Intelligent Control in Robotic Soccer

Ján Vaščák [1], Martin Evin [2]

[1] Centre for Intelligent Technologies, Department of Cybernetics and Artificial Intelligence, Faculty of Electrical Engineering and Informatics, Technical University in Košice, Letná 9, 042 00 Košice, Slovakia, Jan.Vascak@tuke.sk

[2] Department of Production Systems and Robotics, Faculty of Mechanical Engineering, Technical University in Košice, Boženy Němcovej 32, 042 00 Košice, Slovakia, evin.martin@post.sk

Abstract: Robotic control requires solving various problems from the actuator control of players through their action planning up to strategy choice of the whole team. Such an approach supposes implementation of diverse control methods. A suitable framework for this case is offered by multi-agent systems. In this paper some control structures for robotic soccer will be described. Further, the view will be focused on use of adaptive fuzzy controllers based on the Procyk-Mamdani approach. In the paper some experiments with diverse control structures will be described and compared.

Keywords: robotic soccer, multi-agent control, adaptive fuzzy systems.

1 Introduction

Robotic soccer (RS) represents a class of benchmark problems with a high grade of completeness where a number of approaches meets together to solve all problems contented in this area, i.e. besides hardware problems mainly relating to design of control structure as well as individual control algorithms based on various principles. RS is typical with a hierarchical problem structure where diverse tasks are solved on separate control levels [5]. Firstly, a suitable strategy has to be chosen after that trajectories for robots are proposed taking into consideration mutual positions of robots and ball to fulfil goals of the chosen strategy and finally efficient control algorithms have to be used to track designed trajectories on higher levels as fast and accurate as possible. Simultaneously, lots of various limiting conditions are necessary to be considered as barriers, collisions of players, etc. which affect previously designed playing plan. To content this bunch of tasks, problems and limitations it is necessary to decompose RS and to describe relations among individual elements of the playground. Multi-agent approach seems to be a very useful means for problem description of RS as well as design of control structure.

As we meet with many factors whose influences are not clear, e.g. unpredictable behaviour of players, sliding of wheels, etc. and in general it is not easy to incorporate them into an exact mathematical model it is advantageous to use for their description and handling fuzzy logic in some parts of our control design. In addition, to support automatic control design in this paper a class of the so-called adaptive fuzzy controllers (AFC) has been used. It is a wide group of various adaptation methods, e.g. [6, 8]. Concretely, a controller using incremental models based on Jacobeans the so-called Self Organizing Fuzzy Logic Controller (SOFLC) proposed by Procyk and Mamdani [7] was used for our purposes.

To promote the goal to play soccer (football) by robotic means and to compare achieved results special tournaments in RS are organized under auspices of FIRA (Federation of International Robot-soccer) [3] round the world. The overall aim is till 2050 to develop a team of robotic players which would be able to overcome a world champion's team. As nowadays technological level does not yet enable such an intention and to access RS to so many teams as possible there exist several categories of RS with exact rules and prescriptions about parameters of the playground and robots. In this paper we will describe and deal with the one of most played categories the so-called MiroSot one.

2 Multi-agent Approach in Control of RS

From the point of view of artificial intelligence (AI) a match between two teams can be considered as a multi-agent system. Each player (robot) is an autonomous system, i.e. an agent having only a limited range of information about the state of other players. Their communication is restricted as well as their ability to sense the situation. Especially, in the category SimuroSot there are some extra limitations like noised signals, limited energy of players, etc. to simulate real playing situation.

A rough sketch of the control is depicted in Fig. 1. We can observe three basic control levels of this hierarchy. Firstly, the highest (third) level the so-called *strategy choice* has to determine roles to players acting as relatively independent agents but aiming a common goal. On the second level trajectories of robots are constructed. There are at least two modules as *target search* and *path observation* which represent requirements and possibilities of movement, respectively and which can be partially contradictory. The final trajectory depends on finding convenient 'intersection' of these two conditions. A robot has to approach to the ball as the first target and then to shift it to the second target, i.e. the rival's gate (either directly or with help of other players). However, observing such a desired trajectory must not collide with limitations like own players or barriers of the playground. In such a way it is corrected to respond real situation. Both tasks are performed simultaneously and are mutually conditioned. Therefore they are on the conjoint level. The lowest (first) level is responsible for transforming such a movement trajectory into commands controlling rotations of wheels. Besides it also solves various accidental events like unpredicted movements of rivals or defected behaviour of own robots.

