Structure for Behaviourist Representation of Knowledge

Claudiu Pozna

Department of Product Design and Robotics
University Transilvania of Brasov
Bd. Eroilor 28, RO-500036 Brasov, Romania
E-mail: cp@unitbv.ro

Radu-Emil Precup, Stefan Preitl

Department of Automation and Applied Informatics
"Politehnica" University of Timisoara
Bd. V. Parvan 2, RO-300223 Timisoara, Romania
E-mail: radu.precup@aut.upt.ro, stefan.preitl@aut.upt.ro

Emil M. Petriu

School of Information Technology and Engineering
University of Ottawa
800 King Edward, Ottawa, ON, K1N 6N5 Canada
E-mail: petriu@site.uottawa.ca

József K. Tar

Institute of Intelligent Engineering Systems
Budapest Tech Polytechnical Institution
Bécsi út 96/B, H-1034 Budapest, Hungary
E-mail: tar.jozsef@nik.bmf.hu

Abstract: The paper suggests a three level hierarchical structure dedicated to the behaviourist representation of knowledge. The first level, referred to as the strategic one, is used in the problem solving providing the solution. The second level, called the tactical one, is employed in the solution implementation. The third level, named as the operational one, is meant for the solution execution. The first level is a strategically level the second is a tactical level and the third is an operational level. Feedback information from all higher *hierarchical levels is used in learning to develop the lower levels. A case study concerning the control system of an autonomous car is considered to exemplify the design of the new knowledge representation structure.*

Keywords: behaviour, knowledge, robots

1 Introduction

Knowledge Representation (KR) has been an active field in the more than six decades of history of artificial intelligence. The research concerning the KR can be organized in three main approaches, the weak method problem solving, the strong method problem solving, and the agent-based approaches.

The recent research directions in KR belong to the three main approaches. Answering and question-specific KR with web application is treated in [1]. The ontology domain-based KR is proposed in [2]. An integration of entities, relations and problems in KR is suggested in [3]. Attractive applications of KR in prediction and identification are presented in [4, 5].

The aim of this paper is to suggest an original three level hierarchical structure for KR. It is viewed in the framework of conceptual graph KR [6] and subsumption architecture for robots [7]. According to [7] the intelligence is the product of the interaction between an appropriately designed system and its environment. Our idea starts from this point, viz. the need to design a system which allows interactions, in a specific way, with the environment and learn from these interactions. By specific way we understand interactions based on known collections of behaviours. By learning from the mentioned interaction we understand the possibility to develop the collections of behaviours. Therefore the new structure ensures the behaviourist representation of knowledge.

The paper treats the following topics. The conceptual description of the KR structure is presented in the next section. Section 3 is dedicated to the case study where subsystems of the new KR structure are designed and implemented in a control system for an autonomous car. The conclusions are highlighted in Section 4.

2 Conceptual Description of Knowledge Representation Structure

The KR hierarchical structure is presented in Fig. 1. The structure consists of three levels described as follows.

