Computing and Informatics, Vol. 32, 2013, 487–507

ROBUST FIXED POINT TRANSFORMATIONS-BASED CONTROL OF CHAOTIC SYSTEMS

Teréz A. Várkonyi

Doctoral School of Applied Informatics
Óbuda University
96/B Bécsi út., Budapest, H-1034, Hungary
&
Dipartimento di Informatica
Università degli Studi di Milano
7 Via Matilde di Canossa Crema (CR) 26013, Italy
e-mail: varkonyi.teri@phd.uni-obuda.hu

József K. Tar, Imre J. Rudas

Institute of Applied Mathematics
John von Neumann Faculty of Informatics
Óbuda University
96/B Bécsi út, Budapest, H-1034, Hungary
e-mail: tar.jozsef@nik.uni-obuda.hu, rudas@uni-obuda.hu

Abstract. Nowadays, nonlinear control is a very important task because machines are playing an ever increasing role in life. Lyapunov's 2^{nd} method is a popular tool by the use of which various controllers can be designed like adaptive neural networks, fuzzy controllers, and neuro-fuzzy solutions, or the sliding mode controllers and the well-known PID feedback controllers. Robust Fixed Point Transformation is a procedure which can be built for almost any type of controller in case an approximate model is used to estimate the controlled system's behavior. In this paper, a new approach to Robust Fixed Point Transformations (RFPT) is introduced by integrating a second controller in the system. Authors show that this additional, "recalculated" controller not just improves the original controller's results, but halves the tracking errors achieved by the previous RFPT methods.

Keywords: Robust fixed point transformations, Duffing system, nonlinear control, adaptive control, chaos synchronization

Mathematics Subject Classification 2010: 34-H05, 34-H10, 49-J15, 49-K15, 58-E25, 62-F35, 70-Q05, 93-B52, 93-C10, 93C15, 93-C40

1 INTRODUCTION

Nowadays, improving an existing system's results (speed, accuracy, efficiency, etc.) is very important since there is competition between the manufacturers of machines, like industrial machines, cars, public vehicles, robots and labor-saving devices. Because of this competition the factories are more and more supervised and they have to measure up more expectations. To live up these expectations the manufacturers need and vindicate scientific research to be able to increase productivity, but they also use the knowledge to improve e.g. their direction. For example, manufacturing is getting so complex that it can be handled decreasingly without workflow. Workflow is a fast developing method that is used to model complex systems and it is very useful to organize manufacturing [1, 2].

Another expectation for factories is to make their operation more secure and predictable. There are occasions when the machines do not respond in a prescribed way because of an unwanted or unexpected circumstance they get in (like car skidding on ice). The undesired behavior can cause major problems, so it has to be terminated, and also examined to be able to avoid it later. For examination the use of chaos can be very helpful. Since chaos can be the representative of many real situations and it is not predictable, it can test the tolerance level of machines. Chaos also helps to solve many other problems appearing in other fields of industry and also of real life.

The last rising expectation from manufacturers we mention is accuracy. Because of automation, exactitude is crucial to obtain products of high quality standards. There are many opportunities how to improve the accuracy of an industrial robot, one of them being to ameliorate the applied controllers. But the question of money is raised every time. Replacing an existing controller with a better one is sometimes very expensive; so cheap modifications can bring higher financial benefit. Scientists also realize this monetary problem: many papers can be found in the literature how to upgrade existing controllers' results without modifying them considerably.

One of the most popular controllers used in industry is the PID controller. It is effective in many cases, and it can be handled easily. Unfortunately more difficulties and uncertainties come up as the industry grows which makes the present controllers' job more difficult. Many problems appear which cannot be solved via simple PID controllers; so new algorithms are needed to make the PID controllers more effective [3, 4, 5]. The same necessity for extensions exists for the other types of controllers. There are different approaches introducing new control strategies or improving the behavior of an already implemented control technique, like integrating fault tolerance in the design methodology [6, 7], performance enhancement of controllers [8, 9], integration of soft computing [10, 11, 12], or designing improved model-based controllers [13].

One of the possible ameliorating methods is the so-called Robust Fixed Point Transformation (RFPT) [14] which can improve existing controllers' results without too many modifications in the actual system. It is based on the idea that if an approximate model is used to predict the controlled system's behavior, there is a function which can reduce the disadvantages of the approximation. As included in its name, RFPT is characterized by robustness and it has the ability to handle rough approximations without increasing the controller's computational burden considerably.

One of the great advantages of RFPT is that it can be built together with almost any type of controller, for example neural network-based controllers, fuzzy controllers [15], sliding mode controllers, or the well known proportional-integral-derivative (PID) controllers [16].

In this paper, a new approach of Robust Fixed Point Transformations is introduced. The new method is based on the idea of integrating a second controller into the system. This second controller "recalculates" the results got from a previous modus of RFPT. As mentioned, RFPT can improve existing controllers' results, it can lower the tracking error achieved by a traditional controller, in most cases by two orders of magnitude. Thanks to the built-in second controller, the new approach achieves additional tracking error reduction. Ideally, the abatement is about 50 %.

The paper is organized as follows: In Section 2, the chaos synchronization and the Duffing system are introduced. In Section 3, the classical feedback control and the RFPT is explained as a basis of the novel part of the paper. Section 4 introduces a new approach to Robust Fixed Point Transformations. The simulation results are shown in Section 5. In the last section, the conclusions are summarized.

2 CHAOS SYNCHRONIZATION AND THE DUFFING SYSTEM

Chaos is a very common phenomenon when scientists have to deal with unstable systems or biological processes. In many cases, chaos represents an undesired and disadvantageous behavior of a system. For example, if a sliding mode controller causes a system to chatter, it is not a wanted phenomenon. In some cases, the stress level caused by the chatter can actually damage the system itself. So usually it is worthy to prevent chaos. However, chaos may also have useful applications in which for example the chaotic motion details are of interest. One of these applications is chaos synchronization.

Chaos synchronization is a very important part of chaos theory. It has the same goal as dynamical systems theory: it deals with dynamical systems. It is applied in numerous areas not just to supervise natural processes but to test the effectiveness of a controller.

Most of the approaches to chaos synchronization are based on the use of two systems, a master and a slave one. Usually the systems are not identical. The slave system's task is to follow the master's suit. A controller calculates the control signal which takes the slave system to the desired state.

The systems are usually some kind of dynamical systems that show chaotic behavior. Depending on which field we want to study, different attractors can be utilized. For example the Chua circuits [17] are used in electricity, Rössler's attractor [18] is applied in chemistry, the Lorenz systems [19] which was developed to model atmospheric convection and the Duffing systems [20] to model oscillators. Since the Duffing system is more approachable mechanically, the authors find this attractor more usable than the others. In the following, the Duffing attractor is introduced in details.

The Duffing Equation was invented by Georg Duffing in 1918 [20]. His aim was to model certain driven and damped oscillators (for example a spring pendulum). Then his equation was extended to build up a system of first order differential equations. It is called Duffing system. One of the advantages of Duffing system is that it shows chaotic behavior; so it can be utilized for chaos synchronization.

If the above explained master and slave systems are both Duffing systems, then the equations describing them are as follows:

$$\begin{array}{rcl} \dot{x}_1 & = & x_2 \\ \dot{x}_2 & = & -\delta_1 x_2 + \alpha_1 x_1 - \beta_1 x_1^3 + a \cos \omega t + d_1 \end{array} \tag{1}$$

$$\begin{array}{rcl} \dot{y}_1 & = & y_2 \\ \dot{y}_2 & = & -\delta_2 y_2 + \alpha_2 y_1 - \beta_2 y_1^3 + a \cos \omega t + u + d_2 \end{array} \tag{2}$$

where the components x determine the state of the master system and the components y denote that of the slave system.