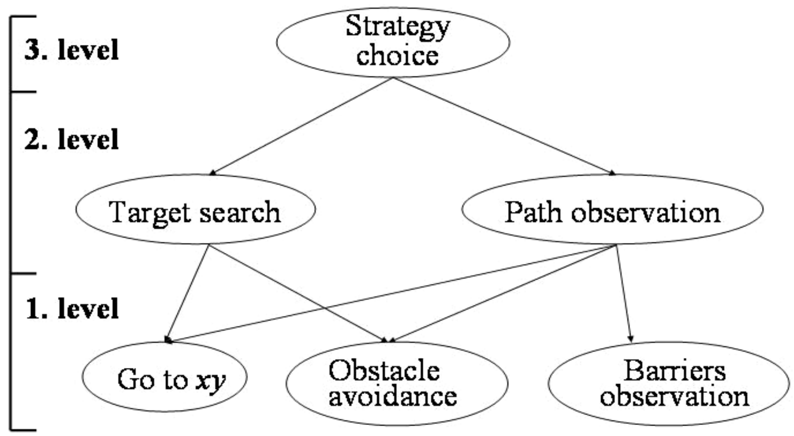

Figure 1

Multi-agent control structure of RS

Of course each module from Fig. 1 can be decomposed into sub-modules as it is shown in the case of *strategy choice.* Principally, we have to decide between defence and attack (Fig. 2).

There exists still one level (not depicted in Fig. 1) the process one which controls the rotations of wheels to ensure tracking control of the chosen trajectory. Its structure depends on physical properties of the given robot which determines inputs and outputs of such a controller.

Concerning means used for implementing control structures from Fig. 1 as well as process controller it can be stated that means of AI are used mostly on higher levels and conventional means on lower ones but is not a law. However, on higher levels more sophisticated decision processes are desired that are much more similar to human reasoning and although on lower levels the decision processes are not so much complicated but questions about control speed, its precision and robustness are very important which is very suitable for using conventional methods based on physical equations. In this paper we tried to compare the efficiency of conventional and AI in form of AFC on the process level, too. The reason is in problems how to design parameters of such a controller. Manual setting up is time consuming and requires considerable skill. In the next chapter the principle of AFC based on design by Procyk and Mamdani will be explained.

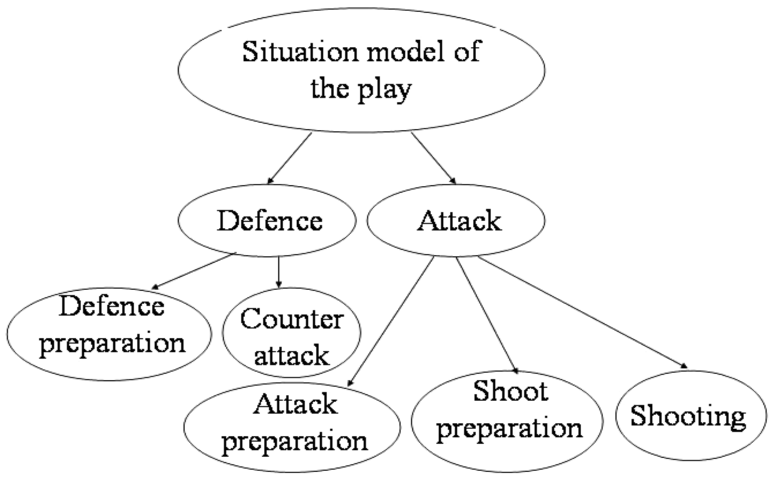

Figure 2
Control structure of strategy choice

3 Structure of Self Organizing Fuzzy Logic Controller by Procyk and Mamdani

In principle, the values of knowledge base parameters can be obtained in two ways: either by identifying the parameters of the controlled system or by measuring the control quality. The first way defines the so-called *parameter-adaptive systems* and the second one defines *performance-adaptive systems*. In the first case, the information about the controlled system obtained in such a way is then to be transformed into the form of fuzzy rules of the controller. Therefore the methods of this kind are known as *indirect methods*, too (see [1, 4]). The performance adaptive systems transform the measured control quality directly to the controller parameters excluding the need of the system identification. They enable also to include other criteria where the minimal control error (control task) seems to be only a special criterion.

