Available online at www.sciencedirect.com

ScienceDirect

Journal of Electrical Systems and Information Technology 2 (2015) 338–346

www.elsevier.com/locate/jesit

Trajectory tracking in 2D under fuzzy controller with variable sampling

Ján Cigánek

Faculty of Electrical Engineering and Information Technology, Slovak University of Technology in Bratislava, Ilkovičova 3, 841 04 Bratislava, Slovakia[1]

Available online 9 December 2015

Abstract

The paper deals with an effective approach of the robust controller design based on the fuzzy logic, and algorithms for variable sampling of trajectory points to improve the control performance of trajectory tracking. The proposed controller design and sampling algorithms are verified in the case study of the selected mechatronic system. All presented results are reached in co-simulation of two different modeling environments, Matlab-Simulink and MSC Adams. MSC Adams is used for the dynamics of the mechatronic system and Matlab-Simulink for the control part of the co-simulation, respectively.

Keywords: Control system; Co-simulation; Fuzzy logic; Mechatronic system; Sampling algorithm; Trajectory tracking

1. Introduction

In many industrial processes there are important activities, where failure can lead to disasters with huge impact on human life, health and environment (air and water pollution, contamination, etc.). This is exactly the place for usage of the various mechatronic and robotic systems with strictly set requirements for the stability and quality of the control system.

To avoid such adverse situations it is necessary to develop procedures, methods and algorithms to be able to identify critical situations in time and conditions in different processes, and select appropriate strategies and management practices to get the extraordinary and emergency states of controlled system to operational and safe states.

Because industrial processes are complex ones, with many inputs, outputs and state variables, optimal decision in critical situations is not trivial and has to be based on scientific approaches from mathematical modeling, that enable to design optimal structure and parameters of the mathematical model.

E-mail address: jan.ciganek@stuba.sk

[1] http://www.fei.stuba.sk

Peer review under responsibility of Electronics Research Institute (ERI).

ELSEVIER Production and hosting by Elsevier

http://dx.doi.org/10.1016/j.jesit.2015.11.003

2. Problem formulation

The paper presents an effective robust fuzzy controller design procedure, which can be used in the control system of linear or nonlinear dynamic processes, which occur in many areas of industry, including robotic systems.

As will be shown in the case study, there is no need in any case to carry out demanding system identification. Therefore, it is not necessary to describe the controlled system by any of the usual forms (continuous or discrete transfer function, state-space model, differential equation, etc.).

For the design the fuzzy controller must take account of all the input and output variables that must be expressed by using the membership functions, and to choose a specific type, count and range of the membership function for each variable.

The most important step of the fuzzy controller design consists of drawing up the fuzzy rules including all the membership functions of selected input and output variables.

2.1. Fuzzy theory

Boolean logic is used to cover situations where it makes sense to think only of two options. However, there are many situations where we need to use some value other than 0 and 1, that is, when we do not know exactly the situation that we are trying to describe – determination of the value using binary logic is simply not sufficient. This issue deals with fuzzy logic, which allows describing the event using multiple value logic and is able to deal with the uncertainty that stems from the expression of the natural language. In everyday life situations we express the statements and conditions using natural language, and the same way we also understand the instruction and vague statements (Ackermann, 1993; Oppenheim and Schaffer, 1989).

The basic advantage of fuzzy logic is the ability to capture verbally expressed information by mathematic formulation. Fuzzy logic can work with ambiguous terms, often used in human speech.

The advantage of this technology is ability to find information based on inaccurate and incomplete data with possibility to find faulty and corrupted information. Fuzzy logic is often used in decision support systems, where the tasks consist of selecting the best alternative. Thus, the aim is to represent true values contained in the range complete truth to complete false (Ciganek and Kozak, 2008).

2.1.1. Fuzzy sets and membership functions

Under the concept of fuzzy set the mathematical apparatus that defines the terms of fuzzy sets and operations that could be executed with them can be understood. Fuzzy sets are used primarily to represent the linguistic values of the linguistic variables and are defined by their membership functions.

