

Multi-Scenario Multi-Objective Optimization of a Fuzzy Motor Controller for the Szabad(ka)-II Hexapod Robot

István Kecskés, Péter Odry

University of Dunaújváros, Táncsics Mihály utca 1/A, 2401 Dunaújváros, Hungary; kecskesi@uniduna.hu, podry@uniduna.hu

Abstract: The aim of this research was to develop a robust motor controller for the Szabad(ka)-II hexapod robot. A Fuzzy-PI controller that utilized a lookup table was chosen because of its reported promising performance and its ability to be embedded in the microcontrollers of the robot. The variables of the controller were defined by a particle swarm optimization method to minimize the five quality objectives related to the walking of the robot. The preferences of the five objectives were successfully expressed by a biased-weighted geometric mean utility function. The resulting optimal solutions were significantly altered by changing the bias and exponential weights of preferences. Therefore, we checked the robustness of the solution against the controller's variables as a secondary objective.

Keywords: multi-scenario; multi-objective optimization; fuzzy control; robot dynamic simulation

1 Introduction

This work presents advanced research for control optimization related to the Szabad(ka)-II hexapod walker robot (Fig. 1), therefore, it is closely aligned with previous research results. The author's previous publication provides detailed information about the Szabad(ka)-II mechatronic device, its structure, sensors, motion requirements, modeling and control challenges.

This paper focuses on the issue of motion quality formulation, and the effects of the changing of preference weights between the objectives. Robust optimization is performed on a fuzzy-based motor controller, while walking quality is defined as a multi-scenario and multi-objective, in a specific simulation environment.

The Szabad(ka)-II is an 18 degree of freedom (DOF) hexapod walker robot. All the 18 joints are driven by 12 V DC gear motors. The embedded electronics executes the control algorithm real-time with a sampling frequency of 500 Hz.

The dynamic simulation model of the Szabad(ka)-II was developed and validated based on measurements performed on a real robot [1]. The cyclic trajectory data were generated offline specifically for the intended walking scenario. The trajectory curve of the tripod hexapod walking [2] has been researched and optimized together with a PI motor controller [3]. The optimization of the Fuzzy-PI motor controllers has also been researched [4].

In this study, the optimization of the motor controller differs from previous research in the following aspects:

- Developing and optimizing a Fuzzy-PI controller that can be embedded into real robot controllers (one Texas Instruments MSP430F2618 microcontroller for each leg). The fuzzy output is calculated by a previously generated lookup table, which has a constant number of dimensions and resolution, see details in [5].
- The simulation has multi-scenario properties. The multi-scenario approach is important to develop a universally optimal and robust motor controller for the intended use of the robot. See the details in Subsection 2.1.

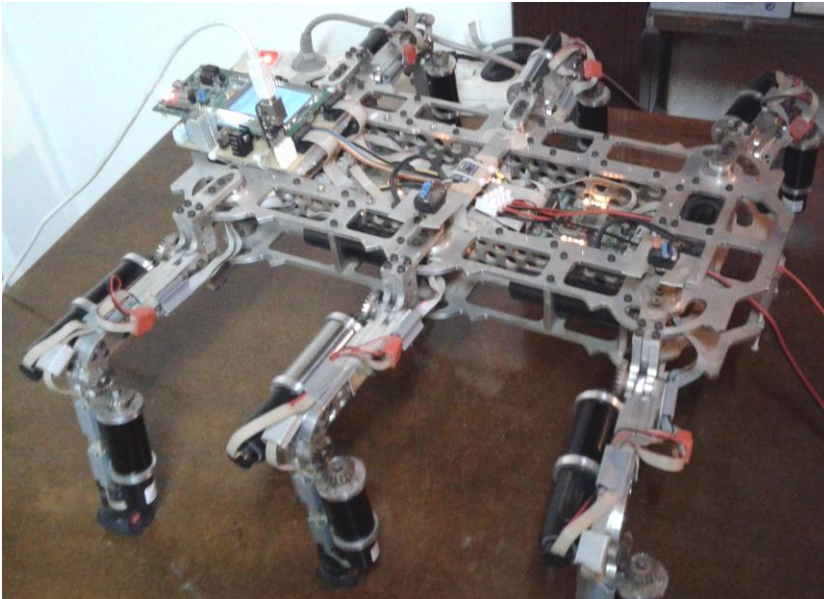


Figure 1
Szabad(ka)-II hexapod robot

The analyzed and optimized system is a multi-scenario multi-objective (MSMO) simulation system. These two properties are described in the second chapter. The third chapter describes the fuzzy-PI motor controller, and the fourth chapter summarizes the experimental results.

