, namely the inverse dynamics and direct dynamics, emerge which should be tackled separately. In the former, given a trajectory that the end-effector must follow, our goal is to find the corresponding actuators' forces/torques required for tracking the desired trajectory. The robot's inverse dynamics is a necessary part for design of model-based controllers, see [46] for example. In the latter, on the other hand, we seek the trajectory that the end-effector follows when the input torques are fed into the platform. The direct dynamics of the robot is used for simulation purposes.

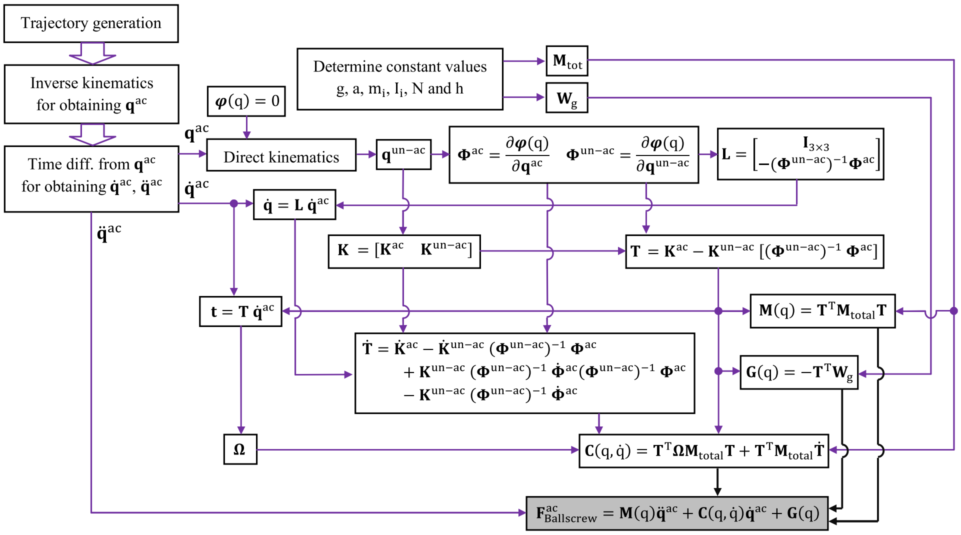

Fig. 5. Solution procedure for inverse dynamics problem.

Note that both these problems solve direct kinematic sub-problem with constraint equations as an integral part. This demands employment of numerical methods such as Newton iterative procedure. Unfortunately, these methods impose a high computational burden on controlling the underlying system. This becomes a matter of concern especially when real-time issues are important factors such as in this case. Therefore, new methods addressing such a limitation are required.

2.5. The computed-torque control (CTC) method

In this part, we briefly discuss the architecture of a non-linear model-based controlling mechanism called the computed torque control (CTC, for short). As suggested by the name, a model-based controller is dependent on the model of the system to be controlled. That is to say, a model-based controller uses the given model to yield higher performance compared to others when dealing with the system under control. The superiority of a model-based mechanism vs. those of non-model-based ones becomes evident when highly non-linear systems such as the 3-PSP parallel manipulator is being considered.

First introduced in [47], CTC is one of the most well-known model-based controllers that relies on calculation of system's dynamics matrices iteratively at each controlling step. In other words, the already described matrices (M, C and G) of the model are recalculated at each iteration. This is due to the fact that the dynamics of a variant system is constantly subject to change. As a result, the foregoing method is not considered amenable to current processing resources, a problem that becomes more severe for complex systems. The CTC is also respected as a feed-back linearizing controller in that it tries to neutralize the effects of system dynamics in the feed-back loop by canceling the dynamic terms. In this sense, from a broader perspective, the system simplifies to a linear form and can be controlled by help of common linear-controllers, such as PIDs. However, linearization of a system is highly contingent upon precise identification of the system at hand, which is rather hard to achieve if not impossible. In fact it is for this reason that open loop controllers are not employed in practice. This also means that, the more we are uncertain about the inferred model, the worse is the performance of the ultimate controller; and hence the most challenging and problematic issue when dealing with model-based controllers (including the CTC) is their sensitivity to noise. Having said this, the significant contribution of our designed controller as a variation of the CTC, is to handle uncertainty that is conducive to inexact system parameters (C, M and G). Note that, different sources of uncertainty influence the outcome. These include the uncertainty added to the system due to inexactness in manipulator's structural measurements (structural uncertainty), the uncertainty that we face when numerical solutions are employed (such as the calculation of direct kinematics) and the uncertainty that we incur because of wear and tear issues, joint backlash and friction, link flexure and other influencing factors. Commonly, the structural uncertainty is the largest source of uncertainty in most applications.

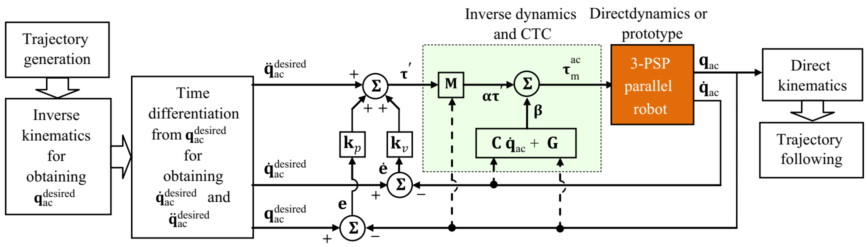

Fig. 6. Block diagram for computed torque method.

Fig. 6 depicts the block diagram of the CTC when applied to our 3PSP robot model. In the scheme devised by the CTC, the control process is broken down into two main parts, namely the *model-based* portion and the *servo* portion. The model-based portion contains a model of non-linearity [46]. Hence, it plays the role of a linearization function. The procedure can be expressed as:

$$\tau_m = \alpha\tau' + \beta \tag{13}$$

where

$$\begin{aligned} \alpha &= M(q) \\ \beta &= C(q, \dot{q})\dot{q}^{ac} + G(q) \end{aligned} \tag{14}$$

The servo portion on the other hand, takes the form of

$$\tau' = \ddot{q}_{ac}^{desired} + k_v\dot{e} + k_p e \tag{15}$$

where $\ddot{q}_{ac}^{desired}$, e and $\dot{e}$ are respectively the desired joint acceleration, joint position and velocity servo errors. The values associated with the unknown gains (k_p and k_v) are also determined based on a desired performance specification. As easily implied by the figure, it is the servo portion that plays the key role through the control process. It also behaves such that the overall platform model acts as a unit mass linear system which is then harnessed by a PD controller.