In fuzzy sets the affiliation of the element to fuzzy sets indicates the value of the membership function, which can take values from interval $\langle 0,1 \rangle$. The important idea is that this value does not represent the probability with which the element belongs to a fuzzy set, but rather it is the strength with which this element belongs to this fuzzy set (Schweizer and Sklar, 1983).

2.1.2. Fuzzy variables, rules and fuzzy systems

For fuzzy sets, unlike the classical set operations (AND, OR, NOT), there is a full spectrum of operations. Order to the vast amounts of aggregate operations brings the term of T-norm (the intersection of fuzzy sets) and T-conorm (the unification of fuzzy sets). For the negation of the operand fuzzy complement is used. The term of T-norm and T-conorm seems to be a bit unreasonable, but is generally used (Vysoky, 1996).

Both of these operations have a number of options for their evaluation, depending on different authors. By default the intersection is used in the T-norm and the unification in T-conorm.

Linguistic variables are used to create simple statements, which can be then coupled using the logical AND, OR operations into the complex statements. The following compound fuzzy statements created from simple fuzzy statements represent fuzzy rules.

The most common way to create fuzzy rules is IF-THEN conditions. Fuzzy rule consists of two parts: the antecedent is the conditional part of the rule and the consequent is the result part of the rule.

There are two basic types of fuzzy systems: Takagi-Sugeno-Kang and Mamdani. The difference between these types of fuzzy systems lies in the consequent section. In the system of the type Mamdani, the antecedent and consequent are represented by fuzzy sets. Therefore, to calculate the result in numerical form from the fuzzy form, a process of defuzzification is needed.

Sugeno type models have the consequent part (also the calculation of the result) in the form of the analytical function, which is the most common linear or constant. Whereas, in the case of the Sugeno type model is not required the defuzzification process, the usage of analytical functions gives directly a sharp value, thereby making the calculating process the higher speed.

2.2. Trajectory sampling algorithms

Let us consider the mechatronic system with the task of the trajectory tracking by the selected number of the trajectory points N and by the selected time horizon T.

Each point of the trajectory r_i; $i=1, 2, \ldots, N$ with associated time t_i is described by the vector of the appropriate coordinates $r_i = [x_i, y_i]$, in two-dimensional space and $r_i = [x_i, y_i, z_i]$ in three-dimensional space.

The following conditions have to be satisfied $t_1 = 0$ and $t_N = T$.

Given the fact that the case study is situated in a two-dimensional space, all of the following equations and algorithms will be presented (derived) for this system.

In order to obtain better performance (quality, stability) of the considered mechatronic system it is possible to use alternative sampling methods that take into account the need to increase the time of the controller in the critical working area.

In presented case study two critical situations are considered those could cause a great inaccuracy of the control loop-working area close to the initial point [0,0] and sharp change of the trajectory direction.

2.2.1. Fixed sampling period

There are many ways to set the sampling by trajectory tracking. The easiest possibility and also commonly used is the fixed sampling period where the time difference between each two following points of the trajectory is equal t_f:

$$t_{i+1} - t_i = t_f; \quad i = 1, \ldots, N-1 \tag{1}$$

where the value of fixed sampling period t_f is given by equation:

$$t_f = \frac{T}{N-1} \tag{2}$$

2.2.2. Sampling algorithm based on the distance to the initial point

The first algorithm is based on the distance of each trajectory point to the origin of the coordinate system (initial point). The closer the trajectory point to the origin is, the more time control circuit gets to achieve this point.

Let us find the maximal distance of the trajectory to the initial point [0,0].

$$d_M = \max_i \sqrt{x_i^2 + y_i^2; \quad i = 1, \ldots, N} \tag{3}$$

Parameter d_M is later used to calculate the total distance of all trajectory points to the maximal considered distance of the system:

$$\Delta_1 = \sum_{i=2}^{N} (c_1 \cdot d_M - \sqrt{x_i^2 + y_i^2}) \tag{4}$$

where the coefficient $c_1 > 1$ represents one degree of freedom (chosen by the user of this sampling method).

Variable sampling is calculated for each point of the trajectory using following algorithm:

$$t_i = t_{i-1} + \frac{c_1 \cdot d_M - \sqrt{x_i^2 + y_i^2}}{\Delta_1} \cdot T \tag{5}$$

while $t_1 = 0$, $t_N = T$.