Here x_1, and y_1 denote the displacements. Their first derivatives (x_2 and y_2) mean velocity, and the second derivatives are responsible for acceleration. The term $a \cos \omega t$ determines the external driving force exciting the chaotic motion of the two systems, where a marks the amplitude and ω is the frequency; the terms x_1^3 and y_1^3 stand for the non-linearity in the restoring force; αs (restoring force), βs (amount of non-linearity in the restoring force), and δs (damping) are also free parameters; u denotes the control force. The d_1 and d_2 components can be interpreted as "disturbance forces".

3 CLASSICAL FEEDBACK CONTROL AND THE BASICS OF RFPT

In this section, authors review those parts of the science that substantiate the novel part of the paper (see Section 4). First the classical feedback control is explained, then the basics of RFPT are shown.

3.1 The Expected-Observed Response Scheme

Usually the classical feedback control tasks are built as follows. There is a prescribed or "desired" behavior r^d for an existing system (in our case, the master system generates r^d, and the existing system is the slave system). The existing system has some kind of "excitation", for example some kind of torque or a control signal u which forces the system to produce the desired response. Different forces (gravity, friction, sometimes an accelerating motor, disturbance, etc.) take effect on the system. The actual value of the control signal has to be calculated with respect to these forces.

In this case, the control task can be formulated by an equation $r^r = \varphi(u)$ which describes the correspondence of the control signal and the actual response r^r of the system (after applying u on it). The problem is that usually the controlled systems are not exactly known. For the proper control signal computation ($u^d = \varphi^{-1}(r^d)$) just approximate models can be used: $u^d_{appr} = \varphi^{-1}_{appr}(r^d)$. This problem causes the main errors in the control tasks since the controllers do not take the approximation into account. The desired control force for the system is achievable only with exact inverse model. So applying this approximate control signal to the system, it gives the realized response. The correspondence between the realized (r^r) and the desired response (r^d) is $r^r \equiv \varphi(\varphi^{-1}_{appr}(r^d)) \equiv f(r^d) \neq r^d$. Since the controlled system is unknown, we cannot determine function $f = \varphi^{-1}_{appr} \circ \varphi$ either. All we can do is to measure its output.

3.2 The Basics of RFPT

As mentioned above, it is hard to find the proper (desired) control signal u^d for an unknown system since we cannot predict the system's behavior. Using an approximate model may help to roughly determine the control signal (u^d_{appr}), but it is not always enough. Extra calculations are needed.

Attached to RFPT, there are two options how to put the above explained deficiency to rights. On one hand it is an option to construct a function G_1 which maps u^d_{appr} closer to u^d: $|G_1(u^d_{appr}) - u^d| < |u^d_{appr} - u^d|$. On the other hand it is possible to construct a function G_2 which transforms r^d closer to r_* so that $\varphi^{-1}_{appr}(r_*) = u^d$ and $|G_2(r^d) - r_*| < |r^d - r_*|$.

In [21], the authors show an iterative fixed point searching algorithm. They prove that by fulfilling two simple conditions, 1. G (where G denotes G_1 or G_2) is continuous and 2. $|G'| \leq M < 1$ ($M \in \mathbb{R}$), the sequence $\{u_0, u_1 = G(u_0), u_2 = G(u_1), \ldots, u_{n+1} = G(u_n)\}$ is convergent and the fixed point of G is $\{u_n\}$'s limit value. The proof is a sequence of equivalent steps. It means that if the two constraints are true and the fixed point of G_1 is u^d (or the fixed point of G_2 is r_*), then the sequence constructed by it will converge to u^d (or r_*). In this paper, the function we use for G_1 is what Tar suggested in [21] for an RFPT-based Model Reference Adaptive Controller (MRAC):

$$G_1\left(u, u^d_{appr}\right) = (u + K)\left(1 + B \tanh\left(A\left(h(u) - u^d_{appr}\right)\right)\right) - K \tag{3}$$

where

- $\varphi^{-1}_{appr}(\varphi(x)) = h(x)$

- $h\left(u^d\right) = u^d_{appr}$
- $G_1'\left(u, u^d_{appr}\right) = \frac{(u+K)ABh'(u)}{\cosh^2\left(A\left(h(u)-u^d_{appr}\right)\right)} + 1 + B\tanh\left(A\left(h\left(u\right) - u^d_{appr}\right)\right).$

The function for G_2 is also proposed by Tar in [14] for the RFPT-based PD controller:

$$G_2\left(r, r^d\right) = \left(r + K\right)\left(1 + B\tanh\left(A\left(f\left(r\right) - r^d\right)\right)\right) - K \qquad (4)$$

where

- $\varphi(\varphi^{-1}_{appr}(x)) = f(x)$
- $f\left(r_*\right) = r^d$
- $G_2'\left(r, r^d\right) = \frac{(r+K)ABf'(r)}{\cosh^2\left(A\left(f(r)-r^d\right)\right)} + 1 + B\tanh\left(A\left(f\left(r\right) - r^d\right)\right).$

Since the properties of these functions (from this point we allude just to the first one) refer to the novel part of the paper (see Section 4), the authors feel necessary to share it here. In the equations, A, B, and K are free parameters. They can be chosen so that the necessary limitation $\left|G_1'\left(u, u^d_{appr}\right)\right| < 1$ (or $\left|G_2'\left(r, r^d\right)\right| < 1$) is guaranteed. It has two fixed points: u^d (or r_*) and $-K$. The latter can be easily excluded because the difference between the two fixed points is measurable only in orders of magnitude (in addition $-K$ is known). This means that if G_1 (or G_2) is flat enough around u^d (or r_*), the iteration will converge to it, so $G_1(u, u^d_{appr})$ (or $G_2(r, r^d)$) will be closer to u^d (or r_*) than u^d_{appr} (or r^d). G_1 and G_2 are robust with respect to variation of the system (formulated by φ). This robustness is a consequence of the strong nonlinear saturation of the sigmoid function tanh(), and can be investigated approximately by the use of an affine approximation of $\varphi(G_1(\varphi^{-1}_{appr}(x)))$ (or $\varphi(\varphi^{-1}_{appr}(G_2(x))))$ in the vicinity of u^d (or r_*). The iteration generated by G_1 and G_2 converge at a considerable speed even nearby their fixed point. Because of their robustness, the function φ has less influence on their behavior.

In the iteration, $h(u)$ and $f(r)$ can be calculated only with one step delay. It means that two of G_1's and G_2's inputs are got from the previous step (u and $h(u)$, or r and $f(r)$). Because of this delay, if u^d or r_* varies quickly, then thanks to the shift, u^d_{appr} or r^d can get out from interval where G_1 and G_2 converge to the right fixed point. This means a possible instability; however, the latest research shows that this disadvantage can be fixed by a fuzzy-like parameter tuning [22], and a VS-type stabilization algorithm [23], but they are not used in this approach.

So if we assume that u^d or r_* varies slowly, then $G_1(u, u^d_{appr})$ or $G_2(r, r^d)$ is a proper choice. In practice, the smaller A is, the wider "window" we get where function G_1 converges to u^d and G_2 converges to r_* (instead of $-K$). After setting A, the better value we find for K, the quicker convergence we gain (these parameters can be set by trial and error). After a few simulations the orders of magnitude of the desired and simulated responses are observable. A and K can be set accordingly (B is always ±1, K is a very big negative and A is a very small positive number).