3. The proposed controller

The chief sources of uncertainty in the underlying platform are the sensory noise and uncertain parameters of the structure. These uncertainties eventually lead to uncertain dynamics or more precisely uncertain matrices M, C and G. To account for the foregoing types of uncertainty, during the training phase, we added some noise (based on a predefined signal to noise ratio) to the elements of those matrices computed in each iteration of solving dynamics equations. The objective is to evaluate the control performance of an interval-valued fuzzy controller when measurements are subject to uncertainty.

Fig. 7(a) demonstrates the proposed architecture which is based on singleton T2FL approximators in the feedback loop. As the figure implies, the uncertain matrices introduced earlier are approximated via type-II fuzzy logic systems. However this time, as opposed to most of common T2FLSs, the employed T2 systems are of special kind in that they exclude the defuzzification (precisiation, in our case) unit, as depicted in Fig. 7(b). In other words, the output of each approximator unit is a T1 fuzzy number (interval) rather than a crisp number. This is done on purpose as doing defuzzification (precisiation) is equivalent to omitting the uncertainty bounds which can be regarded as confidence intervals associated with the approximated matrices, and therefore depriving ourselves from this outstanding capability that T2FSs provide. In so doing, we do not perform defuzzification (precisiation) until the last step, right behind the manipulator block, where the torque vector feeds into the plant model. Note that depending on the dynamical system to be controlled, the proposed architecture may offer other benefits, as well. For example, if the support of the final fuzzy number (interval) that should be defuzzified (precisiated) is larger than a threshold, it could mean that an intolerable amount of uncertainty exists in the system and hence caution must be exercised.

Put it together, the fuzzy torque vector can be now computed as

$$\hat{\tau}_m^{ac} = \hat{M} \otimes \tau' \oplus \hat{C} \otimes \dot{q}_{ac} \oplus \hat{G}, \tag{16}$$

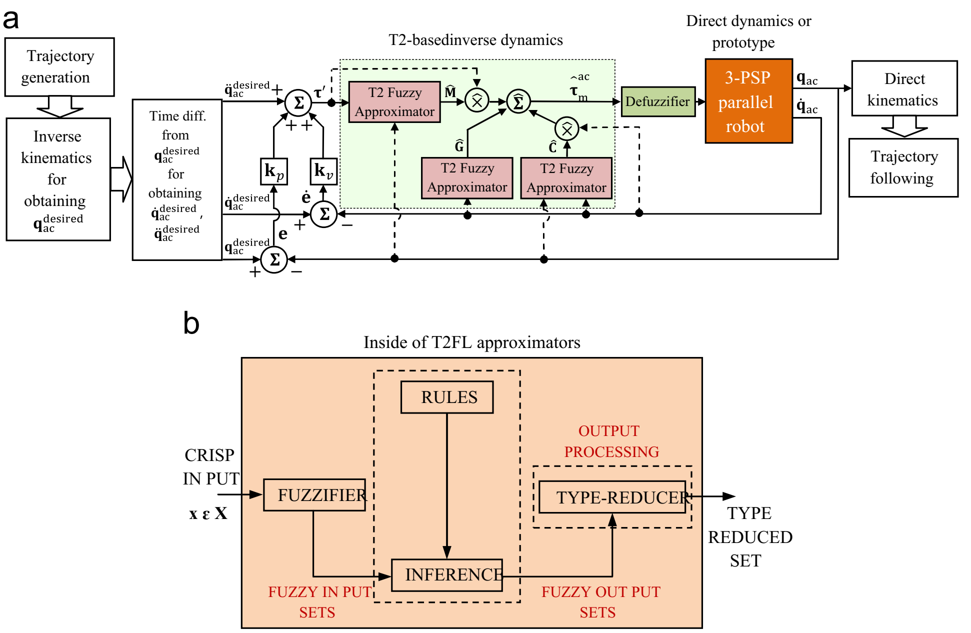

Fig. 7. (a) The proposed model. (b) Inside of the T2 approximators.

where τ' and $\dot{q}_{ac}$ are vectors of crisp numbers and $\hat{M}$, $\hat{C}$, $\hat{G}$ are matrices of fuzzy numbers. The operator $\otimes$ indicates a variation of matrix multiplication in which the summations and multiplications are replaced by the extended ones which were defined in Section 2.1; likewise, the operator $\oplus$ designates the summation of two vectors with fuzzy numbers (intervals) as their elements.

Once the vector of fuzzy (interval) torques computed, it is deffuzified using a center of sets (CoS) deffuzifier to get the final crisp torque value which is then fed into the plant to make it track a desired trajectory. A noteworthy point is that, for the special case of interval-valued FLSs (using the theorems presented in [48]), the precisiator may be moved back to the approximator units without experiencing a change in the final result. This remarkably diminishes the computational burden imposed by the approximators. Yet, it should be emphasized that the simplification is only applicable to the class of interval-valued fuzzy logic systems and not any general type of T2FLSs (e.g. Gaussian T2FLSs) keeping in mind the fact that the proposed control architecture is not limited to interval-valued fuzzy sets and it can be applied to more general cases provided that the overall controller remains amenable to computational resources.

There are two other aspects of any type-II fuzzy system that are of significant importance, namely the selection of fuzzy sets and fuzzy rules. There are generally two approaches to selection of fuzzy sets and fuzzy rules for designing a T2FLS [2], each of which has been successfully implemented in different applications. The first is to devise fuzzy sets and rules in a fully independent manner. The second approach which is called partially dependent method extends a type-I fuzzy controller to a type-II counterpart only by generalizing its fuzzy sets while leaving its architecture intact. This last strategy has two benefits [7]. First, it reduces the number of design parameters and second, the designed controller can be objectively compared and contrasted to its type-I counterpart which was a crucial goal in this research. Accordingly, we pursued the second approach to develop the fuzzy sets and rules.

In order to find the set of fuzzy rules for our type-I and type-II controllers, a spatial ellipsoid trajectory was given to the CTC controller in a noise free environment. Note that, as is, the CTC computations cannot be performed for real-time complex dynamical systems like a parallel robot.