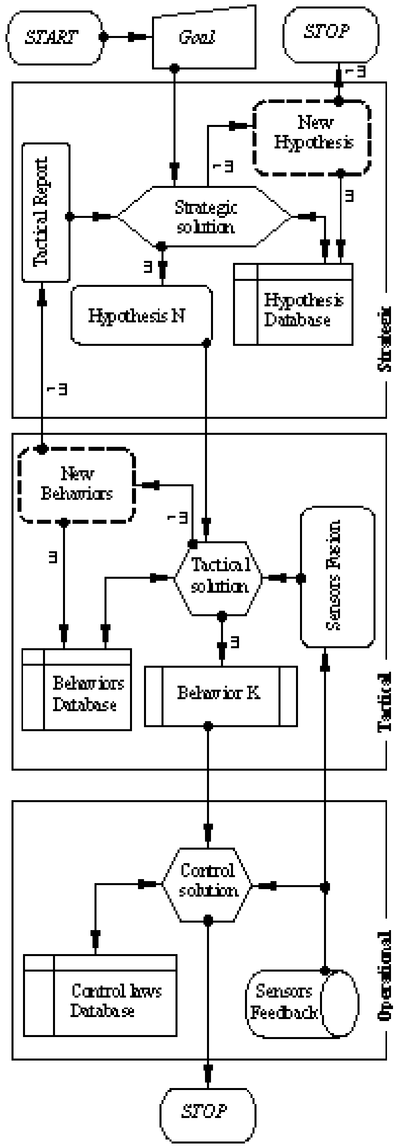

Figure 1
Structure of Knowledge Representation

The first level is the strategic level. It transforms the user goal in to a strategy. It operates as a problem solver engine with two main subsystems, a hypothesis database and a procedure which chooses the best hypothesis and next transforms it into a strategic solution.

The strategic level contains procedures needed for the constructions of the new hypothesis. That construction enables the learning process.

The second level is the tactical level. It is dedicated to the transformation of the strategic solution into a tactical one. Therefore a behaviour data base is queried for the solution.

A behaviour is a network of actions used to implement a part of the strategy. The tactical solutions becomes in a step-by-step manner a network obtained from dynamically linked behaviours.

The tactical level contains procedures needed for the constructions of the new behaviour. That construction supports the learning process.

The third level is the operational level. Its role is the management of the tactical solution. More precisely it triggers and controls each action from the mentioned network, and gathers information from the sensors.

If the sensors confirm the control solution the operational level will continue the tactical solution. If not new tactical solutions are needed.

The structure presented in Fig. 1 shows that the goal (task, problem) is transformed into a hypothesis by the following subsystems:

- the Strategic solution block, which operates also with the Hypothesis Database by a collection of strategic solutions,
- the Tactical Report block by information about the failure of the tactical solutions,
- the New Hypothesis block by a procedure which constructs the new hypothesis.

The selected hypothesis generates and transforms a behaviours network into a Tactical solution block. That is the reason why the mentioned block is also linked with the following subsystems:

- the Behaviors Database, representing a collection of behaviours,
- the Sensor Fusion block, which stands for a set of information on the environment,
- the New Behaviors block, which is a procedure employed in the generation of new behaviours.

The current behaviour contains a set of actions which are triggered by the Control solution block.

The proposed structure contains all the three levels of information processing i.e.

- the abduction, where the best hypothesis is selected from a set of hypotheses,
- the deduction, where the Goal is transformed into a hypothesis or the hypothesis is transformed into a behaviour,
- the induction, which is used for the constructions of new behaviours or hypotheses.

3 Case Study

The case study is dedicated to the design of part of the subsystems presented in the previous section for the control system of an autonomous car. The implementations were done at the University of Applied Science Heilbronn, Germany, in the framework of the Automotive Competence Centre (ACC) “Fahrautomat” project. Therefore the autonomous car is referred to also as the ACC autonomous car or the ACC autonomous robot [8, 9].

The KR structure design is supported by the analysis of the architectures which model the human behaviour. Useful approaches are reported in [10, 11]. We consider that it is suitable to model and implement rather the “human driver decisions act” than the “human driver actions”. Thus the KR structure is organized in terms of Fig. 1.

To obtain the human driver behaviour a preliminary analysis should offer answers to the following questions:

- “how does a common driver act, or what is a driving behaviour?”,
- “can we obtain a fundamental truth about that behaviour and use it in our construction?”,
- “can we identify tools to transform and implement that behaviour into a software architecture?”.

To answer the first question we must give the definition of the behaviour by underlining the semantic characteristics of “driving behaviour”. It is important to derive the category tree of that word making use of {act → activity → (behaviour, practice, ...)}. Therefore the behaviour is an action or a set of actions performed by a person under specified circumstances that reveal some skill, knowledge or attitude.