1.1 Multi-Objective Optimization Algorithms

Over the past decade, a number of multi- and many-objective evolutionary algorithms have been proposed. The most cited procedure is the Nondominated Sorting Genetic Algorithm (NSGA) [6] and its extended version, the NSGA-II method [7], which is currently the most widely read article in this field of science. The authors of the NSGA have provided two sets of test problems. Besides the popular NSGA series, there are several evolutionary multi-optimization search methods, such as Multi-Objective Particle Swarm Optimization (MOPSO) [8], Cuckoo Search [9], Ant colony optimization (ACO) [10], and Multi-objective Differential Evolution (MONDE) [11].

A multi-objective optimization process has several objective functions, and when searching for the optimum solution, the criteria involve finding the best fitness values while making compromises between the objectives. This topic has been addressed well in several studies [12] [13] [14].

The utility function is a specific type of objective function that is used to summarize (aggregate) the original objectives into a single figure of merit" [15] based on any predefined preferences - this is also called scalarization. When the aggregation of the objectives into a single utility function value can be defined unambiguously, then it represents the best solution [16].

One common approach is to combine the objective functions into a single function using weights, and the problem then becomes a matter of how to determine the appropriate weights [17]. However, in case of a real system optimization, it is often difficult to define the importance of these aspects relative to each other and, in addition, how they can be aggregated numerically.

1.2 Controller Multi-Objective Optimization

Structural and control optimization issues in complex dynamic systems are commonly multi-objective problems, as seen in hexapod robots [4] [18] [19], turbojet engines [20] [21], servo motor controllers [22] [23] [24], smart grids [25], train speed controllers [26], vehicle design [27], or building energy consumption problems [28].

Multi-objective optimization is commonly used for controllers where more than one quality objectives are defined. Adaptive PID and PI controllers are optimized using Multi-Objective Generic Algorithm in [22]. Cascaded DC motor controllers are optimized using multi-objective optimization evolutionary algorithms (MOEAs) selecting different solution on the Pareto-set in [29]. Optimal PID controllers are developed by NSGA-II algorithm, and compared to the traditional Ziegler Nichols methods in [30] and [23].

Complex, nonlinear and stochastic systems, such as, a hexapod robots or smart grids are frequently controlled by fuzzy or neuro-fuzzy systems. A multi-objective genetic algorithm is commonly used to find the optimal fuzzy controller, e.g., in [25] and [31] the set of solutions was found on the Pareto front. The results were compared using the same fuzzy controller optimized by a mono-objective genetic algorithm in [25]. Multi-objective Particle Swarm Optimization is also used for motor controller optimization [24].

In the cited optimization research group, the Pareto solution is determined without researching any further robustness or additional criteria to select the final solution. For example, in [30] the solution is selected based on only the primary objective from the Pareto set of two-objectives without any further investigation or explanation. In papers where weighted aggregation is performed to create single-objective from multi-objectives, the weights are not mentioned explicitly [26], or defined empirically.

1.3 Multi-Scenario in Optimization

A scenario in this context means a bunch of modifications in the input or parameterization of the system handled and optimized in parallel mode (e.g., a walker robot is walking, running, turning or sneaking). Scenarios are also referred to as 'situations' or 'modifications' in literature, but in event-based controlling, a scenario is not the same as an event.

The multi-scenario is similar to the multi-objective property. Commonly, the aggregation of scenarios is performed similarly to or together with the multi-objectives. The main purpose of the scenario-oriented approach is to be able to deal with a series of small multi-criteria design problems as opposed to a single large multi-criteria problem [32]. The increase of optimum robustness is the most common additional criteria aimed for in the multi-scenario approach, such as in [32] and [33]. In [34], the multi-scenario optimization was used to reach a robust optimum by dealing with uncertainty.

The MSMO approach has already been developed for theoretical problems [35] but it still appears as a new research area for issues related to motion quality of robot systems.