Similar to most published studies to date, in this work we utilize IT2FL variables with Gaussian primary membership functions and uncertain means. As mentioned in earlier sections, interval type-II fuzzy systems are the most prevalent types of F2FLSs. This is due to the fact that they are much more computationally tractable as a result of further simplifications in type-reduction procedure. This is very critical in complex dynamical systems since the designed controller is subject to real-time limitations. On the other hand, we use Gaussian primary membership functions since they are both smooth and better represent the noise characteristics. We choose the mean variations corresponding to

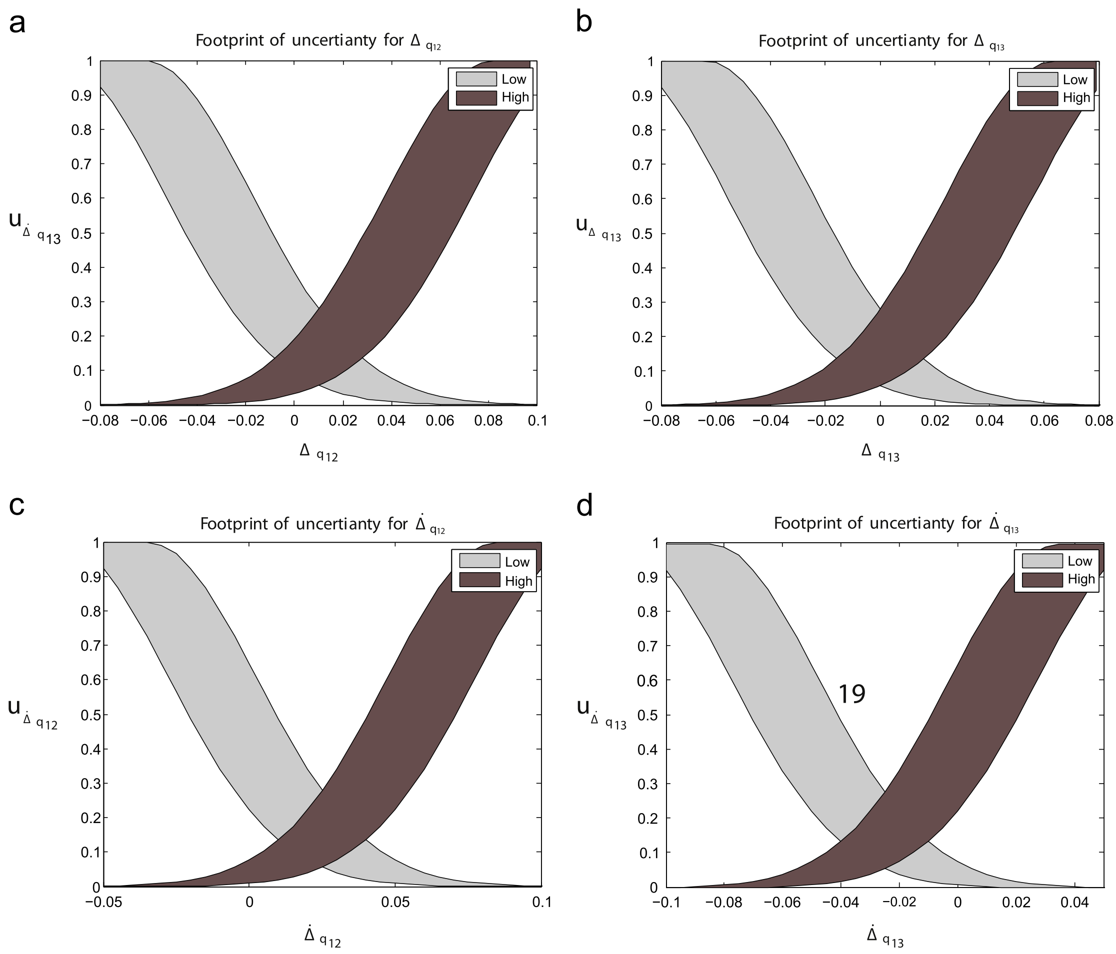

Fig. 8. The input variables are represented with 2 Gaussian Primary membership functions.

membership functions to be inversely proportional to the signal-to-noise ratio (SNR) that we normally observe in a real-world platform of this kind. That is, having a subjective estimation of the aforementioned SNR, the membership functions are generated. After all, having the IT2 membership functions (IT2MFs) generated, it only remains to produce the relevant rules, for which the following steps are carried out in sequence. Then using the inverse kinematics equations, the system variables, q_1, q_2, q_3, $\dot{q}_1$, $\dot{q}_2$ and $\dot{q}_3$, were subsequently computed and recorded. As far as the torque values concerned, the state space of the system can be reduced in size by a mapping a new space, $\{\Delta \mathrm{q}_{12}, \Delta \mathrm{q}_{13}\}$ (Δq_{ij} indicates the difference angle between the ith and jth actuated joints), without a change in the derived torques [45]. This results in a significant reduction in the number of fuzzy rules. Having the new state space, we divided the domain of each variable, that was observed by having the model follow the trajectory mentioned earlier, into two sub-domains and by using *look-up table* procedure [49], 16 fuzzy rules were derived. Dividing the domains into more than two sub-domains did not yield to a noticeable improvement. In case of our type-I controller, we used Gaussian fuzzy sets as they effectively represent uncertainties available in the platform. To extend these fuzzy sets to the class of type-II sets we blurred each of these Gaussian membership functions by sliding them to left and right and hence creating interval-valued Gaussian primary type-II membership functions. For each antecedent, we empirically chose a mean variation equal to one-tenth of the length of its corresponding sub-division in the look-up table. Fig. 8 depicts the footprint of uncertainty for the antecedents. For the consequent fuzzy sets we selected the mean variations to be inversely proportional to the signal-to-noise ratio, in each experiment (see Section 4). If the SNR cannot be estimated (as might be the case in a real-world setting), an alternative approach would be to find the optimal mean variations empirically.

Finally, it should be pointed out that in this paper, we only consider the effects of *numerical* uncertainty. The *numerical* uncertainty leads to uncertain inputs contrary to the *linguistic* uncertainty which results in uncertain rules.

Table 1
Control parameters for each method.