Furthermore, if tanh is not suitable, it can be replaced by any bounded, strictly monotone increasing differentiable $\sigma(x)$ function that fulfils the property $\sigma(0) = 0$, e.g. $\sigma(x) = x/(1 + |x|)$.

4 THE RFPT-BASED "RECALCULATED" PD CONTROLLER

In Section 3, the authors reviewed two options how to build in a simple "deformer" function (G) into a controller so the system gives more accurate response. In the following, on the basis of the previous results, the authors introduce a new and more effective possibility how the system's results can be improved so the tracking error can be decreased significantly.

The proposed idea is based on the theory that in our case, the feedback control has three main tools: a controller ($PD()$), an approximate inverse model ($\varphi_{appr}^{-1}()$), and the system itself ($\varphi()$). In the previous two methods, the authors showed how to build in the improver function between the model and the system [21]

$$\varphi(G_1(\varphi_{appr}^{-1}(PD(r_n^r)))) = r_{n+1}^r \tag{5}$$

and between the controller and the model [14]

$$\varphi(\varphi_{appr}^{-1}(G_2(PD(r_n^r)))) = r_{n+1}^r. \tag{6}$$

Now the authors introduce a new structure to Robust Fixed Point Transformations: how to include the improver function between the system and the controller:

$$\varphi(\varphi_{appr}^{-1}(PD(G_3(r_n^r)))) = r_{n+1}^r \tag{7}$$

In details, based on the logic given in the previous section the goal is to find the function G_3, which maps r^d closer to some r_G^* where $PD\left(r_G^*\right) = r_*$ ($\varphi_{appr}^{-1}\left(r_*\right) = u^d$ and $PD()$ denotes the PD controller) so that $|G_3(r^d) - r_G^*| < |r^d - r_G^*|$. In effect, this means that we have to "lie" to the controller about where the proper place for the system is. So we force it to map its inputs to somewhere else.

The iterative fixed point searching algorithm explained in Subsection 3.2 is adaptable here, too. Based on the logic of (3) and (4), the following function is proposed:

$$G_3\left(PD(r), r^d\right) = \left(PD(r) + K\right)\left(1 + B\tanh\left(A\left(f\left(PD(r)\right) - r^d\right)\right)\right) - K \tag{8}$$

where

- $\varphi(\varphi_{appr}^{-1}(x)) = f(x)$
- $f\left(PD(r_*^G)\right) = r^d$
- $G_3'\left(r, r^d\right) = \frac{(r+K)ABf'(r)}{\cosh^2\left(A\left(f(r)-r^d\right)\right)} + 1 + B\tanh\left(A\left(f\left(r\right) - r^d\right)\right).$

This function has the same conditions as (3) and (4), which means that it has just three free parameters, it is very robust with respect to the disturbances, it converges with considerable speed, but it also has the shift inside, so $G_3(PD(r), r^d)$ is a proper choice only if r^d varies slowly.

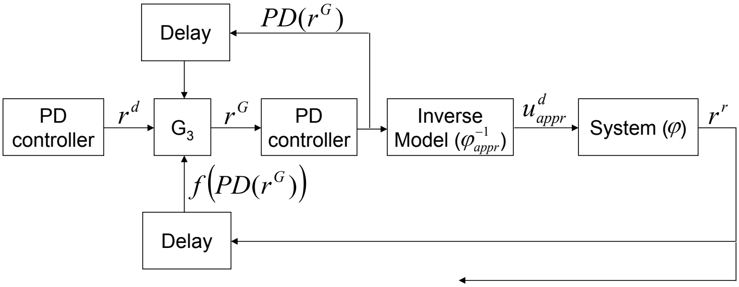

Fig. 1. Block diagram of the RFPT-based "recalculated" controller scheme. An extra controller is added which causes further decrease in the tracking error.

The system with the RFPT-based "recalculated" PD Controller is shown in Figure 1. The logic of the new structure is very similar to that of the RFPT-based traditional PD controller (4), but there is a significant difference: we do not just "deform" r^d, but we calculate a new desired response with the help of a second controller. This new desired response is calculated from the deformed desired response (deformed by G_3), so the inverse model approximation is taken into account. The recalculation of the desired response is made by a second controller, which (in most cases) additionally reduces the tracking error. The drawback of the extra controller is the extra money we have to invest and the extra computational time required. In this paper, a PD-type controller is built in additionally, which means that the extra need of time is negligible and the cost is acceptable (between $ 30 and $ 80).

In our example we use the same controllers, but ad-libitum, they can be different (in this case the improvement depends on the "weaker" controller).

5 SIMULATION RESULTS

As mentioned earlier, to illustrate the efficiency of the new approach of RFPT, in this section we show some simulation results made on chaotic systems. The task is to synchronize two nonlinear Duffing systems that are not identical.

The simulations are made using the MATLAB-Simulink pair. The programs use a solver for ordinary differential equations (ode45). The solver's integration method is automatically set by the softwares depending on stiffness of the problem. The tracking error is strongly related to the integrator's absolute tolerance. We set the tolerance organically in every case (10^{-3}) to be able to compare them. The maximum step size of the integration is also the same (10^{-3}) in every simulation.

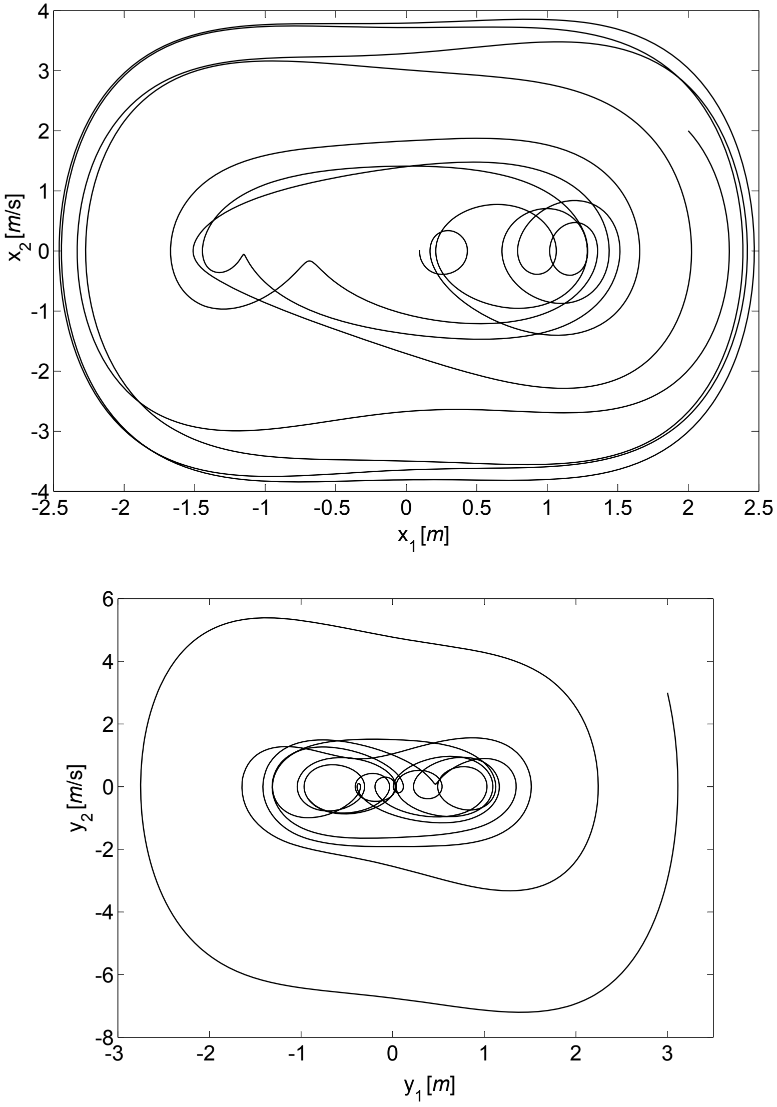

Fig. 2. Chaotic and non synchronous motion of the systems without active control; top: master system, bottom: slave system; x_1, x_2, y_1, y_2: state variables