In case of the value $c_1 \leq 1$ will be the presented algorithm unrealizable. In the most distanced trajectory point to the initial point will cause such parameter value a situation when $t_i \leq t_{i-1}$.

2.2.3. *Sampling algorithm based on the change of the trajectory direction*

The second algorithm is searching the points from where the change of the trajectory direction occurs. The sharper the direction change is the more time the control circuit gets to achieve the next point.

For the calculation of the angle in each trajectory point two vectors are used:

$$\left.\begin{aligned} \bar{a}_i &= (x_{i-1} - x_{i-2}, y_{i-1} - y_{i-2}) \\ \bar{b}_i &= (x_i - x_{i-1}, y_i - y_{i-1}) \end{aligned}\right\} ; \quad i = 3, \ldots, N \tag{6}$$

The equation for the angle of two vectors is in the form:

$$\gamma_i = a\cos\frac{\bar{a}_i \cdot \bar{b}_i}{\|\bar{a}_i\| \cdot \|\bar{b}_i\|} \tag{7}$$

The best value of angle is $\gamma_i = \pi\ rad$ (180°), i.e. without any change of the trajectory direction. The worst situation is by complete change of trajectory direction to the opposite one, i.e. the angle value is $\gamma_i = 2\pi\ rad$ (360°) or i.e. $\gamma_i = 0\ rad$ (0°).

Because of strong start from the initial point $r_1 = [0,0]$ to the second point of the trajectory r_2, let us consider the worst value of angle in this point: $\gamma_2 = 0$.

Let us use a linear function to evaluate the difference between the actual trajectory angle and the best value of angle:

$$y_i = k\,|\gamma_i - \pi| + q; \quad i = 2, \ldots, N \tag{8}$$

To find the values of parameters k and q, it is necessary to use the limit values of angle γ_i:

$$\begin{aligned} &\text{for } \gamma_i = \pi \rightarrow y_i = 1 \\ &\text{for } \gamma_i = \begin{cases} 0 \\ 2\pi \end{cases} \rightarrow y_i = c2 \end{aligned} \tag{9}$$

where the coefficient c_2 is the maximal allowed value of the linear function and represents one degree of freedom (chosen by the user of this sampling method).

Thus, the linear function is in the form:

$$y_i = \frac{c_2 - 1}{\pi}\,|\gamma_i - \pi| + 1; \quad i = 2, \ldots, N \tag{10}$$

The total sum of all values given by the linear function is:

$$\Delta_2 = \sum_{i=2}^{N} \left(\frac{(c_2 - 1) \cdot |\gamma_i - \pi|}{\pi} + 1 \right) \tag{11}$$

and variable sampling is calculated for each point of the trajectory using following algorithm:

$$t_i = t_{i-1} + \frac{((c_2 - 1) \cdot |\gamma_i - \pi|/\pi) + 1}{\Delta_2} \cdot T \tag{12}$$

while $t_1 = 0$, $t_N = T$.

2.2.4. *Combination of both sampling algorithms*

For the last sampling algorithm all the mathematical apparatus from two previous algorithms is used and the total sum was evaluated in the following form:

$$\Delta_{12} = \sum_{i=z}^{N} \left[(c_1 \cdot d_M - \sqrt{x_i^2 + y_i^2}) \left(\frac{(c_2 - 1) \cdot |\gamma_i - \pi|}{\pi} + 1 \right) \right] \tag{13}$$

Variable sampling is then calculated for each point of the trajectory, $i=2, \ldots, N$, using following algorithm (14):

$$t_i = t_{i-1} + \frac{(c_1 \cdot d_M - \sqrt{x_i^2 + y_i^2})((c_2 - 1) \cdot |\gamma_i - \pi| / \pi) + 1}{\Delta_{12}} \cdot T \tag{14}$$

while $t_1 = 0$, $t_N = T$.

2.3. Performance criteria

There are many different criteria to evaluate the performance of the control loop, mostly by comparing reference variables with simulated variables. The most used criteria are sum squared error and mean squared error, which are particularly suited for one-dimensional systems (Ciganek and Noge, 2013a,b).