In order to describe the driving behaviour emphasis is given to the word “custom” which is from the same category tree {act → activity → practice → custom, …}. It is defined as accepted or habitual practice. In many situations the customs have a special nature as the automatism standing for any reaction that occurs automatically without conscious thought or reflection.

Making use of the previous comments our understanding of "driving behaviour" is of an action or a set of actions performed by a person under driving circumstances, which tend / tends to be transformed into customs and even in automatisms. In fact the "driving behaviour" consists of a collection of behaviours including the driver's behaviour when he / she activates the ignition, the driver's behaviour when he / she stops the car, etc. Therefore the following fundamental truth of "driving behaviour" is defined:

- The driver establishes a priori the current driving goal.
- A behaviour is a set of actions.
- The behaviours are linked together creating a system which allows the solving and gives solutions considering the driving circumstances.
- The translation from any behaviour to another one is triggered by an event.
- The system is developed by learning and experience.
- The behaviours presume decisions with incomplete information or even in fuzzy environments.
- The set of actions is transformed in time into customs and automatisms.

Using the above proposition we can focus on the tactical level and model (approximate) the "driving behaviour" by a collection of highly linked programs (behaviours) which are stored in a memory. The decision to run a certain program is made by a Tactical Solution program. That decision is based on the goal of driving and acknowledging about the environment (the driving circumstances). Each program (behaviour) is a succession of instructions (actions) which impose the parameters and trigger the actuators. The software architecture is presented in Fig. 2. Some comments concerning the software architecture are presented as follows.

The strategic level, where the robot must transform his goal into a hypothesis, is replaced with an interface where the human operator imposes the hypothesis.

The Tactical Solution analyzes the hypothesis in the driving circumstances which are obtained from the sensors. The results of that process are the state vector of the robot (the desired position, velocity, etc.), and the decision to run certain program from the Behaviors subsystem.

The Output Interface supports the human operator in reading the state vector and the errors of the robot. It enables also the memorization of the past robot state trajectories.

The Actuators Communications outputs the data to the microcontrollers attached to all actuators. The Sensors inputs the data from the sensors.

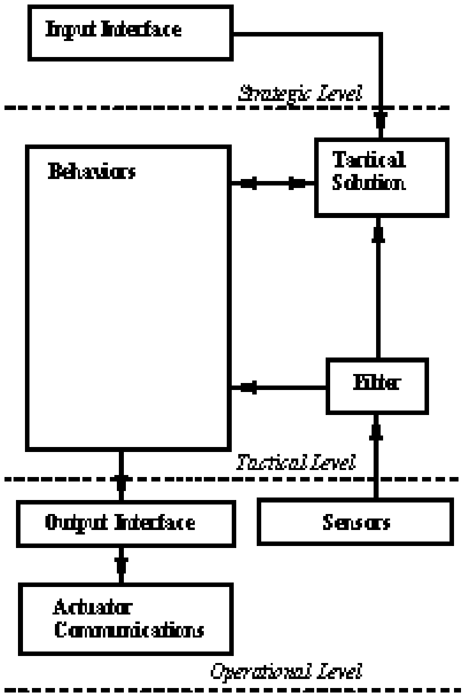

Figure 2
Software architecture

The programs (bricks) are developed making use of three different structures named basic behaviour, error behaviour and simple behaviour. The structures of the Behaviors subsystem are defined in Fig. 3. The main differences between the bricks are the connection type (P – previous, N – next, E – error, QI – quick in, QO – quick out) and the direction of information flow.

The decisions on which brick to be connected are made by the Tactical Solution. That program compares the goal of the robot with the driving circumstance, establishes the status vector, and enables the brick which must run. After these decisions the program continues to compare the robot goal with driving circumstance. If the result is acceptable, nothing is changed (the same brick is run), in contrary, a “Crisis” or a “Failure” event is brow caste. “Crisis” means that a new behaviour is needed, so the status vector as well as the brick is changed. “Failure” means that we do not have solutions (behaviours) to solve the problem and we must stop safely the robot. All those processes are presented systematically in Fig. 4.