SOFLC belongs to the category of *performance-adaptive systems* and its structure is shown in Fig. 3. Control criteria are contented in the block of *performance measure* where the quality is evaluated by the *performance index p(k)* which expresses the magnitude and direction of changes to be performed in the knowledge base of the controller. The basic design problem of AFC consists in the design of *M*, where for each time sample *t=K.T* (*K=0,1, ...*) a simplified *incremental model* of the controlled system $M=J.T$ (J - Jacobean) is computed. It represents a supplement to the original model to reach a zero control error and is analogous to the linear approximation of the first order differential equation or in other words to gradients, too. As Jacobean (1) is a determinant of all first derivatives of the system with n equations $f_1, \ldots, f_n$ of n input variables $x_1, \ldots, x_n$ it means J is equal to the determinant of the dynamics matrix, i.e. it is a numerical value describing all n gradients in the sense of a characteristic value.

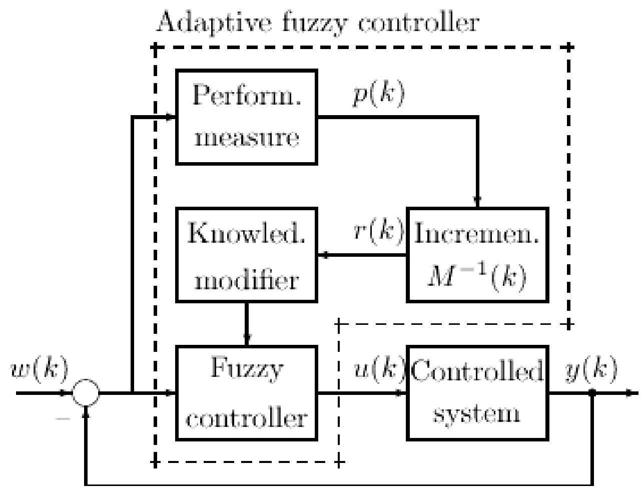

Figure 3
Self-organizing fuzzy logic controller

Now we need to transform this incremental description of a controlled system to the description of a controlling system, i.e. a controller. Considering the properties of the feed back connection we can see that $y(k) \approx e(k)$ ($w(k)$ is known). As inputs and outputs of a controlled system change to outputs and inputs of a controller, respectively we can get the controller description like an inverse function of $y(k) = f_M(u(k))$, i.e. the model of the controller is $u(k) = f^{\,1}{}_M(y(k))$. Because J is a number, then M^{1} is the reverse value of $J.T$. The reinforcement value $r(k)$ is computed as $r(k) = M^{1}.p(k)$.

$$J = \begin{vmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ & & \vdots & \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{vmatrix} = \begin{vmatrix} grad\ f_1 \\ \cdots \\ grad\ f_n \end{vmatrix} \qquad (1)$$

The knowledge base adaptation can be either relation-based or rule-based. In this case the second way of adaptation was used. In general (for both methods), it is based on removing such rules $R_{new}(k)$ that caused a 'bad' control in the previous time step $R_{bad}(k)$ and including new 'reinforced' rules, i.e. for next time step $k+1$ we will get:

$$R(k+1) = (R(k) \cap \overline{R_{bad}}(k)) \cup R_{new}(k) \tag{2}$$

Each fuzzy rule r_p ($p = 1, ..., N_r$) of n inputs and one output represents their Cartesian product and is also a fuzzy relation $R_p = A_{1,p}\ x\ ...\ x\ A_{n,p}\ x\ B_p$. The knowledge base R is then a union of such rules (fuzzy relations) and after substituting into (2) it will be changed to (3). $R_{bad}(k)$ can be a union of all previously fired rules, too. However, for the sake of simplicity we will consider only one rule with the greatest strength α and therefore $A^{bad}_1\ x\ ...\ x\ A^{bad}_n$ is its premise. Reinforcement value $r(k)$ corrects only the consequent of such a rule and B_{new} is the fuzzified result of $y(k)+r(k)$, i.e. $fuzz(y(k)+r(k))$. The simplest fuzzification is in the form of singletons but in general, other forms are possible, too.