Type of controller	K_p	K_d	Noise level (α)	No. of MFs
CTC	100	10	–	–
T1 fuzzy CTC	100	10	0%, 5%, ..., 50%	2
T2 fuzzy CTC	100	10	0%, 5%, ..., 50%	2
PD	1000	100	–	–

4. Simulation results

In this section, we perform different trials to show how the newly introduced controller surpasses other competing methods. We compare the results both in terms of performance in the face of uncertainty and computational complexity with respect to defined criteria. The selected approaches for comparison are a PD, a classical CTC, and a type-I fuzzy CTC based controller, explained shortly. Through the whole simulation process, a spatial ellipsoid trajectory is the desired path that we want the manipulator's tool-tip to trace. Table 1 demonstrates parameters associated to each method through the whole simulation.

4.1. Type-I reduced FLC

Similar to a wide variety of papers published to date, in this work, our main focus is on the behavior of the proposed controller for different signal-to-noise ratios. Accordingly, to compare the results, the best counterpart is a type-I reduced controller. That is to say, we are interested to see how the new types of fuzzy systems can do the job with better performances for us. Usually this is done by downgrading the designed type-II fuzzy system into type-I, simply through reducing the type-II membership functions by one degree, or equivalently, by setting the amounts designated as *mean variations* to zero (this clearly implies that there is not any remaining uncertainty in description of fuzzy variables after removing the footprint of uncertainty). In that case, since the output of a type-I approximator is a crisp number, there is no need for any kind of algebraic operation on type-I fuzzy numbers.

The implemented T1FLS is characterized with the same parameters as the IT2FLS except that it is reduced to a T1FLS. This is necessary in order to have an appropriate comparison.

4.2. Proportional-derivative (PD)

As a widespread classical controller of the PID family, we have implemented a PD controller according to the feedback gains demonstrated in Table 1. The advantage with this type of controller is the ease of implementation and the fact that it is realizable in real-time and hence is useful for online applications. The drawback, on the other hand, is that the controllers of this family are only suitable for linear plants; a problem which becomes more severe when the degree of non-linearity increases or when the robot is designed to work at high speeds and precisions. This is also easily understood by consulting Table 1. According to this table, the proportionate and derivative gains selected for the PD controller during the simulation scenarios are by far greater than the ones associated to the other three approaches, or the controller becomes unstable otherwise. Note that we do not embed the *integral* gain in order to make the PD system more comparative to the CTC controller. Note also that since the PD controller does not consider system's dynamics (M, C and G), we apply it to the exact model of the manipulator in the context of simulations.

4.3. Computed-torque control (CTC)

Based on the brief discussion presented earlier, as a model-based classical controller, the CTC is known to offer high precision but at cost of having several important limitations including a required exact model, a very computationally powerful processor, and a deterministic environment to operate in. These conditions altogether make CTC an ideal choice merely in simulation environments.

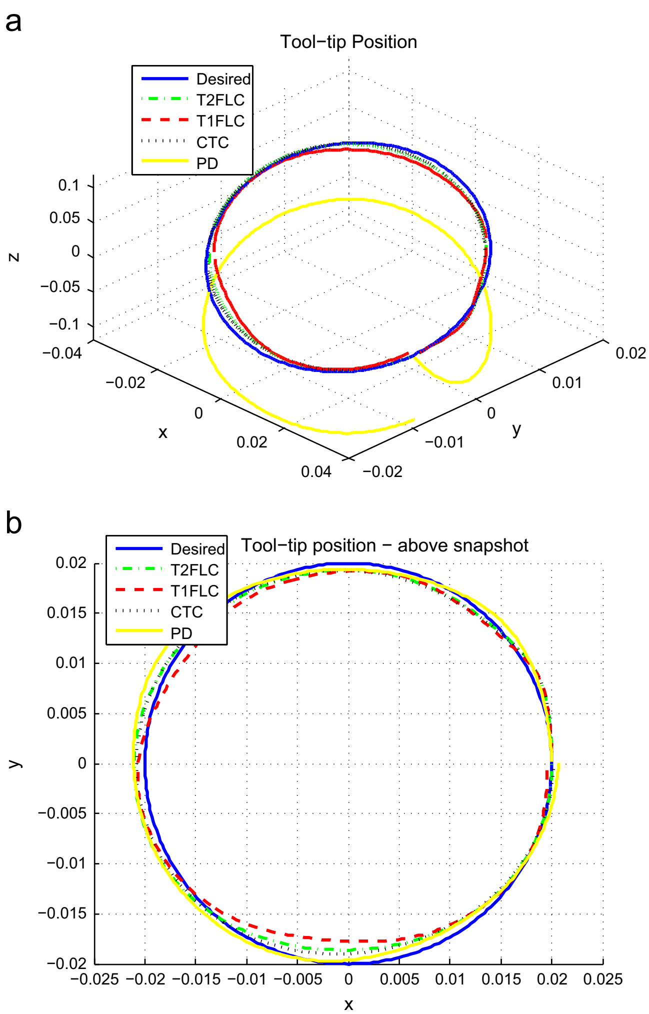

Fig. 9. Path followed in face of 0% noise ($\alpha = 0\%$).

As previously elaborated, during the simulations the dynamic matrices are subject to some additive noise with different noise levels. In other words, for each element m_{ij} of the foregoing matrices, the corresponding noisy element is obtained as follows:

$$\hat{m}_{ij} = m_{ij}(1 + \alpha.r) \tag{17}$$

where r is a uniformly distributed random number over $[-1, +1]$ and α is the amount of noise in percentage. For a quantitative comparison between the performance of the described T1FLS and T2FLS, we also define the following measure based on *sum of absolute errors*, as follows:

$$\text{Performance of T2 vs. T1} = \frac{(SAE_{T_1} - SAE_{T_2})}{SAE_{T_1}} \times 100\% \tag{18}$$

In the first part of our simulations, we employed all previously discussed controllers when there is no noise to interfere. We intend to observe the applicability of each method to an ideal platform. Fig. 9(a) depicts the traversed paths from 3D perspective and Fig. 9(b) shows a 2D snapshot taken from above. As shown in the figures, all control methods except PD track the desired trajectory with good precision. Table 2 confirms this conclusion numerically. It is also clear from the table that the T2FLS outperform the T1FLS even when we are not facing uncertainty. This verifies that a T2FLS may be a good alternative for a model-based controller. It should also be emphasized that using a classical

Table 2
Sum of absolute errors (SAE) for $\alpha = 25\%$ and $\alpha = 0\%$.