The parameter values for the two Duffing systems are set to $\alpha_1 = 1N$, $\alpha_2 = 0.8N$, $\beta_1 = 1\frac{N}{m^2}$, $\beta_2 = 1.5\frac{N}{m^2}$, $\delta_1 = 0.2N$, $\delta_2 = 0.3N$, $\omega = 2Hz$, and $a = 1.2Nm$. In our case, the approximate inverse model has the same structure as the systems Equations (1) and (2), but different parameters are assumed: $\hat{\alpha} = 1.5N$, $\hat{\beta} = 0.5\frac{N}{m^2}$, and $\hat{\delta} = 0.1N$. In the simulations, the initial values are $y_1 = 3m$, $y_2 = 3m/s$, $x_1 = 2m$, and $x_2 = 2m/s$. In the sequel, simulation results are presented for the above parameter settings. For the tracking error relaxation the following PD-type equation is used:

$$\dot{y}_2^{Des} = \dot{y}_2 + 2\Lambda(x_2 - y_2) + \Lambda^2(x_1 - y_1) \tag{9}$$

where $\Lambda = 5/s$.

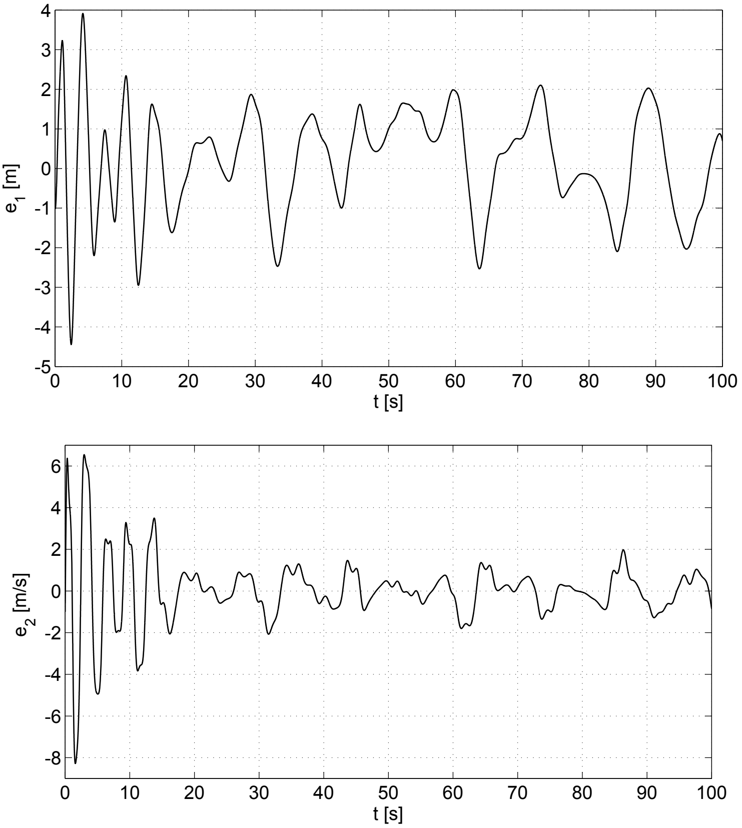

Fig. 3. The tracking errors of the state variables of the slave system (top: $e_1 = x_1 - y_1$, and bottom: $e_2 = x_2 - y_2$) without control

In the first step, the two systems are presented: Figure 2 shows the chaotic behavior of them. Figure 3 displays the tracking errors ($e_1 = x_1 - y_1$ and $e_2 = x_2 - y_2$)

without control. The figures show that there is a significant difference between the two systems.

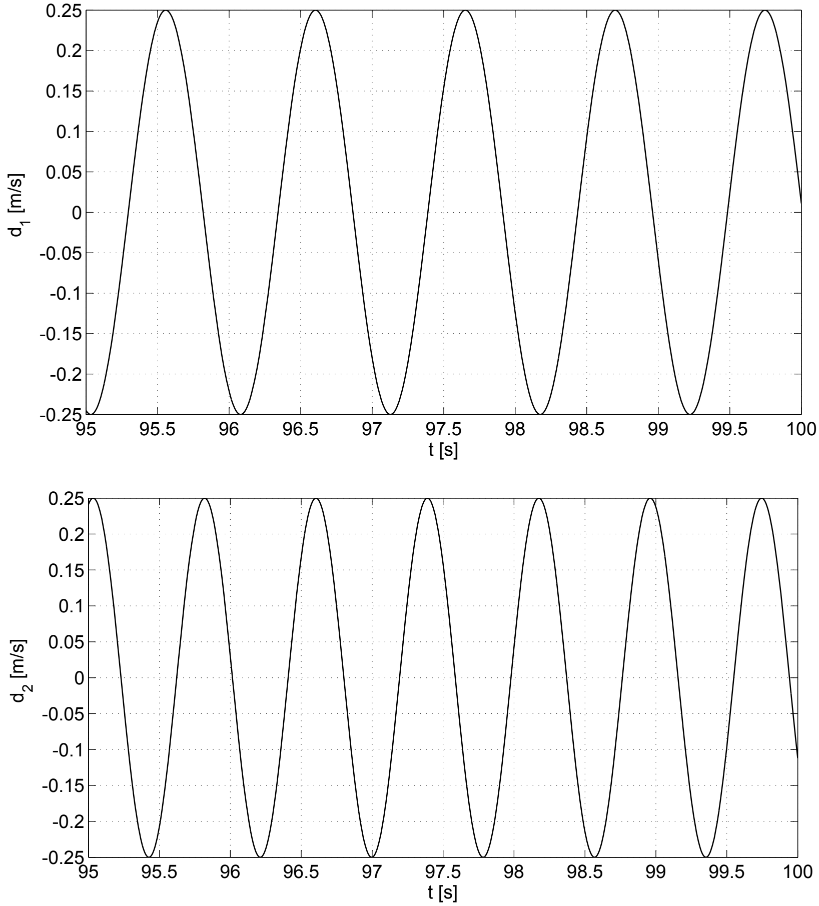

Fig. 4. The disturbance forces applied on the master (top) and the slave (bottom) systems (defined in Equations (1)–(2))

The simulations were made both with and without disturbance. The tanh() function makes the RFPT-based controllers very robust, so the disturbances do not increase the tracking errors' order of magnitude in the RFPT-based simulations. That is why the authors do not feel it is necessary to show the "disturbed" results in this paper. The disturbing sine waves are presented in Figure 4.