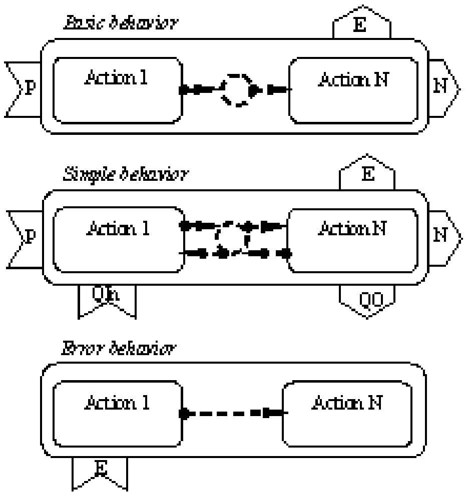

Figure 3
Structures of programs in Behaviors subsystem

Concluding, the software has a three level structure. The goal of the robot is imposed by the human operator via the Input interface.

The tactical level (Fig. 4) finds the solution linking several programs (bricks). A brick is a succession of actions. An action sets the parameters and triggers the actuators.

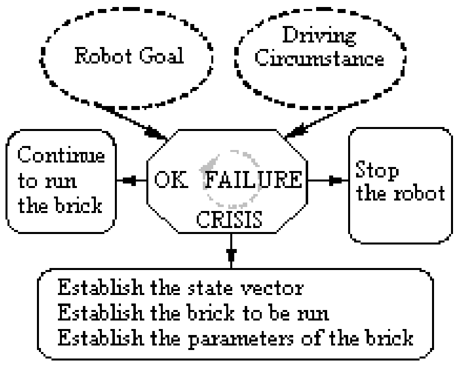

Figure 4
Architecture of Tactical Solution program

Two control loops are implemented at the higher and lower hierarchical level. The higher level control loop is implemented by the Tactical Solution aiming the fulfilment of the robot goal by means of appropriately defined performance specifications. The operational level control loop is placed at the lower hierarchical level. Its tasks are solved by each microcontroller connected to the afferent actuator and sensor.

The control program is implemented in Matlab. It uses the xPC toolbox. The communications tools between the actuators, sensors and the control program tools are built via the CANopen network. Some details on the experimental setup are presented in Figs. 5-7.

Figure 5
General structure of experimental setup

The following state-space model of the car [12] is used to design the lower level control loop:

$$\begin{bmatrix} \dot{\beta}(t) \\ \ddot{\psi}(t) \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \cdot \begin{bmatrix} \beta(t) \\ \dot{\psi}(t) \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \cdot \delta \cdot \tag{1}$$

The parameters in (1) are

$$\begin{aligned} &a_{11} = -2c/(mV),\ a_{12} = (cl_s - cl_f)/(mV^2), \\ &a_{21} = c(l_s - l_f)/J_z,\ a_{22} = -c(l_s^2 + l_f^2)/(J_z V), \\ &b_1 = c/(mV),\ b_2 = cl_f/J_z, \end{aligned} \tag{2}$$

V is the car velocity that is considered to be constant, β is the angle between the direction of the velocity V and the car direction, δ is the steering angle accepted as control signal, ψ is the car direction angle, m and J_z are the mass and the momentum of the car, respectively, c is the rotational stiffness of the wheels, l_f and l_s are the lengths from the mass centre to the front and back wheels, respectively.