2 Multi-Scenario Multi-Objective Quality Definition of a Walker Robot

To the best of the authors' knowledge, there are no specific and applicable quality definitions for walker robot control problems. Practically, there are no definitions for the required quality aspects, nor is it clear to what extent the aspects should be

optimized. In our definitions, goodness is divided into simpler elements, which are established by common sense and empirical experience. Thus, the quality description is multi-objective.

2.2 Multi-objective Quality Definition

A multi-objective optimization process has several objective functions, and when searching for the optimum solution, the criteria involve finding the best fitness values while compromising between the objectives [12] using any preference between them.

The determination of the robot walking quality is not a trivial task. However, the most commonly seen criteria are the maximum traction and the minimum power consumption [36]: “In rough terrain, traction should be maximized. In benign terrain, power consumption should be minimized.”

The walking quality or driving quality definition is multi-objective. Previously, five quality objectives were defined for the Szabad(ka)-II robot walking [4]; see Table 1. In addition to the energy consumption and maximum walking speed, the vibration and jerks that appear in dynamics are addressed. The minimization of such high accelerations or rigid collisions is generally examined in robotics [37].

Table 1
Hexapod walking objectives

Quality Goal	Objective (to be minimize)	Symbol
achieve the maximum walking speed	the reciprocal value of mean velocity in the X direction	f_5
minimize the electrical energy consumption	electric energy consumption for walking one meter	f_4
minimize the torque on the joints and gears, thus minimizing the jerks in the motor current	root mean square of torque measured in the 18 joints	f_1
minimizing the robot's body acceleration in all three-dimensional directions	root mean square of magnitude of 3D body acceleration	f_2
minimizing the robot's body angular acceleration in all three-dimensional directions	root mean square of magnitude of 3D body angular acceleration	f_3

2.2 Multi-Objectives Aggregation

Some type of manual selection is required for the Pareto solution sets that result from the multi-objective optimizer algorithms [15], because only one solution at a time can be implemented in the real application.

The fitness values are calculated by aggregating these five ($M=5$) objectives by a so-called utility function. We propose a bias-weighted utility function and the production operation for the aggregation (geometrical mean). This function has a bias (b_m) and exponent (e_m) weights for each objective. The bias weights can reduce the strong influence of near zero values, while the exponent weights express the relative importance between the objectives. Equation (1) describes this utility function resulting in the scalar fitness value (f_{SC}) for one simulation scenario. The X represents the design variables, which are optimized by the optimizer algorithm.

$$f_{SC}(X) = \prod_{m=1}^M (b_m + f_m(X))^{e_m} \quad (1)$$

The well-defined quality formulation and proper weighting of the objectives are important [4]. In this study, these quality definitions are used. However, the preferences between these objectives are defined carefully and empirically. Different variations of these preferences are presented in Section 3.2.

2.3 Multi-Scenario Simulation

A driving solution is sought that provides robust and universally optimal behavior for all possible movements or walking tasks of the walker robot. The scenario-oriented approach offers an advantageous solution to this issue, as stated in the conclusion of [27]: The all-situation problem can thus be decomposed into several scenarios to form multiple objective functions, where these scenarios can be typical cases of all possibilities (generally an infinite number of situations). The main criteria in the selection of these targets and the determination of the number of scenarios should be a diversity of the required maneuvers as much as possible.

There is no guidance on how to select and how many scenarios are necessary, e.g., in [38], there are only two scenarios. The selected six typical scenarios for the Szabad(ka)-II robot demonstrate the possible intended use (which is just an example, because this robot was built for research purposes). In the selection of these scenarios, we considered the possible types of motions that the real robot can perform in the given laboratory conditions.

Table 2 lists the six scenarios and their parameters used for the optimization of the Szabad(ka)-II robot fuzzy controllers. The *load* means a real cargo, mounted on the robot body.

This ellipse-based leg trajectory was first published in [39]. The 3D leg trajectory curves are generated based on a half-ellipse. The *width*, the *stride*, the *height*, and the *radius* parameters are predefined or calculated from other scenario requirement parameters, such as *withers* or *turn*. These parameters differ for each scenario, as shown in Table 2.