$$R(k+1) = \begin{array}{l} \left[\bigcup_{p=1}^{N_r} (A_{1,p} \cap \overline{A_1^{bad}})\ x \ldots x\ A_{n,p}\ x\ B_p \right] \cup \\ \ldots \cup \left[\bigcup_{p=1}^{N_r} A_{1,p}\ x \ldots x\ (A_{n,p} \cap \overline{A_n^{bad}})\ x\ B_p \right] \cup \\ \cup \left[\bigcup_{p=1}^{N_r} A_{1,p}\ x \ldots x\ A_{n,p}\ x\ (B_p \cap \overline{B^{bad}}) \right] \cup \\ \cup \underbrace{(A_1^{bad}\ x \ldots x\ A_n^{bad}\ x\ B^{new})}_{R_{new}} \end{array} \tag{3}$$

As seen in (3) there is a problem of rule expansion. If n is number of inputs and $nr(k)$ is number of rules in the step k then in the next step $k+1$ the number of rules will be $nr(k)\ .\ (n + 1) + 1$. Besides shapes of membership functions, in other words their meaning, are changed, too. To eliminate this problem an auxiliary *garbage collection* mechanism is necessary. For this purpose similarity relations (in our case Hamming measure) were used which in general compare two objects (in this case membership functions) and their similarity is represented by a number from the range [0; 1]. We can use following rules using similarities s to reduce the knowledge base.

1 If s has a *small* value then add new membership functions as well as rules generated in (3).

2 If s has a *medium* value then replace compared membership functions with their average and omit identical rules.

3 If s has a *high* value then do not add any new membership functions and add new rules with original functions if they do not yet exist.

Such a garbage collection can be done either after each adaptation step or after the entire adaptation has been done. On base of our experience [11] the adaptation after each step was chosen.

4 Control Structures for RS

For control of robots two variables are calculated: speed v and direction ω which are converted to rotations of the left n_L and right n_R wheel, respectively. Original control structure PID-1 which was later modified in two other ones [2] is shown in Fig. 4. There was not used multi-agent approach and a two-layer control was applied. In the higher level the movement of a player to the ball was solved. Various mutual situations were described by IF-THEN rules. The lower level was realised by two conventional feedback PID controllers where desired rotations n_L and n_R are compared to real ones $n_L^{'}$ and $n_R^{'}$ and their differences $n_L^{''}$ and $n_R^{''}$ represent corrections of actuators.

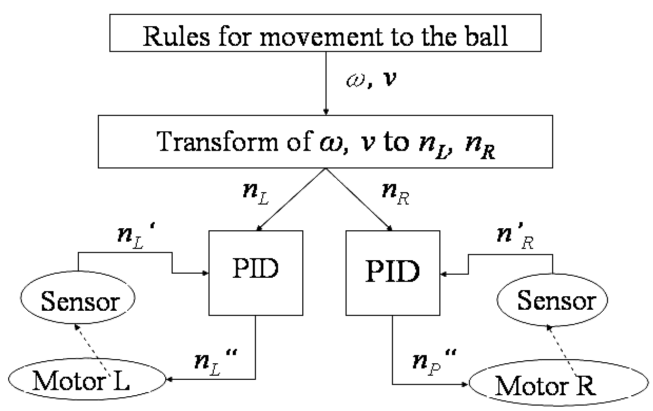

Figure 4

Control structure PID-1 (without multi-agent approach)

As the strategic level is absent in the previous structure it can be used only for simpler tasks but not for a complex play. Further, problems with sliding of wheels on the playground plate arise and in case of collisions with rivals their power acting was unsatisfactory. Therefore a control structure of the second generation PID-2 (see Fig. 5) was created to minimize these problems. This approach is already multi-agent (Fig. 1). The process control level is realised by PID controllers. The first level is performed by a rule-based expert system for *power action* to achieve satisfactory power acting on rivals in the case of collision. Similarly, by other two expert systems further two control layers are realised.