In the next step, simulation results of the three different controllers are shown without the RFPT extensions. Figures 5 and 6 illustrate the tracking errors of the first ($e_1 = x_1 - y_1$) and second ($e_2 = x_2 - y_2$) state variables. The figures reveal that the Model Reference Adaptive Controller cannot achieve synchronization. The other two controllers are successful, but the proposed new "recalculated" structure gives more accurate results than the traditional one (thanks to the second controller).

The differences between the desired and realized responses without RFPT are illustrated in Figure 7. The MRAC version predicts the failure of the synchronization at the early stage of the simulation when the tracking errors are "small" yet. The "recalculated" structure achieves most accurate results again.

In the last step, simulation results are shown with RFPT. The values of the free parameters of G_1, G_2, and G_3 are marked in Table 1. Figures 8 and 9 illustrate the tracking errors of the first and second state variables. The figures reveal that the RFPT-based traditional PD controller lowers the tracking error by more than two orders of magnitude. The RFPT-based MRAC now succeeds and generates similar tracking error like the RFPT-based traditional PD. The proposed "recalculated" controller lowers the error by additional 50 % compared to the other methods.

	A	B	K
G_1	2×10^{-5}	-1	70 000
G_2	10^{-2}	1	-100
G_3	10^{-2}	1	-100

Table 1. The values of the free parameters of G_1, G_2 and G_3

The differences between the desired and realized responses in the RFPT-based case can be seen in Figure 10. The extended traditional PD and MRAC approaches result in similar errors. The proposed "recalculated" controller halves the error here too.

6 CONCLUSIONS

The method of Robust Fixed Point Transformations is often applied to improve existing and well behaving controllers' results in case an approximate model is used in the control task. However, the method raises the question of stability; the recent research shows that stability is reachable if RFPT is used. In this paper, a new approach to Robust Fixed Point Transformations is introduced. The approach is based on the idea of integrating a second controller into the system. The great advantage of the second controller is that the proposed new RFPT-based "recalculated" controller halves the tracking error achieved by the previous methods of RFPT. However, it gives the burden of increased computational time, but in case of simple controllers, like the PD-type ones, the drawback is insignificant.

Acknowledgement

This work was supported by Óbuda University and the National Development Agency using the Hungarian National Scientific Research Fund in Project OTKA CNK 78168. Teréz A. Várkonyi also wishes to thank to Professor Vincenzo Piuri for his advices during the work.

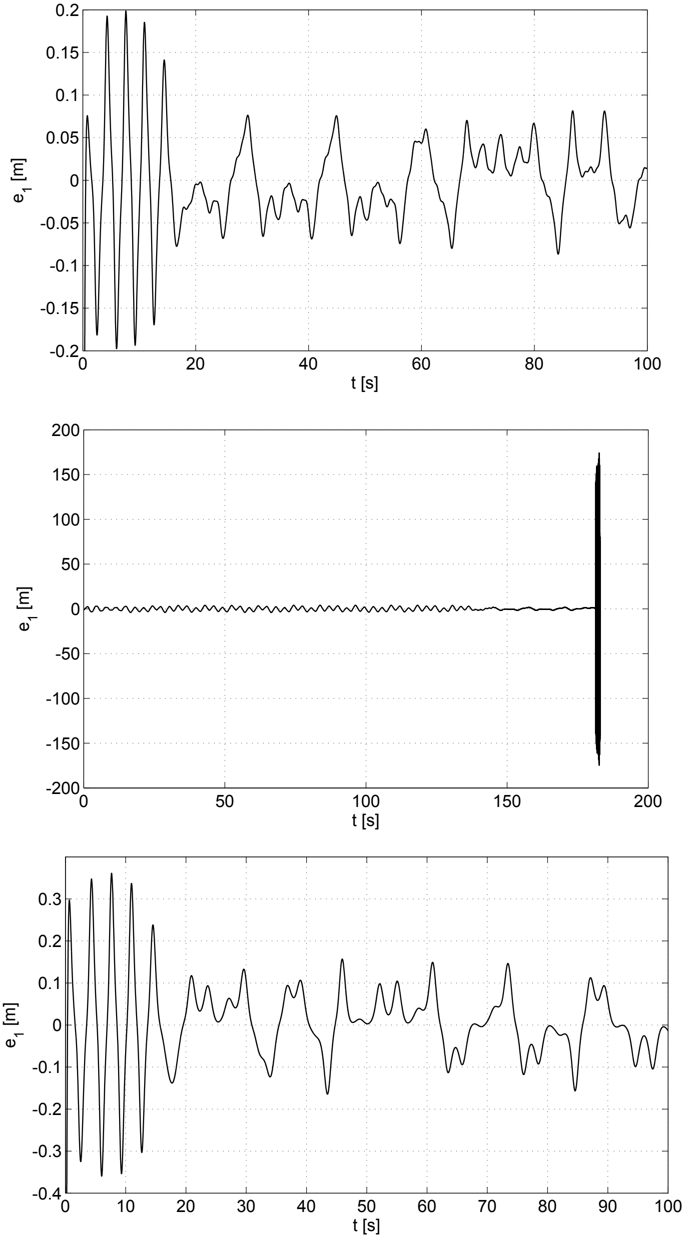

Fig. 5. The tracking errors of the first state variables of the slave systems without applying RFPT ($e_1 = x_1 - y_1$); top: "recalculated" PD, middle: MRAC, bottom: traditional PD. As it is shown, the MRAC fails. The traditional PD succeeds, but does not accomplish as well as the "recalculated" PD which has two PD controllers

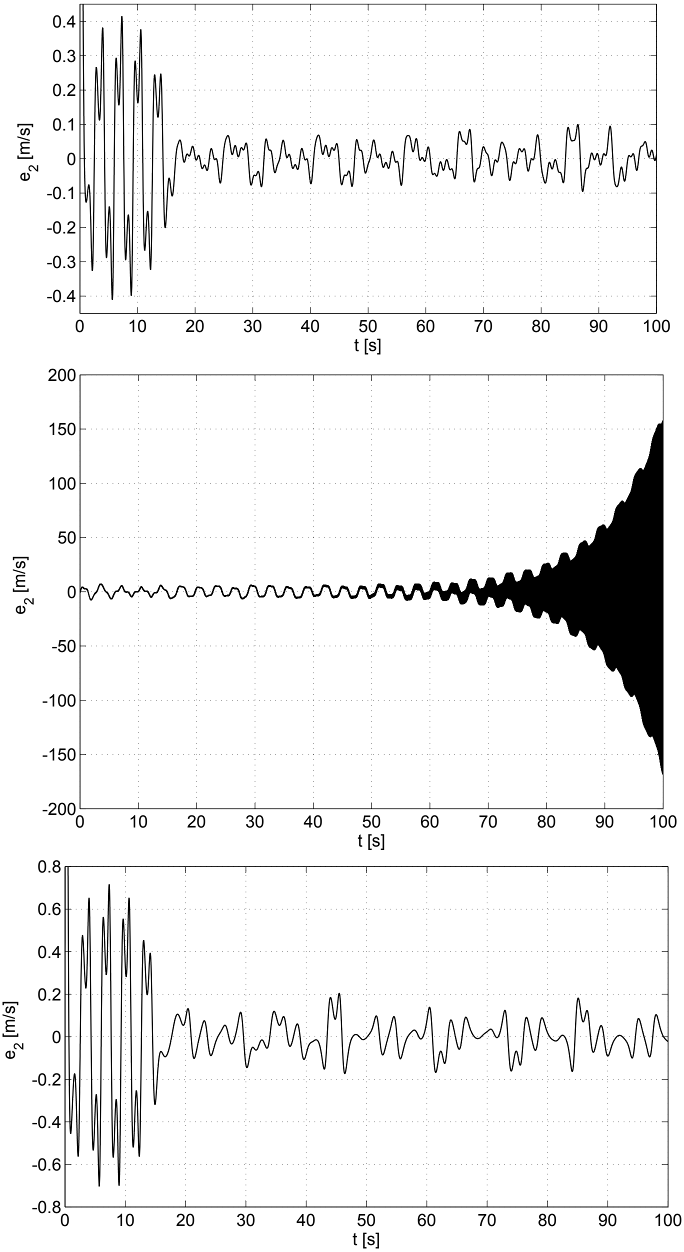

Fig. 6. The tracking errors of the second state variables of the slave systems without applying RFPT ($e_2 = x_2 - y_2$); top: "recalculated" PD, middle: MRAC, bottom: traditional PD. As it is shown, the MRAC fails. The traditional PD succeeds, but does not accomplish as well as the "recalculated" PD which has two PD controllers