Table 2
Parameters of six walking scenarios

Scenario description				Trajectory parameters			
ID	Gait	Speed	Load	Turn	Time	Radius	Withers
1.	Tripod normal	Fast	0 kg	0	1.5 s	0.16 m	0.15 m
2.	Tripod normal	Slow	0 kg	0	2.2 s	0.16 m	0.15 m
3.	Tripod normal	Fast	2 kg	0	1.5 s	0.145 m	0.20 m
4.	Tripod normal	Slow	2 kg	0	2.2 s	0.145 m	0.20 m
5.	Tripod slope	Fast	0 kg	0	1.5 s	0.20 m	0.14 m
6.	Turn right	Fast	0 kg	0.5	1.5 s	0.16 m	0.15 m

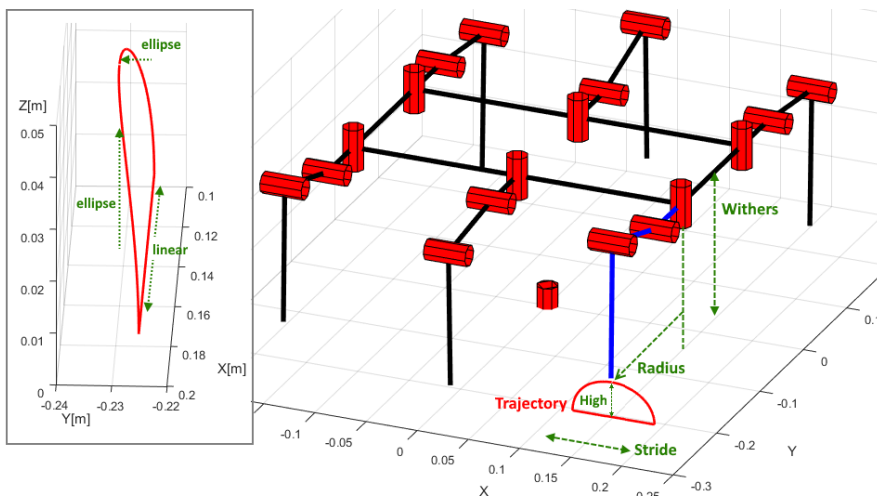


Figure 2

Ellipse based leg trajectory for the tripod hexapod walking of the Szabad(ka)-II robot

Each of the six legs received the same curves with inverted phases according to the tripod walking. This curve is adjusted only with the parameter *turn* if the robot performs a turn. The joint trajectories are calculated from this leg trajectory using inverse kinematics, which was described in a previous publication [1].

2.4 Multi-Scenario Aggregation

Parallel execution of these scenarios ($K=6$) provides a multi-scenario objective function that is primarily a multi-objective function. Multi-scenario problems are regularly solved by aggregating all the objectives of all scenarios into a large multi-criteria problem, which is confirmed by previous studies [27], [35]. The global scalar fitness values (f_G) are calculated with the geometrical mean applied for the scenario's fitness values (f_{SC}); see equation (2).

$$f_G(X) = \sqrt[k]{\prod_{k=1}^K f_{SC}(X, k)} \quad (2)$$

2.5 Simulation Model

The kinematic model describes the movement, while the dynamic model shows the forces and torque effects on the robot body and engine, as well as the electrical activity of the motor. The kinematic and dynamic models are essential for the effective development of robots, especially if the controllers are under research, which is confirmed by several of the studies [40], [41], [19].

The current simulation model was described in paper [1] (it includes a detailed kinematic and dynamic model of the real Szabad(ka)-II robot). This model includes the kinematics and dynamics of the 18 DOF robots, model of the DC motor and gearboxes, model of the PWM amplifiers, model of the encoders and current sensors, model of the ground contacts, and model of the controllers embedded in the robot electronics. Fig. 3 illustrates the simulation model implemented in Simulink.