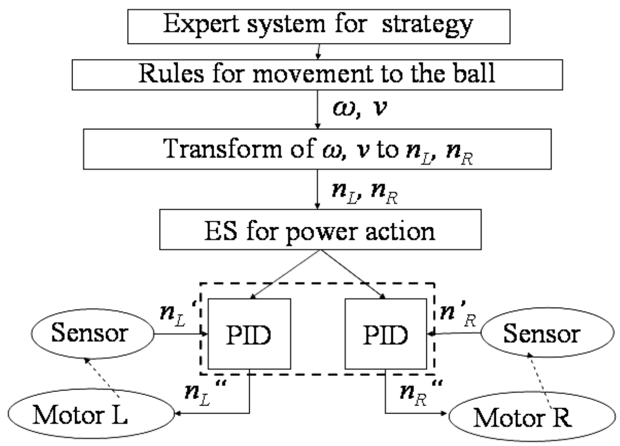

Figure 5

Control structure PID-2

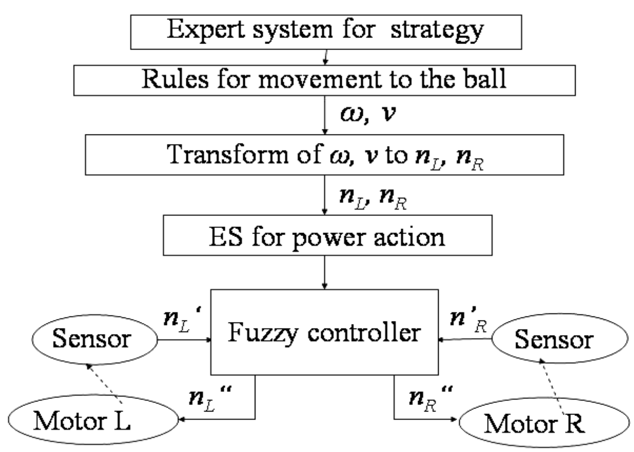

Figure 6

Control structure FUZZY

To avoid problems with sliding another modification was created using a fuzzy controller (named as FUZZY) replacing two PID controllers (see Fig. 6). Firstly, the problem of clocking differences of two controllers was eliminated and secondly it was possible better to cover diverse inaccuracies. To avoid necessity of manual setting up manifold parameters a SOFLC was used as described in the previous chapter.

5 Experiments

To compare properties of proposed control structures four kinds of experiments were done (see Fig. 7) with measuring following errors:

1 path deviation – forward movement
2 path deviation – bidirectional movement
3 rotation angle deviation round the own axis
4 circle deviation

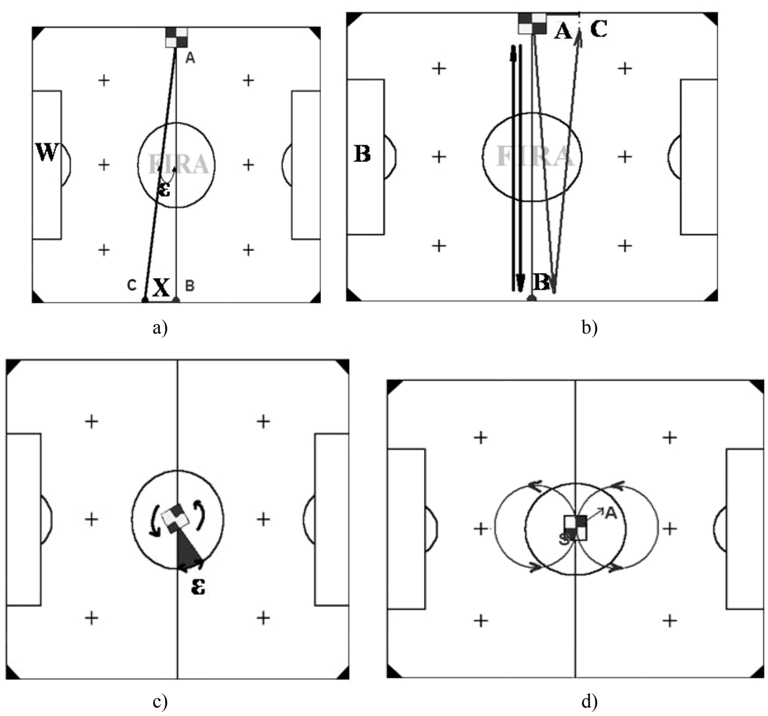

Figure 7
Experiments a) forward movement; b) bidirectional movement; c) rotation round the own axis; d) circling

In the case of straightforward movements (experiments N. 1 and 2) the percentage error was referred to the playground breadth *B*. So in the first case it was computed as *X / B* and in the second case as *AC / B* where *X* and *AC* are deviations from desired goal point (Fig. 7). The other two experiments relate to rotation movements. The percentage error in the third experiment was calculated as division of the deviation angle *ε* and total angle range (*ε / 3,6*). In the last experiment it was calculated as division of the deviation distance *SA* (*S* – desired final point, *A* – real final point) and circle radius *r*, i.e. *(SA / r). 100*. Of course, the deviation angle *ε* can be measured also in the first experiment. Obtained results are shown in Tab. 1.