It is possible to use modifications of those criteria in multidimensional systems. SDE presents the sum of the distance error between reference and simulated points of trajectory. MDE is the mean value of SDE criteria.

$$\mathrm{SDE} = \sum_{i=1}^{N} \sqrt{(x_{r,i} - x_{s,i})^2 + (y_{r,i} - y_{s,i})^2} \tag{15}$$

$$\mathrm{MDE} = \frac{1}{N} \sum_{i=1}^{N} \sqrt{(x_{r,i} - x_{s,i})^2 + (y_{r,i} - y_{s,i})^2} \tag{16}$$

where N represents number of trajectory points, $x_{r,i}$ and $y_{r,i}$ represent x- and y-coordinates of i-th point of reference trajectory, $x_{s,i}$ and $y_{s,i}$ represent x- and y-coordinates of i-th point of simulated trajectory.

It seems to be interesting to evaluate also the total reference trajectory length, which the control system tracked, in terms of evaluation control performance.

The following criterion represents error ratio of SDE to the length of passed trajectory d expressed in percentage.

$$\mathrm{PDE} = \frac{\mathrm{SDE}}{d} 100\,[\%] \tag{17}$$

3. Case study

Our simulated system contains two arms connected with rotational joint on their ends. One of these two arms has the opposite end free and the second arm has rotational joint with ground (frame) on the opposite end so that the marker on the free end is able to move in 2D coordinate system as dependence on angles of both arms. The action radius of the marker is equal to length of both arms (Fig. 1).

The object of interest is marker on the free end and its coordinates. Coordinates along x-axis and y-axis are output variables from this system (feedback variable in control). The system is moved by torques applied in joints, so that these torques are inputs in Adams model and at the same time also outputs from controller.

It means the task of the case study is to design such controller that will be able to keep tracking of the reference trajectory along both axes in 2D coordinate system.

Mentioned geometry is presented in Fig. 2. Mass properties and all construction parameters of this mechatronic system are in Table 1, and both arms have the same parameters.

3.1. Co-simulation

Co-simulation is based on exporting of the Adams model to Matlab-Simulink. To do this, you have to set up the input and output variables in Adams.

Keep in mind that input variables in Adams model are also the output variables of the controller in Matlab. So for better orientation, reference variables are coordinates along x- and y-axis by time variable. Input variables for controller are control deviations of both reference angles. It is necessary to use recalculation in order to obtain values of angles

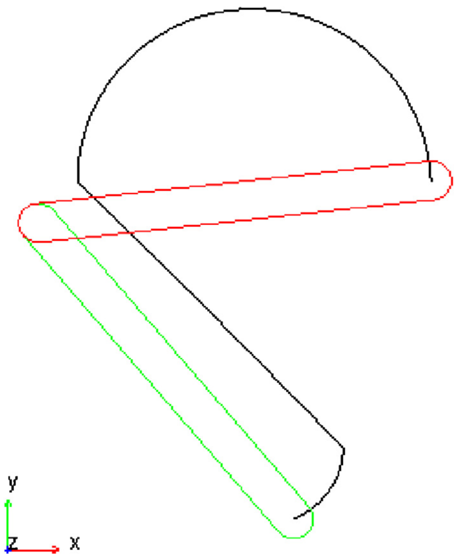

Fig. 1. Geometry of the mechanical system simulated in MSC Adams environment.

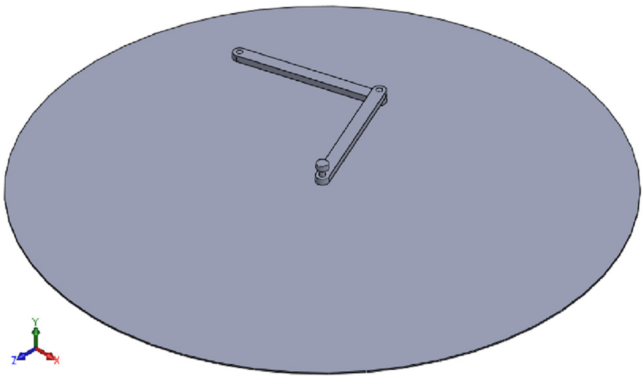

Fig. 2. Geometry of mechanical system in 3D isometric view.

from the reference coordinates. So, reference variable and feedback variable were positions in 2D coordinate system and then they have to be recalculated to angles, lately to control deviation of angles. Output variables of controller are also input variables to Adams environment. Output variables of Adams environment are real coordinates of simulated system.