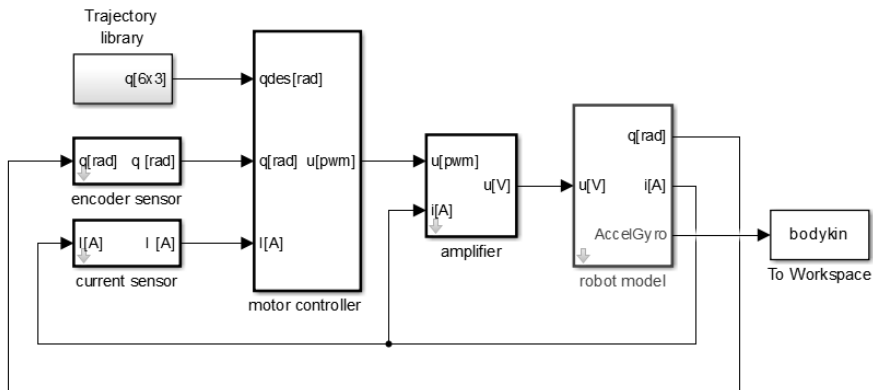


Figure 3

Simulink model of the Szabad(ka)-II robot – the root level. More details are given in [1]

2.6 Optimization Algorithm

In previous research [4], the examined motor controllers were optimized together with the parameters of the leg trajectory of the Szabad(ka)-II robot. The particle swarm optimization (PSO) method was chosen from 12 heuristic optimization methods through a benchmark-based selection and with the help of specific test functions [4]. The applicability of swarm-based optimizations of the hexapod robot structure and its walking are summarized in reference [42].

Therefore, in this study, we used the already developed algorithm of the PSO (original implementation available in [43]). Fig. 4 illustrates the block diagram of the implemented optimization system in a Matlab/Simulink environment. This implementation is capable of parallelizing iterations, storing iteration results immediately after its calculation and analyzing the evolution during the work. These functionalities are especially developed for long term optimizations (when calculations last more than days or weeks).

The Matlab code is available in webpage [44].

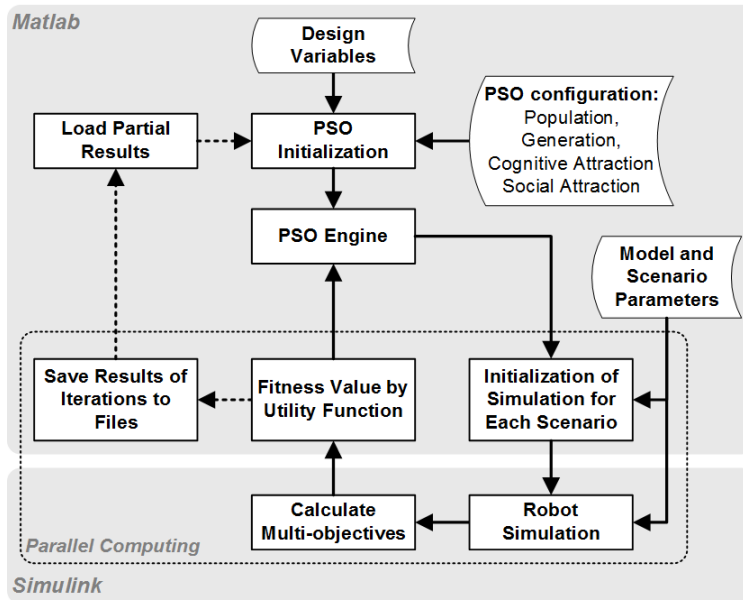


Figure 4

Block diagram of the PSO optimization system in a Matlab/Simulink environment for the MSMO robot simulation

3 Fuzzy-PI Motor Controller

The main advantage of the Fuzzy Logic System is that it can extract heuristic rules that contain if-then statements from human experience [38]. Fuzzy Logic Systems are introduced to learn the behaviors of the unknown dynamics of the robot and wheel actuators due to their universal approximation properties. In this way, the external disturbances and approximate errors can be efficiently counteracted by employing smooth, robust compensators [45].

The fuzzy controller can provide a more comprehensive solution compared to the PID controller [46]. This is confirmed by our previous studies:

- A fuzzy-PI motor controller with three input variables was constructed and compared with a previously used PI controller for the Szabad(ka)-II walker robot [4].
- A fuzzy route controller was introduced and compared with a simple PID route controller [40].
- A fuzzy-I motor controller was developed and optimized to ensure better control performance to protect the Szabad(ka)-II walker robot's electro-mechanical equipment against high peaks or jerks, and was compared to a PID controller. [41]

3.1 Motor Controller of Szabad(ka)-II Robot

In this context, the fuzzy-PI controller type is a PI controller, where the P - proportional tag is defined by a Fuzzy Logic Controller (fuzzy); see Fig. 5. This control system includes the following:

- The fuzzy controller is implemented as a lookup table (LUT), published in [5]. Therefore, its name became "fuzzy LUT" in this context. This controller has two inputs: the angle error and the motor current.
- Each motor current is measured by the robot's microcontroller with a 12-bit resolution AD converter.
- The desired joint angles are predefined (see Section 2.3) and sent from a Matlab program implemented in the PC client side.
- The measured joint angles are calculated based on an encoder sensor mounted on the motor.
- The I integrator tag's output is added to the P proportional tag and results in the control voltage. This voltage drives a PWM amplifier with a 10-bit resolution DA converter.