Experiment	**1**		**2**	**3**		**4**
Error	**ε [ε]**	**%**	**%**	**ε [ε]**	**%**	**%**
PID-1	16,74	30,08	51,15	19,1	5,3	-
PID-2	5,49	9,61	5,69	9,8	2,72	14,75
FUZZY	4,31	7,54	3,85	11,7	3,25	24,75

Table 1

Comparison of control structures by errors

In tab. 1 we can see the structure PID-1 was fully unsatisfactory. Even it was not able to fulfill the fourth experiment and therefore its results are not indicated. The quality of structures PID-2 and FUZZY is comparable where in the case of straightforward movement FUZZY is better and in the case of rotational movement the order is changed to PID-2.

Conclusions

The control structure PID-1 is still typical for many teams. Although its quality seems to be very bad in comparison to other two structures but during a play it is not so much serious because the play is broken off very often and robots operate on short trajectories due to quick changes of playing situation. However, new strategies do not bring only much better quality but in the case of PID-2 also simpler hardware structure. Adaptive fuzzy approach contributed in two points. Firstly, it was able better to operate on greasy surfaces and with noised visual signal. Its robustness was unambiguously better. Secondly, it was again shown (another example e.g. in [9, 10]) that adaptation mechanisms are suitable also for highly dynamic systems with short time responses like for instance RS and time of design was considerably shortened.

References

[1] Baker W. L., Farrell J. A.: An introduction to connectionist learning control system, In: D. A. White, D. A. Sofge (Eds.), Handbook of Intelligent Control-Neural, Fuzzy and Adaptive Approaches, van Nostrand Reinhold Inc., New York, 1992

[2] Evin, M.: Intelligent Control Design in Robotic Soccer, Master Thesis, Technical University in Košice, Slovakia, 2005, p. 55

[3] Federation of International Robot-soccer: http://www.fira.net/

[4] Kim Y. T., Bien Z.: Robust self-learning fuzzy controller design for a class of nonlinear MIMO systems, Int. Journal Fuzzy Sets and Systems, Elsevier Publisher, Holland, N. 2, Vol. 111, 2000, pp. 17-135

[5] Madarász L., Ottmár P., Timko, M.: Situational Control of Complex Systems and Artificial Intelligence, In: Proc. of Jubilee Scientific Conference, Bánki Donát, Budapest, Hungary, September 1999, pp. 177-185

[6] Mls, K.: Hybrid Fuzzy Cognitive System (in Czech), III. Slovensko-Český seminár Kognícia, umelý život a počítačová inteligencia, Stará Lesná 14.-16.5.2003, elfa, s.r.o., Košice, s. 233-238

[7] Procyk T. J., Mamdani E. H.: A linguistic self-organizing process controller, Automatica No. 15, 1979, pp. 15-30

[8] Tar J. K., Rudas I. J., Bitó J. F.: Comparison of the Operation of the Centralized and the Decentralized Variants of a Soft Computing Based Adaptive Control, In: Proc. of Jubilee Conference Budapest Tech, September 4, ISBN 963 7154 31 0, pp. 331-342

[9] Tar J. K., Rudas I. J., Bitó J. F., Horváth L., Kozlowski K.: Analysis of the Effect of Backslash and Joint Acceleration Measurement Noise in the Adaptive Control of Electro-mechanical Systems, In: Proc. of the 2003 IEEE International Symposium on Industrial Electronics (ISIE 2003), June 9-12, 2003, Rio de Janeiro, Brasil, CD issue, file BF-000965.pdf, ISBN 0-7803-7912-8, IEEE Catalogue N. 03th8692

[10] Vaščák, J.: Hybrid Adaptive Control of LEGO Robots, In: SAMI 2003 - 1st Slovakian – Hungarian Joint Symposium on Applied Machine Intelligence, Herľany, Slovakia, 2003, pp. 225-235, ISBN 963 7154 140

[11] Vaščák, J.: Use of Adaptive Fuzzy Systems in Prediction and Control (in Slovak), In: 8. conference Inteligentní systémy pro praxi, Seč u Chrudimi, Czech Republic, 2004, pp. 35-36, ISBN 80-239-3680-8