Type of employed controller	0% noise	25% noise
CTC	0.5354	–
T1 fuzzy CTC	0.9540	2.0294
T2 fuzzy CTC	0.6256	1.7603
PD	14.3090	–

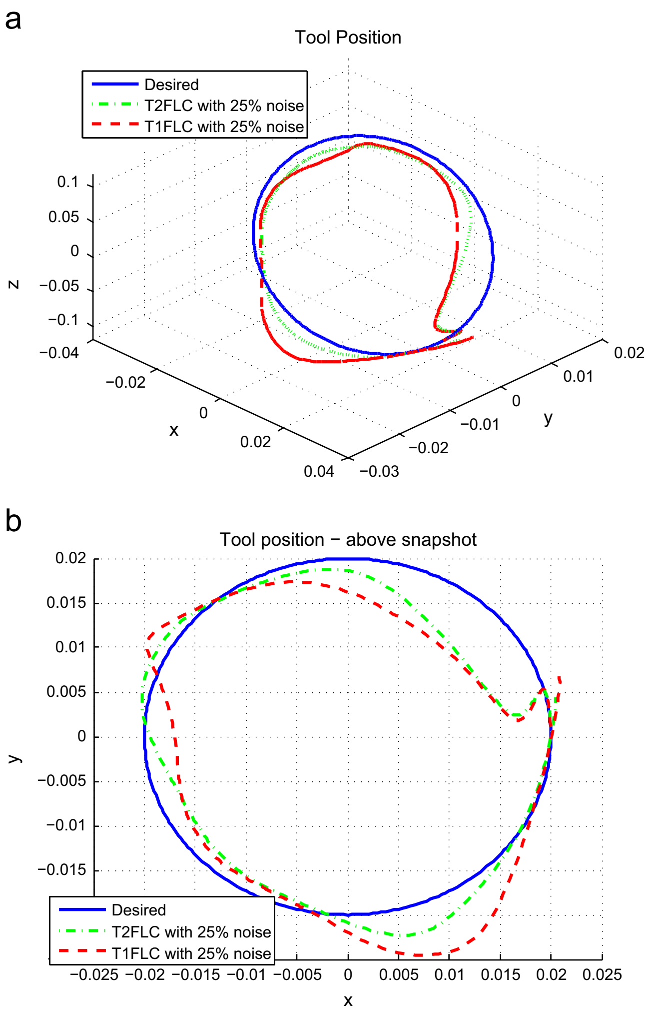

Fig. 10. Path followed in face of 25% noise ($\alpha = 25\%$).

CTC is conducive to an iterative computational process which entails a high overhead to the processor as opposed to a fuzzy controller (of both types.) Note that the amount of error that the CTC incurs originates from the numerical uncertainties that emerge when solving complex dynamic equations. Another noteworthy point is that during the whole simulation, the PD lags the desired path by a constant offset which stems from the fact that having disregarded the gravity matrix (G) leads to a steady state error in the control process.

As always credited with, a type-II fuzzy logic system is a powerful tool to handle uncertainty especially when lower order counterparts (i.e. type-I and classical systems) fall short in fulfilling it. By doing so, in the next scenario, we intend to clarify this by adding 25% noise to our system and observe the behavior of both T1 and T2 systems in an uncertain environment. In this regard, Fig. 10(a) and (b) shows the path chased by the manipulator's tool-tip once these

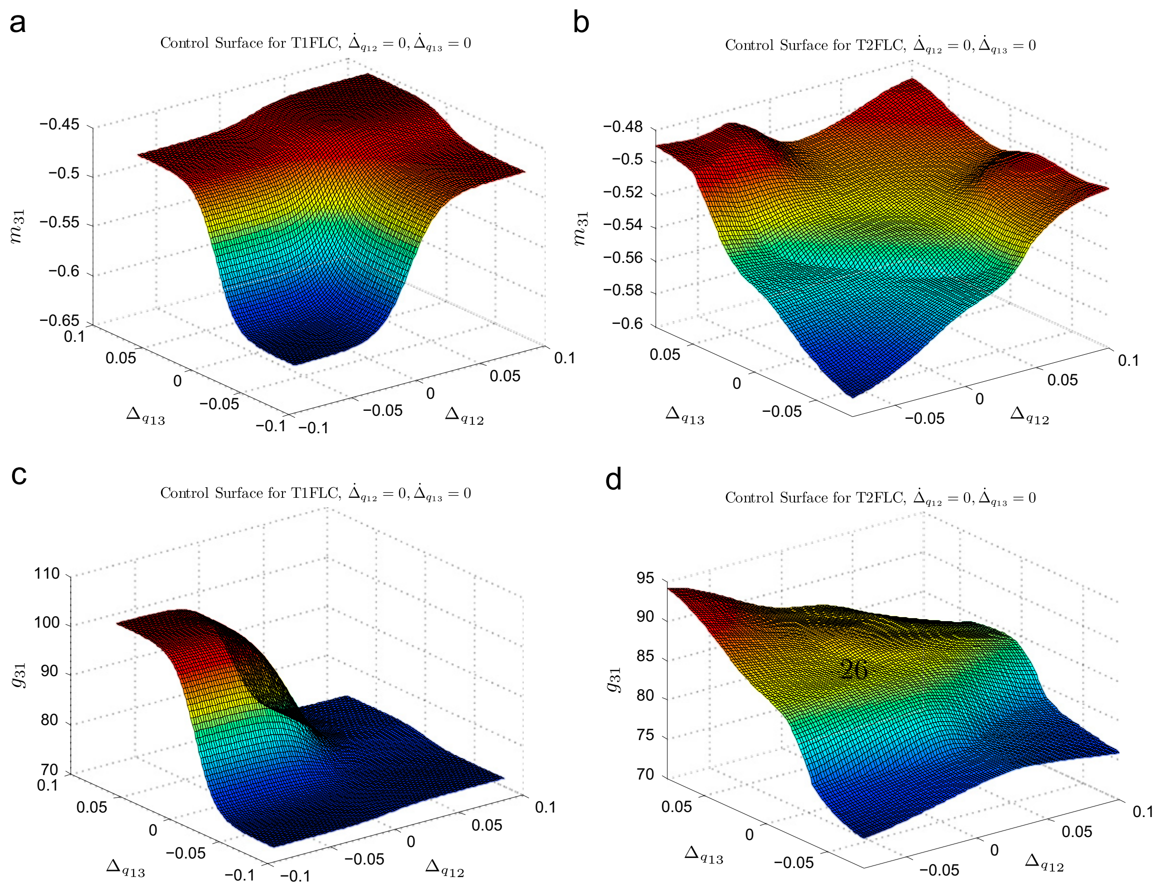

Fig. 11. Control Surface associated to the T2FLC and T1FLS for two different configurations.