Table 1
Construction parameters of mechanical system.

Parameter	Value	Unit
Length of arm (between key markers)	0.5	m
Width of arm	0.045	m
Depth of arm	0.0225	m
Material density	7801	kg m^{-3}
Mass of arm	3.83	kg
Moment of inertia about x axis of arm	7.57	kg m^2
Moment of inertia about y axis of arm	7.52	kg m^2
Moment of inertia about z axis of arm	7.96	kg m^2

Table 2
Fuzzy rules.

e\de	mf1	mf2	mf3
mf1	mf1	mf2	mf3
mf2	mf2	mf3	mf4
mf3	mf3	mf4	mf5
mf4	mf4	mf5	mf6
mf5	mf5	mf6	mf7

The recalculation from coordinates to angles is shown in Eqs. (18) and (19).

$$\varphi_1 = a\tan\left(\frac{y}{x}\right) - a\cos\left(\frac{\sqrt{x^2+y^2}}{2\cdot r}\right) \tag{18}$$

$$\varphi_1 = -2\cdot\left(90 - a\cos\left(\frac{\sqrt{x^2+y^2}}{2\cdot r}\right)\right) \tag{19}$$

where x and y are coordinates of the marker on the end of the second arm, r represents the length of one arm, and φ_1 and φ_2 are the angles of both arms.

3.2. Controller design

We decided to design a fuzzy controller in a feedback structure with four input and two output variables. Control deviation and the time derivation of the control derivation of two angles (manipulating with the arms) were used as inputs to fuzzy controller. Torques reacting in the rotational joints were used as outputs (Ciganek and Noge, 2013a).

Mamdani type was set as type of fuzzy controller. Five membership functions were used in the range ⟨−180°, 180°⟩ of gauss type for control deviation inputs. Three membership functions in range ⟨−35°/s, 35°/s⟩ of gauss type were used for derivations of control deviation. Seven membership functions of triangular type were used for outputs (torques). The range for the first angle is ⟨−50 N m, 50 N m⟩ and for the second angle is ⟨−25 N m, 25 N m⟩. Therefore 30 fuzzy rules in total were set to connect all membership functions. The combinations of 2 input variables for each arm are presented in Table 2.

Table 2 represents fuzzy rules as dependence of two inputs (control deviation, derivation of control deviation) and one output. There are no cross dependence of input signals and outputs, hence the rules for the second output is the same. It is possible to say, that this could be also done by two separate controllers, but the decision was to do it as one MIMO controller.

Fig. 3 is schematic of control loop. The reference trajectory is set by two lookup tables. The upper one represents the trajectory points along x-axis by time and the lower one along y-axis. The next blocks represent Matlab-function for recalculation of coordinates to reference angles of both robotic arms. Behind the evaluation of the control deviation

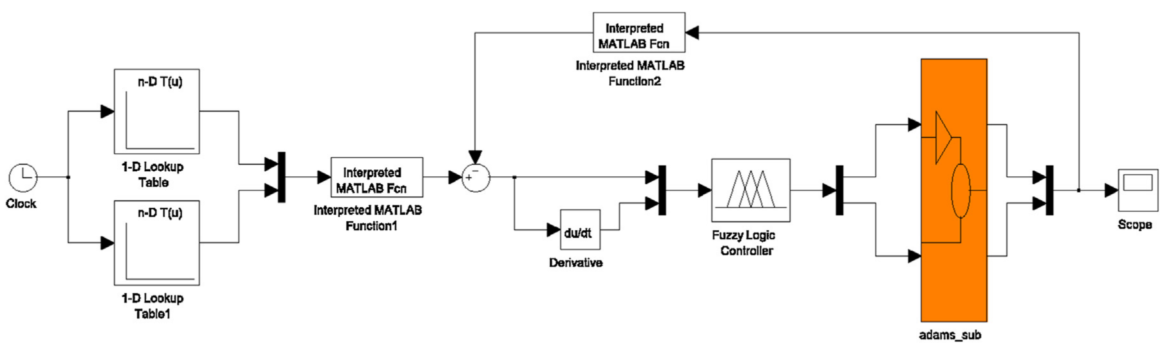

Fig. 3. Block scheme of control loop.