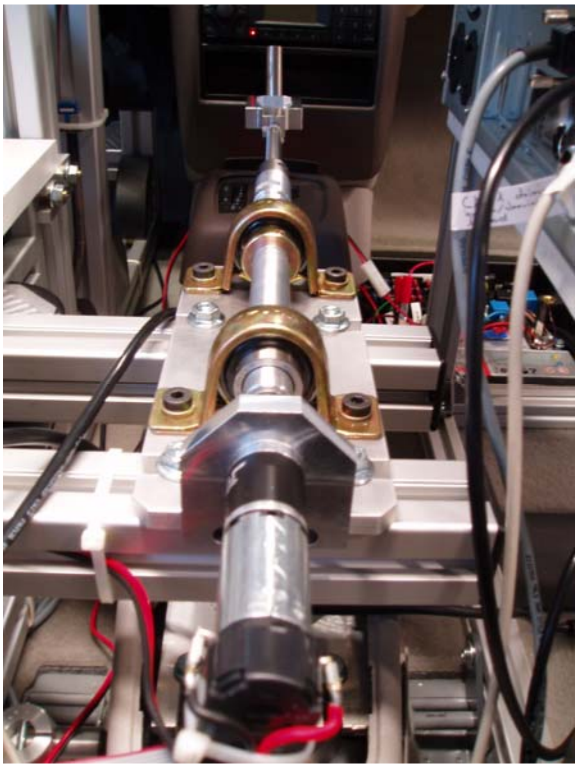

Figure 6
Gearbox stick mechanism

The trajectory model is obtained on the basis of the hypotheses [13]

$$\begin{aligned} y &= V(\theta_\Delta + \beta), \\ \theta_\Delta &= \theta_P - \psi, \\ \dot{\theta}_\Delta &= V\kappa_P - \dot{\psi}, \end{aligned} \tag{3}$$

where y is the controlled output standing for the distance between the desired trajectory and the car mass centre, θ_P is the angle of the trajectory tangent, ψ is the angle of the car referential frame, β is the car speed angle (in the car referential frame), and κ_p is the desired trajectory curvature.

The combination of the models (1) and (3) leads to the complete mathematical model of the car. Its input-output representation assuming zero initial conditions is [9]

$$Y(s) = G_{y,\delta}(s)\Delta(s) + G_{y,\kappa_P}(s)\mathrm{K}_P(s), \tag{4}$$

where $G_{y,\delta}(s)$ and $G_{y,\kappa_P}(s)$ are transfer functions with respect to the control signal and disturbance input, respectively:

$$G_{y,\delta}(s) = V\frac{b_1 s^2 + [(a_{12}-1)b_2 - a_{22}b_1]s - (a_{21}b_1 - a_{11}b_2)}{s^4 - (a_{11}+a_{22})s^3 + (a_{11}a_{22} - a_{12}a_{21})s^2},$$
$$G_{y,\kappa_P}(s) = \frac{V^2}{s^2}, \quad (5)$$

$Y(s)$ is the Laplace transform of $y(t)$, $\Delta(s)$ is the Laplace transform of $\delta(t)$, and $\mathrm{K}_P(s)$ is the Laplace transform of the disturbance input $\kappa_P(t)$.

Figure 7
Steering wheel mechanism

The frequency domain design can be applied for the unstable plant (5). It leads to the PID controller with the transfer function

$$G_C(s) = k_C + k_d / s + k_i s, \quad (6)$$

where k_C, k_d, k_i are the proportional, derivative and integral gains, respectively.

Next the transfer function in (6) in can be decomposed as the parallel connection of a PI controller with the transfer function $G_{PI}(s)$ and a PD controller with the transfer function $G_{PD}(s)$:

$$\begin{aligned} &G_C(s) = G_{PI}(s) + G_{PD}(s), \\ &G_{PI}(s) = k_{C1} + k_d / s, \\ &G_{PD}(s) = k_{C2} + k_i s, \\ &k_C = k_{C1} + k_{C2}, \end{aligned} \quad (7)$$

and (7) allows the design of low-cost Mamdani or Takagi-Sugeno PID-fuzzy controllers in terms of the modal equivalence principle [14-16]. The controller structures and designs can compensate for the additional modelled or un-modelled dynamics and nonlinearities associated to the controlled plant.