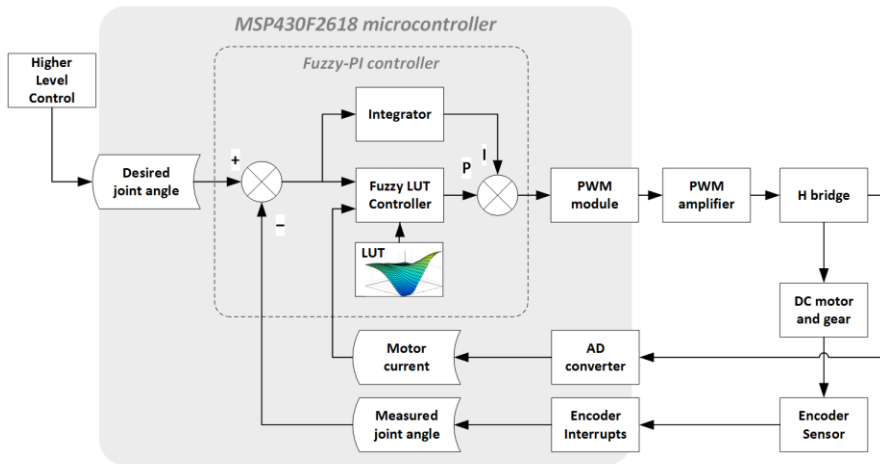


Figure 5

Block diagram of the Fuzzy-PI motor control design and implemented for 18 joints of the Szabad(ka)II robot

3.2 Fuzzy Controller

The aim of the fuzzy-P controller is the same as the proportional tag of a traditional PID controller. However, this fuzzy controller is capable of taking into account the motor current and ensuring softer behavior for high motor currents. Moreover, when the motor current is extremely high, inverse output can be ensured to protect the electro-mechanical system. These requirements are represented by the six fuzzy rules; see Table 3 and Fig. 6.

Fig. 7 shows the surface that is established by the proposed rules, which will be transformed to a LUT in the embedded implementation. It demonstrates the nonlinear control behavior along the two input variables.

Table 3
Rules of the proposed fuzzy-P controller

Rules					Comment
	<i>ErrorAngle</i>	<i>MotorCurrent</i>	<i>Voltage</i>	<i>Weight</i>	
1.	Zero	-	Null	0.5	direct P controlling rules for normal behavior
2.	Pos	-	Pos	1	
3.	Neg	-	Neg	1	
4.	Zero	small	Null	0.5	inverse P controlling rules for protection again high motor current
5.	Neg	large	pos-ex	1	
6.	Pos	large	neg-ex	1	