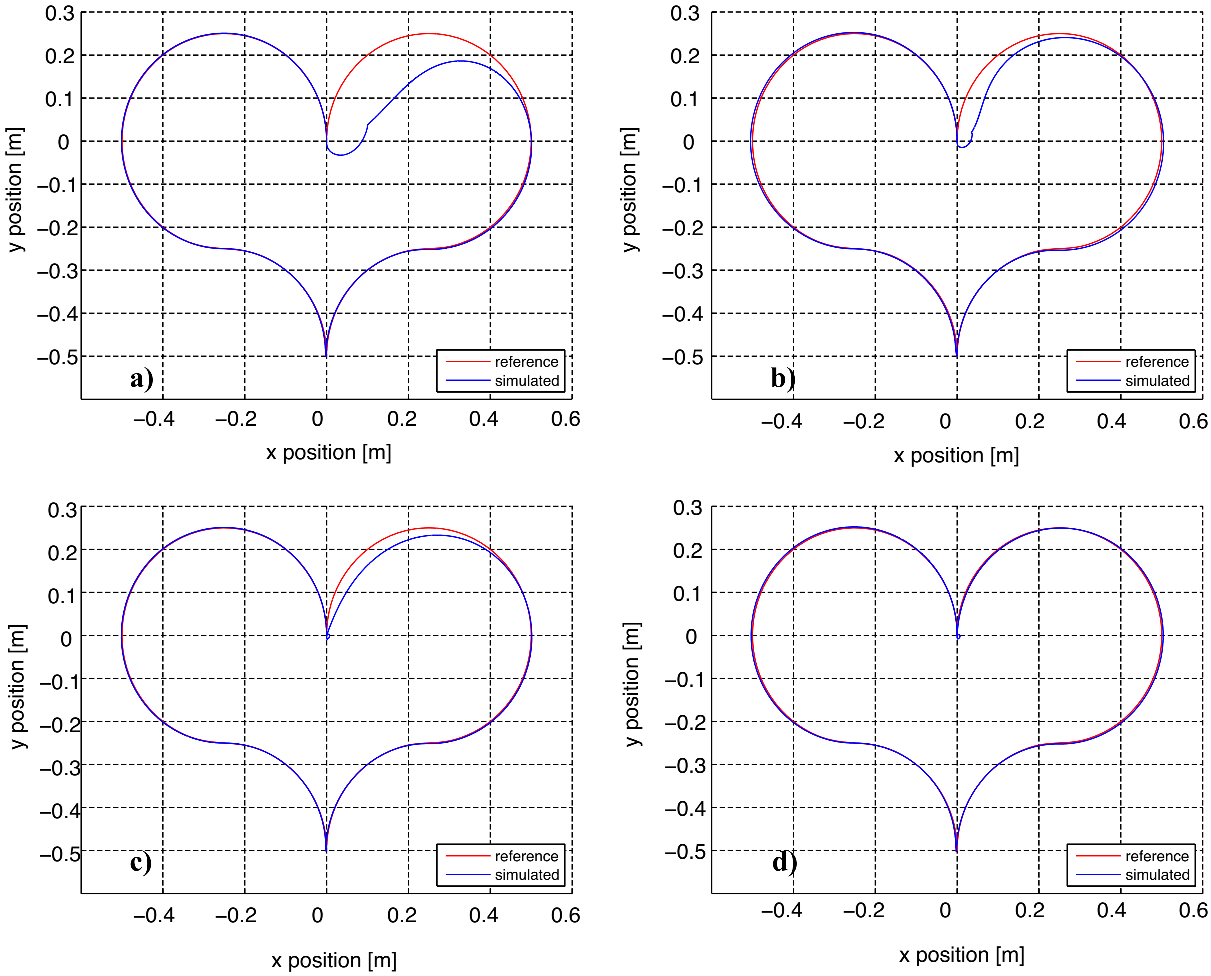

Fig. 4. Comparison of reference and simulated trajectory under fuzzy controller.