Conclusions

This paper proposes a three level KR structure based in the idea that the intelligence is supported by appropriate designed systems and interactions with the environment. The new structure operates at strategic, tactical and operational levels. The functions of the three levels are given in relation with the human mental process thus the modelling of the abduction, deduction, and induction is achieved by the KR structure.

The control system of the ACC autonomous car is considered as the application of the new structure. It should be underlined that only a part of the KR structure was designed. Therefore the first direction research direction aims the design of the overall KR structure for the ACC autonomous car.

The paper gives also directions for design of one of the control loops of the autonomous car. The future research will be dedicated to the design and implementation of the procedures that allow the design of new behaviours and hypotheses.

Another future research direction concerns the design of control structures for the autonomous car and similar autonomous robots. Use will be made of low-cost models and controllers that should ensure very good control system performance indices [17-27].

Acknowledgement

The paper is supported by the Romanian Ministry of Education, Research and Innovation through the PNII Idei project 842/2008, the CNCSIS and CNMP of Romania. The co-operations between The Budapest Tech Polytechnical Institution and the “Politehnica” University of Timisoara (PUT), and the University of Ljubljana and the PUT, in the framework of the Hungarian-Romanian and Slovenian-Romanian Inter-governmental Scientific & Technological Cooperation Programs are acknowledged. The support offered by the University of Applied Science Heilbronn, in the framework of the ACC project, is also acknowledged.

References

[1] E. Sneiders: Automated FAQ answering with question-specific knowledge representation for web self-service, in Proceedings of 2nd Conference on Human System Interaction (HSI ’09), Catania, Italy, 2009, pp. 298-305

[2] Y. Sun, Z. Li: Ontology-based domain knowledge representation, in Proceedings of 4th International Conference on Computer Science & Education (ICCSE 2009), Nanning, China, 2009, pp. 174-177

[3] Y. Peng, T. Wang, Y. Z. Wang: Research about intrusion detection based on entities-relations-problems knowledge representation system in e-

government environment, in Proceedings of First International Workshop on Education Technology and Computer Science (ETCS ’09), Wuhan, China, 2009, pp. 665-668

[4] G. Lee, S. Doyle, J. Monaco, A. Madabhushi, M. D. Feldman, S. R. Master, J. E. Tomaszewski: A knowledge representation framework for integration, classification of multi-scale imaging and non-imaging data: preliminary results in predicting prostate cancer recurrence by fusing mass spectrometry and histology, in Proceedings of 2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro (ISBI ’09), Boston, MA, 2009, pp. 77-80

[5] N. Qadeer, R. Velik, G. Zucker, H. Boley: Knowledge representation for a neuro-symbolic network in home care risk identification, in Proceedings of 7th IEEE International Conference on Industrial Informatics (INDIN 2009), Cardiff, UK, 2009, pp. 277-282

[6] J. F. Sova: Knowledge Representation: Logical, Philosophical, and Computational Foundations, Brooks Cole, Pacific Grove, CA, 2000.

[7] R. Brooks: A robust layered control system for a mobile robot, IEEE Journal of Robotics and Automation, 1986, Vol. 2, No. 1, pp. 14-23

[8] C. Pozna, F. Troester: Human behavior model based control, Acta Polytehnica Hungarica, 2006, Vol. 3, No. 3, pp. 59-70

[9] C. Pozna, F. Troester, R.-E. Precup, J. K. Tar, S. Preitl: On the design of an obstacle avoiding trajectory: method and simulation, Mathematics and Computers in Simulation, 2009, Vol. 79, No. 7, pp. 2211-2226

[10] T. Al-Shihabi, R. Mourant: A framework for modeling human-like driving behaviors for autonomous vehicles in driving simulators, in Proceedings of 5th International Conference on Autonomous Agents, Montreal, Canada, 2001, pp. 286-291