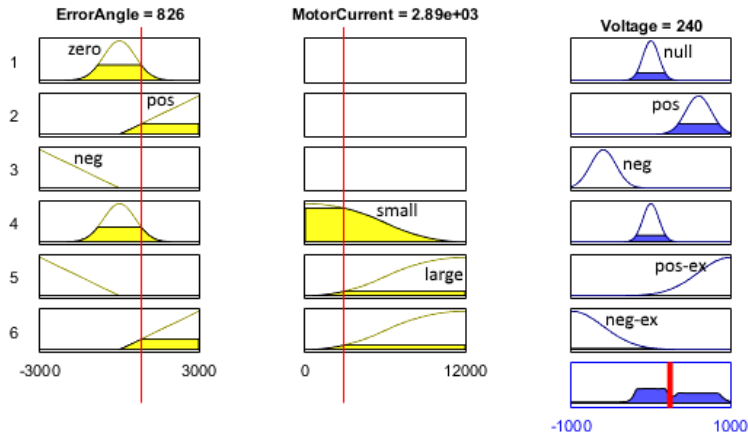


Figure 6

Demonstration of the rules of the proposed fuzzy-P controller

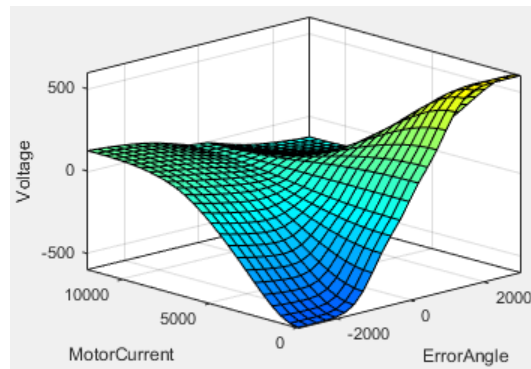


Figure 7

Surface of the proposed fuzzy-P controller – the basis of the fuzzy lookup table in the embedded implementation

This kind of controller was previously tested under extreme mechanical situations [47] and proposed to protect the robot in such situations. An adaptive control mechanism is proposed in [47] by changing the rule's weights in the fuzzy controller: "The suggested solution of mechanism control lies in the turning on or turning off of some membership functions in the fuzzy control. Changing the weight of the rules in the control algorithm we can modify the characteristics of the controller so as to be optimal in the case of drop test and walking as well."

In this study, the weights of the fuzzy rules are optimized by the PSO to increase the multi-objective walking quality during multi-scenarios.

3.3 Design Variables

In this context, the parameters that are changed by the optimizer algorithm are called design variables, and the other parameters that influence the objectives but are not changed by the optimizer are called design parameters. In this case, there are some constant design parameters, and there are some that differ between the scenarios (considered as scenario design parameters).

In this study, the optimal motor controller is searched for by the previously designed leg trajectories and walking algorithm. The fitness function is multi-objective as introduced in Section 1.1.

Table 3 lists the selected design variables ($N=9$) related to the fuzzy-PI motor controller. The minimum and maximum values are selected empirically and based on the previous experience given in [4]. The symmetric rules (2-3 and 5-6) are handled together as proposed by [4]. The unit of inputs and outputs are in integer coded format inherited from the ADC and DAC, but the transfer multipliers are mentioned in Table 4 in the Unit and Domain column. The fuzzy output membership function domain includes three functions that are convertible to each other without adding or removing any parameters (triangular – *trimf*, gaussian – *gaussmf*, and Π -shaped membership function - *pimf*). This is important in the optimization algorithm, for a constant number of design variables.

Table 4
Design variables – selected fuzzy controller parameters to be optimized

Abbr.	Variable Description	Min. Value	Max Value	Unit and Domain
I	Integrator tag	0.1	1	V/rad
FI1R	Fuzzy input 1 (<i>Aerr</i>) range	1500	6000	Rad/10430
FI2R	Fuzzy input 2 (<i>Im</i>) range	6000	24000	A/2079
FOR	Fuzzy output 1 (<i>P</i>) range	500	2000	V/(511/11.3)
FOMF	Fuzzy output membership function	1	3	1: <i>trimf</i> 2: <i>gaussmf</i> 3: <i>pimf</i>
FW1	Fuzzy rule 1 weight	0.1	1	
FW23	Fuzzy rule 2 and 3 weight	0.1	1	
FW4	Fuzzy rule 4 weight	0.1	1	
FW56	Fuzzy rule 5 and 6 weight	0.1	1	

4 Results

4.1 Optimization Results

The PSO method was applied to increase the MSMO walking quality of the Szabad(ka)-II robot by searching for the best motor controller. The MSMO fitness evaluation and aggregation were described in Chapter 2. The design variables of the motor controller and their boundaries were defined in Section 3.3.

The PSO algorithm has its own parameters, which were defined based on previous experiences [4], [42], [3]. These parameters include the cognitive attraction of 0.5 , Social Attraction of 1.5 , generation number of 25 , and the population number of 25 . However, in this phase the population and generation numbers were set relatively small compared to the final optimization instance. Here, the aim was to research the method, to make the multiple runs of the optimization faster during the development, and to run one final larger optimization for the real implementation at the end. I.e. the larger sized optimization reach the global optimum with higher probability [48], but not obligatory [49].