Table 3
Controlling response time for each method.

Control mechanism	Execution time
PD	Less than 0.1 ms
CTC	255 ms
T1 fuzzy CTC	0.15 ms
T2 fuzzy CTC	1.5 ms

controllers are put to use. As easily implied by the second figure and Table 2, the proposed controller tracks the desired trajectory in a more accurate and smoother fashion than its downgraded counterpart. The smoothness comes from the fact that type-II fuzzy systems in general behave in a smooth way.

To be more specific, Fig. 11 depicts the control surfaces pertaining to the type-II and type-I approximators used in this scenario for two elements of the dynamic matrices, namely m_{31} and g_{31}. Fig. 11(a) and (c) shows the control surface associated to the T1LFSs designed for $\alpha = 25\%$ when $\Delta \underline{q}_{12}$, $\Delta \underline{q}_{13}$ take on values in their domain and $\Delta \dot{\underline{q}}_{12}$, $\Delta \dot{\underline{q}}_{13}$ remain zero. Similarly, Fig. 11(b) and (d) shows the corresponding T2FLSs. Notice how latters (type-II surfaces) are smoother than the formers (type-I surfaces) especially around the origin. This holds more or less for other employed approximators, as well. The state of being smooth especially around the origin is a chief reason for why T2FLSs better eliminate persistent oscillations around an operating point [50,34].

In regards to robustness, the performance of a controller when different levels of uncertainty are experienced are a matter of concern. In this last scenario, the performance of both T1 and T2 control mechanisms over a wide range of

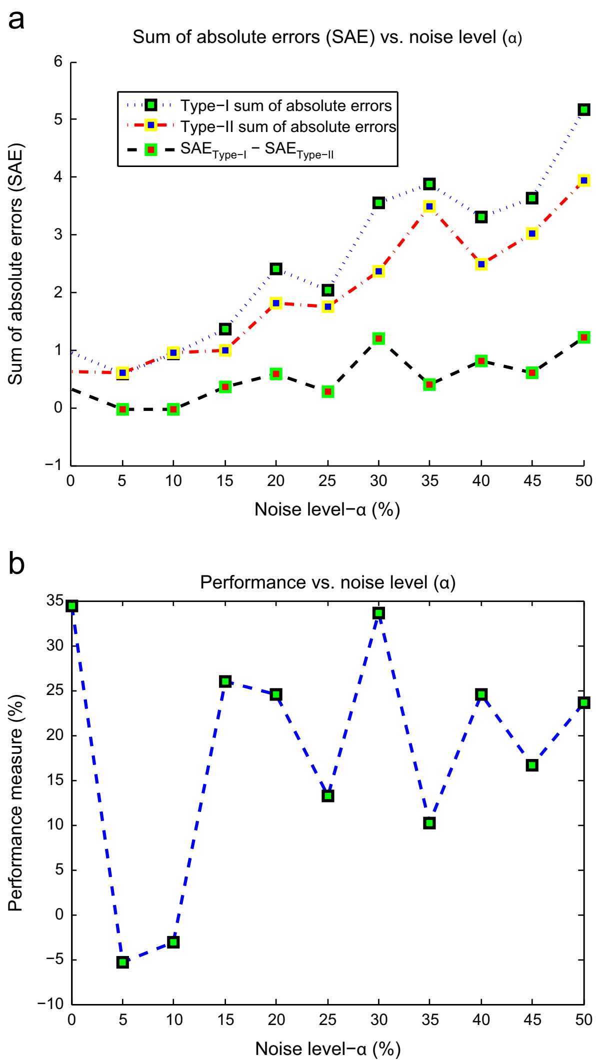

Fig. 12. Error and performance measure vs. α.

noise levels are investigated. Accordingly, we repeat the previous scenario several times with α as the only changing parameter in each trial. Fig. 12(a) shows the plot of error as well as error difference versus α. Based to this diagram, the SAE for the proposed control mechanism is better than the T1 counterpart in nearly all of the trials. However, for lower amounts of uncertainty, the amounts for this error are competitive. The point is that the strength of type-II fuzzy systems becomes prominent when a considerable amount of uncertainty is present in the control process. In our case, this is evident for $\alpha > 10$. Notice that for $\alpha = 30\%$, the error difference diagram takes its highest value. This is due to the selected membership functions (see Fig. 8). In other words, the generated T2 membership functions are sub-optimal for a circumstance in which $\alpha = 30\%$. To quantitatively evaluate our approach vs. its downgraded counterpart, Fig. 12(b) depicts the performance of each method at different noise levels based on the formula proposed in (18). According to this figure, the performance of the latter approach surpasses the former in almost all cases with the highest (disregarding the case of $\alpha = 0\%$) values coming at $\alpha = 30\%$ which in this sense conforms to Fig. 12.

In the end, it should be noted that though our focus in this work is not the real-time implementation issues, we did compare response time corresponding to all the previously mentioned controllers, as demonstrated in Table 3. Note that by real-time execution we mean that the control action must be carried out within a time interval short enough to be useful [51] (2 ms in our case). The controlling codes were all written in C++ for the sake of optimality and run on a computer with a 3.0 GHz Pentium 4 processor.