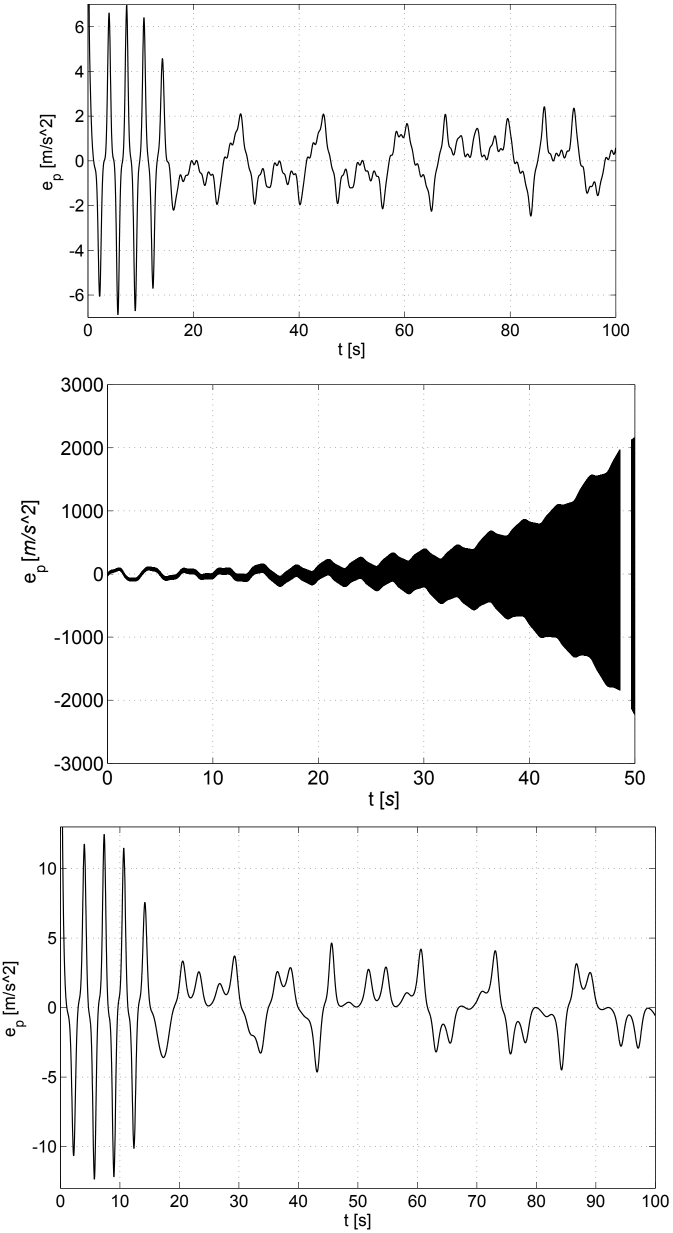

Fig. 7. The difference between the desired and the realized response ($e_p = \dot{y}_2^d - \dot{y}_2^r$) without applying RFPT ($e_2 = x_2 - y_2$); top: "recalculated" PD, middle: MRAC (0–50 seconds), bottom: traditional PD. As it can be seen, The MRAC's predicts the failure in an early stage. The traditional PD succeeds, but not as well as the "recalculated" PD which has two PD controllers

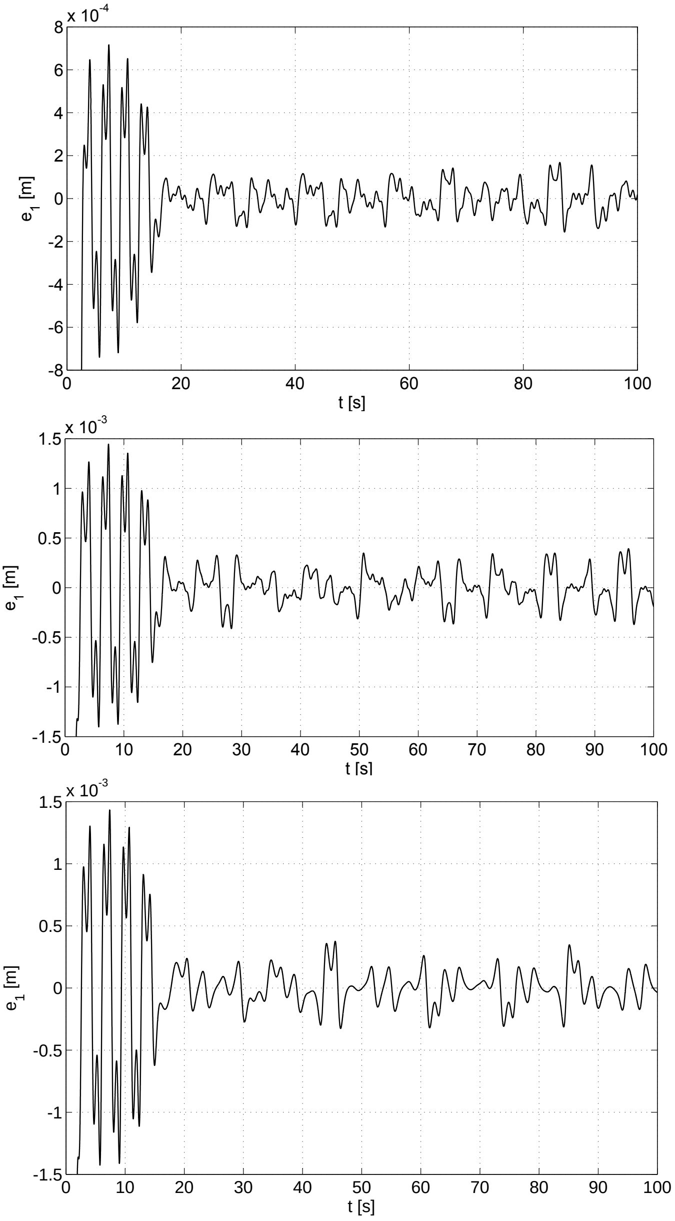

Fig. 8. The tracking errors of the first state variables of the slave systems with RFPT ($e_1 = x_1 - y_1$); top: "recalculated" PD, middle: MRAC, bottom: traditional PD. It can be seen well they all reduce the tracking error by more than two orders of magnitude, but the introduced new "recalculated" PD gives 50 % better result because of the second PD controller.