[11] L. Quispel, S. Warris, M. Heemskerk, L. J. M. Mulder, P. C. van Wolffelaar: Automan: a psychologically based model of human driver, in Clinical Assessment, Computerized Methods and Instrumentation, F. J. Maarse, A. E. Akkerman, A. N. Brand, and L. J. M. Mulder, Eds., Swets & Zeitlinger, Lisse, 2003, pp. 205-219

[12] R. Isermann: Diagnosis methods for electronic controlled vehicles, Vehicle System Dynamics, 2001, Vol. 36, No. 2-3, pp. 77-117

[13] K. Hrich: Optimization of emergency trajectories of autonomous vehicles with respect to linear vehicles dynamics, in Proceedings of IEEE/ASME 2005 International Conference on Advanced Intelligent Mechatronics, Monterrey, CA, 2005, pp. 245-252

[14] S. Galichet, L. Foulloy: Fuzzy controllers: synthesis and equivalences, IEEE Transactions on Fuzzy Systems, 1995, Vol. 3, No.2, pp. 140-148

[15] R.-E. Precup, S. Preitl, I. J. Rudas, M. L. Tomescu, J. K. Tar: Design and experiments for a class of fuzzy controlled servo systems, IEEE/ASME Transactions on Mechatronics, 2008, Vol. 13, No. 1, pp. 22-35

[16] R.-E. Precup, S. Preitl, J. K. Tar, M. L. Tomescu, M. Takács, P. Korondi, P. Baranyi: Fuzzy control system performance enhancement by iterative learning control, IEEE Transactions on Industrial Electronics, 2008, Vol. 55, No. 9, pp. 3461-3475

[17] L. Horváth, I. J. Rudas: Modelling and Solving Methods for Engineers, Elsevier, Academic Press, Burlington, MA, 2004

[18] P. Baranyi, L. T. Kóczy, T. D. Gedeon: A generalized concept for fuzzy rule interpolation, IEEE Transactions on Fuzzy Systems, 2004, Vol. 12, No. 6, pp 820-837

[19] Zs. Cs. Johanyák, S. Kovács: A brief survey and comparison on various interpolation based fuzzy reasoning methods, Acta Polytechnica Hungarica, 2006, Vol. 3, No. 1, pp. 91-105

[20] Y.-S. Lu: Smooth speed control of motor drives with asymptotic disturbance compensation, Control Engineering Practice, 2008, Vol. 16, No. 5, pp. 597-608

[21] K. Guesmi, N. Essounbouli, A. Hamzaoui: Systematic design approach of fuzzy PID stabilizer for DC-DC converters, Energy Conversion and Management, 2008, Vol. 49, No. 10, pp. 2880-2889

[23] C.-Y. Chen, T.-H. S. Li, Y.-C. Yeh: EP-based kinematic control and adaptive fuzzy sliding-mode dynamic control for wheeled mobile robots, Information Sciences, 2009, Vol. 179, No. 1-2, pp. 180-195

[24] S. Blažič, I. Škrjanc, S. Gerkšič, G. Dolanc, S. Strmčnik, M. B. Hadjiski, A. Stathaki: Online fuzzy identification for an intelligent controller based on a simple platform, Engineering Applications of Artificial Intelligence, 2009, Vol. 22, No. 4-5, pp. 628-638

[25] D. Hladek, J. Vaščák, P. Sinčák: Multi-robot control system for pursuit-evasion problem, Journal of Electrical Engineering, 2009, Vol. 60, No. 3, pp. 143-148

[26] M. Wu, J. Yan, J.-H. She, W.-H. Cao: Intelligent decoupling control of gas collection process of multiple asymmetric coke ovens, IEEE Transactions on Industrial Electronics, 2009, Vol. 56, No. 7, pp. 2782-2792

[27] M. Kalyoncu, M. Haydim: Mathematical modelling and fuzzy logic based position control of an electrohydraulic servosystem with internal leakage, Mechatronics, 2009, Vol. 19, No. 6, pp. 847-858