Table A1
Architectural and physical parameter values for underlying the 3-PSP parallel robot.

a_i's length	a	181	mm
Length of the tool	h	70	mm
Mass of each moving block	$m_1 = m_2 = m_3$	7.175	kg
Mass of S-joints' constituents	$m_4 = m_5 = \ldots = m_{12}$	0.357	kg
Mass of Star	m_{13}	1.758	kg
Inertia of each moving block	$I_1 = I_2 = I_3$	diag(0.003,0.39,0.3294)	kg/m^2
The inertia of each S-joint	$I_4 = I_5 = \ldots = I_{12}$	diag(0.094, 0.094, 0.1117) $\times 10^{-3}$	kg/m^2
Star inertia	I_{13}	diag(0.6451,1.2901,0.6451)	kg/m^2

The PD controller, as easily observed from the table, is the fastest method but according to the discussion presented earlier, it has a poor performance for complex systems especially when it is required to do a control action with high speed and precision. On the other hand,the time taken by CTC to complete a single control cycle is remarkably high which makes it totally useless for our case even if it may be an accurate model. Finally, both the T1 and T2 fuzzy based CTC meet our timing expectations (although the former is faster) and thus can be successfully tested in a real experiment (as suggested in Section 6).

5. Conclusion

In this paper a new T2FL based method for control of a complex non-linear dynamical system is proposed subject to uncertainty. In short, we enhance a model-based classical controller to cover two important issues, the numerical uncertainty and the real-time realization. We compare the proposed method to the type reduced controller with a similar architecture as well as two classical controllers. The observed results show that the proposed mechanism is both robust to the noise and amenable to current computer processing resources. Overall, the new method emerges as a promising approach in other dynamical systems as well.

6. Future works

We believe that to fully handle the uncertainty in complex dynamical systems, a true type-II fuzzy logic system is a preferred alternative. Our work, similar to many others in its category, exploits interval based uncertainty handling due to its computational advantages. In other words, the excessive computational burden of type-II fuzzy systems leaves a challenging problem for real-time complex decision making/analysis. As a future work we aim to address this problem.

Appendix A. Details of the employed manipulator

See Table A1.

References

[1] J. Mendel, H. Wu, Uncertainty bounds and their use in the design of interval type-II fuzzy logic systems, IEEE Trans. Fuzzy Syst. 10 (5) (2002) 622–639.
[2] J. Mendel, Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions, Prentice-Hall, Upper-Saddle River, NJ, 2001.
[3] N.N. Karnik, J.M. Mendel, Centroid of a type-II fuzzy set, Inf. Sci. 132 (2001) 195–220.
[4] L.A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning-1, Inf. Sci. 8 (1975) 199–249.
[5] J.M. Mendel, Uncertainty, fuzzy logic, and signal processing, Signal Process. J. 80 (2000) 913–933.
[6] J.M. Mendel, Computing with words, when words can mean different things to different people, in: Proceedings of the Third International ICSC Symposium on Fuzzy Logic and Applications, Rochester, NY, 1999, pp. 158–164.
[7] O. Linda, M. Manic, Uncertainty-robust design of interval type-2 fuzzy logic controller for delta parallel robot, IEEE Trans. Ind. Inf. 7 (4) (2011) 661–670.
[8] C.-F. Juang, Y.-W. Tsao, A type-2 self-organizing neural fuzzy system and its FPGA implementation, IEEE Trans. Syst. Man Cybernet. Part B: Cybernet 38 (6) (2008) 1537–1548.
[9] Y.-S. Chen, L. Yao, Robust type-2 fuzzy control of an automatic guided vehicle for wall-following, in: International Conference on Soft Computing and Pattern Recognition. SOCPAR '09, 2009, pp. 172–177.