and its derivation for both angles the input signals to the fuzzy controller are led. The output signal from the controller consists of two parts – torques as control action for both robotic arms. The orange subsystem represents exported mechanical model from Adams environment. On the output of the Adams model the simulation coordinates of the marker on the end of robotic arm are situated.

3.3. Simulation results

In this part of paper suitability of designed fuzzy controller for two different types of trajectory is shown. The first trajectory (in the form of heart) consists of 4 circular parts and the second one (in the form of star) consists of 16 lines respectively. Both trajectories start and also finish at the same point – at the initial point [0,0].

In both trajectory examples the difference and improvement of the control performance using fixed sampling period and three types of algorithm for variable sampling of the trajectory points will be shown. The results are presented in graphical and also numerical form.

Reference trajectory in the form of heart is set by two vectors. Each of these vectors is containing 2884 sample points represented by coordinates along *x*- and *y*-axis. Simulation time for heart form is 40 s. Total tracked distance in this case is 3.14 m. Results are presented in Fig. 4. The control performance is evaluated by SDE, MDE and PDE criteria.

In Fig. 4a the trajectory is sampled by fixed step, with SDE = 0.77908, MDE = 2.7014e−04 and PDE = 23.80%.

In Fig. 4b the trajectory is sampled by the first algorithm, coefficient $c_1 = 1.1$, with SDE = 0.38564, MDE = 1.3372e−04 and PDE = 12.28%.

In Fig. 4c the trajectory is sampled by the second algorithm, coefficient $c_2 = 200$, with SDE = 0.21390, MDE = 7.4168e−05 and PDE = 6.81%.

In Fig. 4d the trajectory is sampled by combined algorithm, coefficients $c_1 = 1.35$ and $c_2 = 250$, with SDE = 0.07745, MDE = 2.6856e−05 and PDE = 2.47%.

4. Conclusion

This paper tries to point the importance of soft computing methods. Soft computing methods are successfully deployed in the field of mechatronic system control and as it is presented in case study, it is possible to obtain excellent results in terms of quality and stability. Our case study is oriented to fuzzy logic and variable sampling algorithms for trajectory tracking. We proposed robust fuzzy controller and variable sampling algorithms, which distribute points in time taking into account critical points. Results are evaluated by performance criteria and there was great improvement. Next step in the future might be combination of fuzzy logic and neural networks (called ANFIS), where is presumed obtaining even better results in terms of observed criteria.

The presented results have been reached by co-simulation of Matlab-Simulink and MSC Adams. Dynamic model in MSC Adams was cross connected with control loop created in Matlab-Simulink.

Acknowledgement

This paper is supported by APVV project No. APVV-0772-12.

References

Ackermann, J., 1993. Robust Control: Systems with Uncertain Physical Parameters. Springer-Verlag, London, UK.

Ciganek, J., Kozak, S., 2008. Robust controller design using algebraic theory. In: Int. Conf. Cybernetics and informatics, Zdiar, Slovak Republic.

Ciganek, J., Noge, F., 2013a. Fuzzy Logic Control of Mechatronic Systems. In: IEEE Int. Conf. on Process Control, Strbske Pleso, Slovak Republic.

Ciganek, J., Noge, F., 2013b. Fuzzy Logic Control of Robotic Arm. In: The 22nd International Workshop on Robotics in Alpe-Adria-Danube Region, Portoroz, Slovenia.

Oppenheim, A.M., Schaffer, R.W., 1989. Discrete-Time Signal Processing. Prentice-Hall, Englewood Cliffs.

Schweizer, B., Sklar, A., 1983. Probabilistic Metric Spaces. North-Holland, New York.

Vysoky, P., 1996. Fuzzy Rizeni. CVUT, Praha, ISBN: 80-01-01429-8.