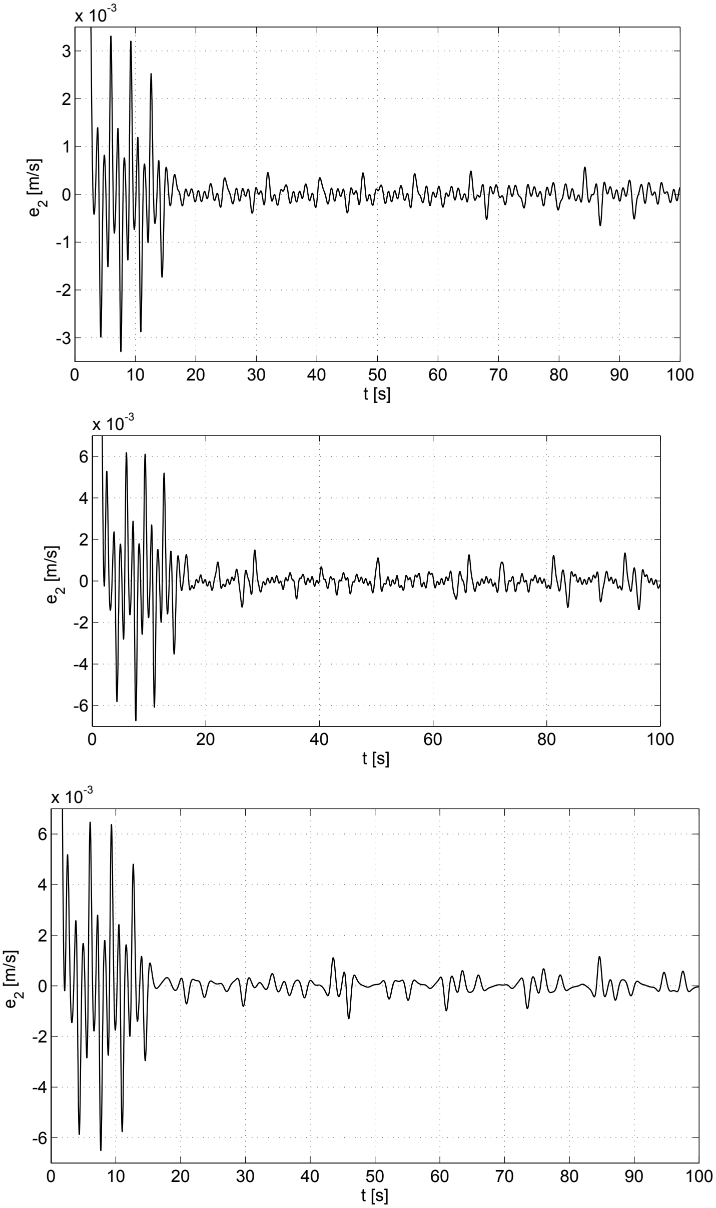

Fig. 9. The tracking errors of the second state variables of the slave systems with RFPT ($e_2 = x_2 - y_2$); top: "recalculated" PD, middle: MRAC, bottom: traditional PD. It can be seen well that they all reduce the tracking error by more than two orders of magnitude, but the proposed new "recalculated" PD gives 50 % better result because of the second PD controller

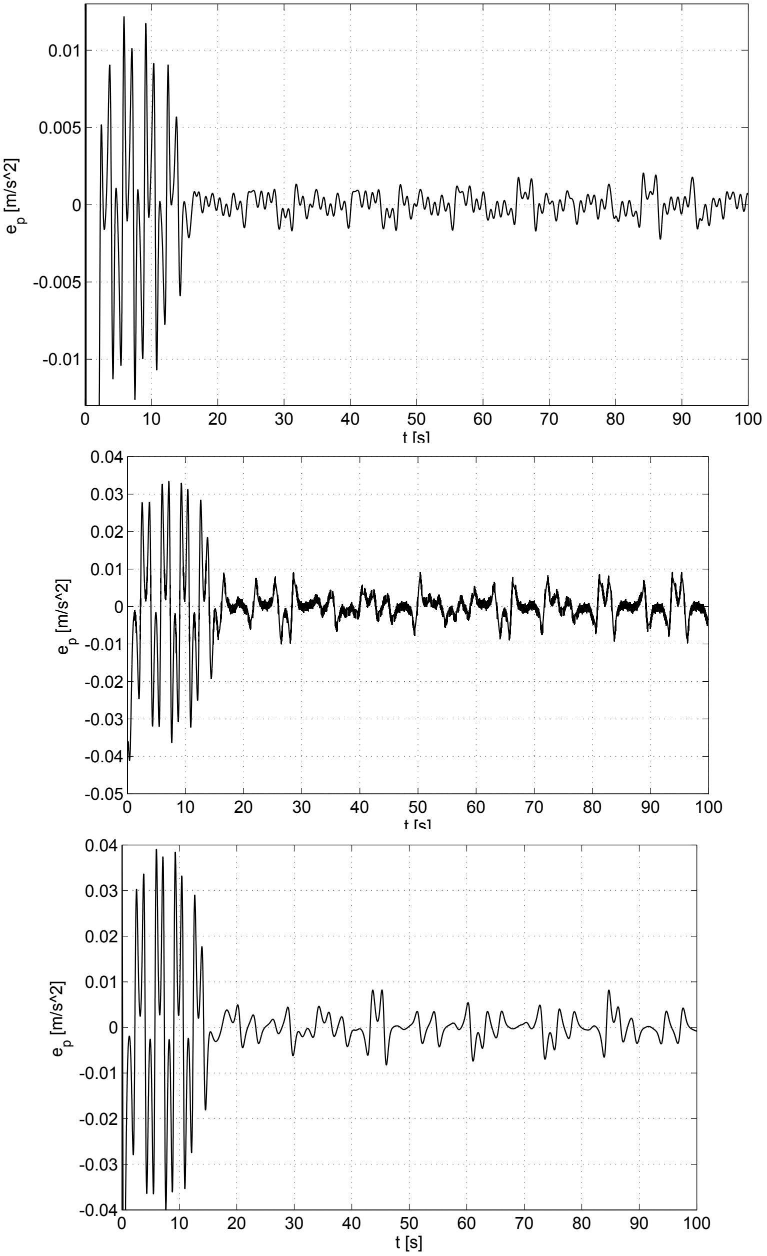

Fig. 10. The difference between the desired and the realized response ($e_p = \dot{y}_2^d - \dot{y}_2^r$) with RFPT ($e_2 = x_2 - y_2$); top: "recalculated" PD, middle: MRAC, bottom: traditional PD. As it can be seen the traditional PD and the MRAC generate similar results. The error achieved by the proposed new "recalculated" controller is one third of the others because of the integrated second controller

REFERENCES

[1] Tick, J.: Fuzzy Extension to P-Graph Based Workflow Models. In: Proceedings of the IEEE 7th International Conference on Computational Cybernetics, ICCC2009, Palma de Mallorca, Spain, 26–29 November 2009, pp. 109–112.

[2] Tick, J.: Visualisation and Simulation of P-Graph based Workflow Systems. In: Proceedings of the 6th IEEE International Symposium on Applied Computational Intelligence and Informatics, SACI 2011, Timisoara, Romania, 19–21 May 2011, pp. 231–234.

[3] Carillo, F. J.: An Improved Hybrid Adaptive PID Control System for Turning. In: Proceedings of the Third IEEE Conference on Control Applications, 24–26 August 1994, Vol. 2, pp. 1441–1442.