[10] D. Wu, W. Tan, A type-2 fuzzy logic controller for the liquid-level process, in: 2004 IEEE International Conference on Fuzzy Systems, 2004. Proceedings. vol. 2, 2004, pp. 953–958.
[11] T. Dereli, A. Baykasoglu, K. Altun, A. Durmusoglu, I.B. Turksen, Review: industrial applications of type-2 fuzzy sets and systems: a concise review, Comput. Ind. 62 (2) (2011) 125–137.
[12] O. Castillo, P. Melin, A review on the design and optimization of interval type-2 fuzzy controllers, Appl. Soft Comput. 12 (4) (2012) 1267–1278.
[13] R.-E. Precup, H. Hellendoorn, A survey on industrial applications of fuzzy control, Comput. Ind. 62 (3) (2011) 213–226.
[14] O. Castillo, P. Melin, Soft Computing for Control of Non-Linear Dynamical Systems, Physica-Verlag, 2001.
[15] P. Melin, O. Castillo, Intelligent control of non-linear dynamic plants using type-2 fuzzy logic and neural networks, in: Fuzzy Information Processing Society. Proceedings. NAFIPS. 2002 Annual Meeting of the North American, 2002, pp. 22–27.
[16] P. Melin, O. Castillo, A new method for adaptive model-based control of non-linear plants using type-2 fuzzy logic and neural networks, in: The 12th IEEE International Conference on Fuzzy Systems. FUZZ '03, vol. 1, 2003, pp. 420–425. http://dx.doi.org/10.1109/FUZZ.2003.1209400.
[17] P. Melin, O. Castillo, A new method for adaptive control of non-linear plants using type-2 fuzzy logic and neural networks, Int. J. Gen. Syst. 33 (2) (2004) 289–304.
[18] Y.-M. Li, Y.-Y. Sun, Type-2 t-s fuzzy impulsive control of nonlinear systems, Appl. Math. Model. 36 (6) (2012) 2710–2723.
[19] T. Kumbasar, I. Eksin, M. Guzelkaya, E. Yesil, Type-2 fuzzy model based controller design for neutralization processes, ISA Trans. 51 (2) (2012) 277–287.
[20] R. Sepulveda, O. Montiel, O. Castillo, P. Melin, Embedding a high speed interval type-2 fuzzy controller for a real plant into an FPGA, Appl. Soft Comput. 12 (3) (2012) 988–998.
[21] A. Al-khazraji, N. Essounbouli, A. Hamzaoui, F. Nollet, J. Zaytoon, Type-2 fuzzy sliding mode control without reaching phase for nonlinear system, Eng. Appl. Artif. Intell. 24 (1) (2011) 23–38.
[22] E. Kayacan, Y. Oniz, A.C. Aras, O. Kaynak, R. Abiyev, A servo system control with time-varying and nonlinear load conditions using type-2 tsk fuzzy neural system, Appl. Soft Comput. 11 (8) (2011) 5735–5744.
[23] I. Atacak, O. Bay, A type-2 fuzzy logic controller design for buck and boost dc–dc converters, J. Intell. Manufact. (2010) 1–12.
[24] G.M. Mndez, L. Leduc-Lezama, R. Cols, G. Murillo-Perez, J. Ramirez-Cuellar, J.J. Lpez, Modelling and control of coiling entry temperature using interval type-2 fuzzy logic systems, Ironmaking Steelmaking 37 (2) (2010) 126–134.
[25] W. Flores, E. Mombello, J. Jardini, G. Ratta, Fuzzy risk index for power transformer failures due to external short-circuits, Electr. Power Syst. Res. 79 (2009) 539–549.
[26] Y. Qui, H. Yang, Y. Zhang, Y. Zhao, Polynomial regression interval-valued fuzzy systems, soft computing—a fusion of foundations, Methodol. Appl. 12 (2008) 137–145.
[27] R. Gupta, U. Pareek, I. Kar, Soft computation of turbine inlet temperature of gas turbine power plant using type-2 fuzzy logic systems, in: Fuzzy Systems Conference, FUZZ-IEEE 2007, IEEE International, 2007, pp. 1–6. http://dx.doi.org/10.1109/FUZZY.2007.4295384.
[28] M.-Y. Hsiao, T.-H.S. Li, J.-Z. Lee, C.-H. Chao, S.-H. Tsai, Design of interval type-2 fuzzy sliding-mode controller, Inf. Sci. 178 (6) (2008) 1696–1716.
[29] P. Torres, D. Saez, Type-2 fuzzy logic identification applied to the modeling of a robot hand, in: IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2008 (IEEE World Congress on Computational Intelligence), 2008, pp. 854–861.
[30] R. Martinez, O. Castillo, L. Aguilar, Optimization with genetic algorithms of interval type-2 fuzzy logic controllers for an autonomous wheeled mobile robot: a comparison under different kinds of perturbations, in: IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2008 (IEEE World Congress on Computational Intelligence), 2008, pp. 901–908. http://dx.doi.org/10.1109/FUZZY.2008.4630477.
[31] L. Astudillo, O. Castillo, L. Aguilar, R. Martinez, Hybrid control for an autonomous wheeled mobile robot under perturbed torques, in: P. Melin, O. Castillo, L. Aguilar, J. Kacprzyk, W. Pedrycz (Eds.), Foundations of Fuzzy Logic and Soft Computing, Lecture Notes in Computer Science, vol. 4529, Springer, Berlin, Heidelberg, 2007, pp. 594–603.
[32] N. Baklouti, A. Alimi, Motion planning in dynamic and unknown environment using an interval type-2 tsk fuzzy logic controller, in: Fuzzy Systems Conference. FUZZ-IEEE 2007. IEEE International, 2007, pp. 1–6. http://dx.doi.org/10.1109/FUZZY.2007.4295647.
[33] H. Hagras, A type-II fuzzy logic controller for autonomous mobile robots, in: Proceedings of IEEE FUZZ Conference, Budapest, Hungary, 2004.
[34] H.A. Hargras, A hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots, IEEE Transaction on Fuzzy Systems 12 (14) (2004).
[35] C.-D. Li, J.-Q. Yi, Y. Yu, D.-B. Zhao, Inverse control of cable-driven parallel mechanism using type-2 fuzzy neural network, Acta Automat. Sin. 36 (3) (2010) 459–464.
[36] D. Dubois, H. Prade, Operations on fuzzy numbers, Int. J. Syst. Sci. 9 (1978) 613–626.
[37] D. DuBois, H. Prade, Fuzzy Interval Analysis. The Handbooks of Fuzzy Sets Series, Springer-Verlag, 2000 pp. 483–581.
[38] W.A. Lodwick, Interval and Fuzzy Analysis: A Unified Approach Advances in Imaging and Electron Physics, vol. 148, Elsevier, 2007 pp. 75–192.
[39] C. Lynch, H. Hagras, V. Callaghan, Parallel type-2 fuzzy logic co-processors for engine management, in: Fuzzy Systems Conference, FUZZ-IEEE 2007, IEEE International, 2007, pp. 1–6.
[40] C. Lynch, H. Hagras, V. Callaghan, Using uncertainty bounds in the design of an embedded real-time type-2 neuro-fuzzy speed controller for marine diesel engines, in: 2006 IEEE International Conference on Fuzzy Systems, 2006, pp. 1446–1453.
[41] E. Hisdal, The IF THEN ELSE statement and interval-valued fuzzy sets of higher type, Int. J. Man–Mach. Stud. 15 (1981) 385–455.
[42] N.N. Karnik, J.M. Mendel, Q. Liang, Type-II fuzzy logic systems, IEEE Trans. Fuzzy Syst. 7 (1999) 643–658.
[43] A. Rezaei, A.R. Akbarzadeh, P. Mahmoodi Nia, M.R. Akbarzadeh -T, Position, Jacobian and workspace analysis of a 3-PSP spatial parallel manipulator, Robot Comput. Integr. Manuf. 29 (4) (2013) 158–173.

[44] A. Rezayi, Control of a 3 D.O.F PSP Parallel Manipulator, Master's Thesis, Ferdowsi University of Mashhad, Mashhad, 2011.
[45] H.R. Hassanzadeh, A New Type-II Fuzzy Logic Based Controller for Non-linear Dynamical Systems with Application to a 3-PSP Parallel Robot, Master's Thesis, Department of Computer Engineering, Ferdowsi University of Mashhad, Iran, 2010.
[46] J. Craig, Introduction to Robotics: Mechanics and Control, third ed., Prentice-Hall, 2004.
[47] R.P. Paul, Modeling, Trajectory Calculation, and Serving of a Computer Controlled Arm, Technical Report AIM-177, Stanford University, Artificial Intelligence Laboratory, 1972.
[48] N.N. Karnik, J.M. Mendel, Operations on type-II fuzzy sets, Fuzzy Sets Syst. 122 (2001) 327–348.
[49] L. Wang, A Course in Fuzzy Logic and Control, Prentice-Hall, Englewood Cliffs, NJ, 1997.
[50] I.A. Hameed, Simplified architecture of a type-2 fuzzy controller using four embedded type-1 fuzzy controllers and its application to a greenhouse climate control system, Proc. Inst. Mech. Eng., Part I: J. Syst. Control Eng. 223 (5) (2009) 619–631.
[51] R. Arkin, Behavior-Based Robotics, MIT Press, 1998.