[4] Skoczowski, S.: A Method for Improving the Robustness of PID Control. IEEE Transactions on Industrial Electronics, Vol. 52, 2005, No. 6, pp. 1669–1676.

[5] Guo, W.: Application of Improved PID Model Algorithmic Control Algorithm. In: Proceedings of the International Conference on Intelligent Computation Technology and Automation, ICICTA, 20–22 October 2008, pp. 309–312.

[6] Piuri, V.: Dynamic Reallocation of Processes and System Dimensioning in Fault-Tolerant Control Systems. In: Proceedings of Instrumentation and Measurement Technology Conference, IMTC/93, Conference Record., IEEE, 18–20 May 1993, pp. 752–757.

[7] Piuri, V.: Design of Fault-Tolerant Distributed Control Systems. IEEE Transactions on Instrumentation and Measurement, Vol. 43, 1994, No. 2, pp. 257–264.

[8] Lee, J.: On Methods for Improving Performance of PI-Type Fuzzy Logic Controllers. IEEE Transactions on Fuzzy Systems, Vol. 1, 1993, No. 4, pp. 298–301.

[9] Xie, W.: Improved L2 Gain Performance Controller Synthesis for TakagiSugeno Fuzzy System. IEEE Transactions on Fuzzy Systems, Vol. 16, 2008, No. 5, pp. 1142–1150.

[10] Tar, J. K.: Integration of Soft Computing and Fractional Derivatives in Adaptive Control. Computing and Informatics, Vol. 24, 2005, No. 6, pp. 603–616.

[11] Várkonyi-Kóczy, A. R.—Álmos, A.—Kovácsházy, T.: Genetic Algorithms in Fuzzy Model Inversion. In: Proceedings of the 8th IEEE International Conference on Fuzzy Systems, FUZZ-IEEE '99, Seoul, Korea, 22–25 August 1999, Vol. 3, pp. 1421–1426.

[12] Várkonyi-Kóczy, A. R.—Kovácsházy, T.—Takács, O.—Benedecsik, Cs.: Anytime Algorithms in Intelligent Measurement and Control. In: CD-ROM Proceedings of the 2000 World Automation Congress, WAC 2000, Maui, USA, 11–16 June 2000, p. ISIAC-156.1-6.

[13] Matsui, N.: An Improved Model-Based Controller for Power Turbine Generators on Grid System of Shipboard. IEEE Transactions on Industry Applications, Vol. 48, 2012, No. 4, pp. 1237–1242.

[14] Tar, J. K.—Bitó, J. F.—Nádai, L.—Tenreiro Machado, J. A.: Robust Fixed Point Transformations in Adaptive Control Using Local Basin of Attraction. Acta Polytechnica Hungarica, Vol. 6, 2009, No. 1, pp. 21–37.

[15] Várkonyi, T. A.: Fuzzyfied Robust Fixed Point Transformations. In: Proceedings of the 16th International Conference on Intelligent Engineering Systems, INES 2012, Lisbon, Portugal, 13–15 June 2012, pp. 457–462.

[16] Tar, J. K.—Bitó, J. F.—Rudas, I. J.: Replacement of Lyapunov's Direct Method in Model Reference Adaptive Control with Robust Fixed Point Transformations. In: Proceedings of the 14th IEEE International Conference on Intelligent Engineering Systems, INES 2010, Las Palmas of Gran Canaria, Spain, 5–7 May 2010, pp. 231–235.

[17] Chua, L. O.—Lin, G.: Canonical Realization of Chua's Circuit Family. IEEE Transactions on Circuits and Systems, Vol. 37, 1990, No. 7, pp. 885–902.

[18] Rössler, O. E.: An Equation for Continuous Chaos. Physics Letters A, Vol. 57, 1976, No. 5, pp. 397–398.

[19] Lorenz, E. N.: Deterministic Non-Periodic Flow. Journal of the Atmospheric Sciences, Vol. 20, 1963, No. 2, pp. 130–141.

[20] Duffing, G.: Forced Oscillators with Variable Eigenfrequency and their Technical Meaning (Erzwungene Schwingungen bei Veränderlicher Eigenfrequenz und ihre Technische Bedeutung). Vieweg & Sohn, Braunschweig 1918.

[21] Tar, J. K.—Rudas, I. J.—Kozłowski, K. R.: Fixed Point Transformations-Based Approach in Adaptive Control of Smooth Systems. In: Krzysztof R. Kozłowski (Eds.): Robot Motion and Control 2007: Lecture Notes in Control and Information Sciences 360, Springer Verlag London Ltd. 2007, pp. 157–166.

[22] Várkonyi, T. A.—Tar, J. K.—Rudas, I. J.: Fuzzy Parameter Tuning in the Stabilization of an RFPT-based Adaptive Control for an Underactuated System. In: Proceedings of the 12th International Symposium on Computational Intelligence and Informatics, CINTI 2011, Budapest, Hungary, 21–22 November 2011, pp. 63–68.

[23] Várkonyi, T. A.—Tar, J. K.—Rudas, I. J.—Krómer, I.: VS-type Stabilization of MRAC Controllers Using Robust Fixed Point Transformations. In: Proceedings of the 7th International Symposium on Applied Computational Intelligence and Informatics, SACI 2012, Timisoara, Romania, 24–26 May 2012, pp. 389–394.

Teréz A. Várkonyi received her M. Sc. as a teacher of mathematics and informatics from Eötvös Lóránd University, Budapest, in 2010. She started her Ph. D. studies at Doctoral School of Applied Informatics at Óbuda University, Budapest in 2010. She also joined a dual Ph. D. program of the Doctoral School of Computer Science at Università degli Studi di Milano in 2012. Her research interests include adaptive control, nonlinear systems, and anytime systems. She was awarded the Best Student Paper Award from SOFA 2010 and the Baltazár Frankovič Young Researcher Award of the Hungarian Fuzzy Association in 2011.

József K. Tar graduated as physicist from Eötvös Lóránd University, Budapest, Hungary, in 1981. Between 1981 and 1987 he had a doctoral apprenticeship at the Central Research Institute for Physics, Budapest (far infrared plasma density diagnosis of TOKAMAK discharges). Between 1984 and 1993 he had various positions at TUNGSRAM Co. Ltd. In 1989 he received his Ph. D. degree in robotics. Between 1993 and 2011 he was Associate Professor at the legal predecessor of Óbuda University, from 2011 he is Full Professor. In 2012 he received his Dr. Sc. degree in nonlinear control. His research area is adaptive, robust nonlinear control.

Imre J. Rudas graduated from Bánki Donát Polytechnic in 1971, received the Master Degree in mathematics from Eötvös Loránd University, the Ph. D. in robotics from the Hungarian Academy of Sciences in 1987, and the Doctor of Science degree from the Hungarian Academy of Sciences in 2004. He received his first Doctor Honoris Causa degree from the Technical University of Košice, Slovakia and the second one from "Polytechnica" University of Timisoara, Romania. He is active as a Full University Professor. He served as the President of Budapest Tech from 2003 till 2010. He was elected in 2010 as the President of Óbuda University. He is a fellow of IEEE, Senior AdCom member of IES, he served IES as a Vice-President in 2000–2001. He is the chair of IEEE Hungary Section, Vice-President of the Hungarian Academy of Engineering. He is a member of various national and international scientific committees, founder of conference series, and editor of scientific journals. His research interests include computational cybernetics, robotics, soft computing, and fuzzy sets. He has published books, more than 600 papers in books, various scientific journals and international conference proceedings.