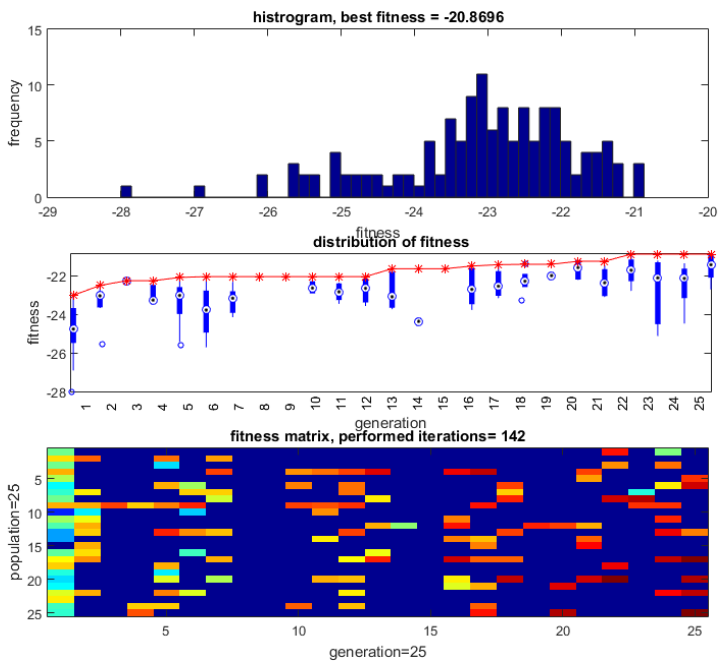


Figure 8

Statistical results of the PSO (case A) for the fuzzy-PI controller evaluated by the MSMO approach. The top graph shows the distribution of the global fitness values (f_G) occurring in the optimization, the middle graph shows its distribution for each generation, and the bottom graph shows its distribution over generations and populations

Fig. 8 graphically shows the statistical analysis of the first optimization (A case). It confirms that the PSO during the generations continuously found a better solution, as it the maximum curve (red curve) illustrates in the middle graph. On the other hand, there is no proof that the best solution from the tested 142 iterations - considered “optimum” - is the real global optimum. However, the global optimum within a weaker tolerance is expected based on the previous research [4].

The given optimum is given in Table 6, while the fuzzy controller surface for the A case is illustrated in Fig. 8.

The preference weights for the MSMO fitness evaluation according to equation (1) are listed in Table 4 (A case). The explanations for these preference weights are described in next subsection.

4.2 Various Preferences of Multi-Objective

There is no universal guidance for predefining the preferences between the objective methods as introduced in Chapter 2. In this section, we analyze the effect of changing the preferences to the optimum values.

Table 5 lists the tested preference values, where the preferences for the A and B cases are generated randomly, while the C case preferences are set manually to default values (bias $b=0$, exponent $e=1$).

Table 5
Multi-objective preference changes by modifying the weights of the utility function

Utility Function Case	Bias weight of utility function (b_m)	Exponent weight of utility function (e_m)
A	[0.9 0.8 0.24 0.5 0.8]	[0.1 0.7 0.3 0.3 0.1]
B	[0.4 0.7 0.3 0.4 1]	[0.2 0.6 0.8 0.1 0.2]
C	[0.0 0.0 0.0 0.0 0.0]	[1.0 1.0 1.0 1.0 1.0]

Table 6 lists the given optimum for these three cases. We can observe significant differences between the cases for most of the design variables.

Table 6
Calculated optimums of the design variables by the three different multi-objective preferences (note: global fitness values are not comparable between the cases because of the different utility functions)

Design Variables	Result A	Result B	Result C
I	0.171	0.160	0.607
FI1R	5837	5400	5629
FI2R	20111	18693	18242
FOR	781	500	527
FOMF	2 (gaussmf)	3 (pimf)	3 (pimf)

FW1	0.187	0.347	0.581
FW23	0.961	0.972	0.889
FW4	0.789	0.592	0.702
FW56	0.927	0.814	0.188
<i>Global fitness (f_G)</i>	<i>20.8696</i>	<i>59.5132</i>	<i>1396908</i>

Fig. 9 shows the fuzzy surfaces for these three optimums. The observable significant differences of these surfaces also confirm that the different preference weights lead to different solutions.

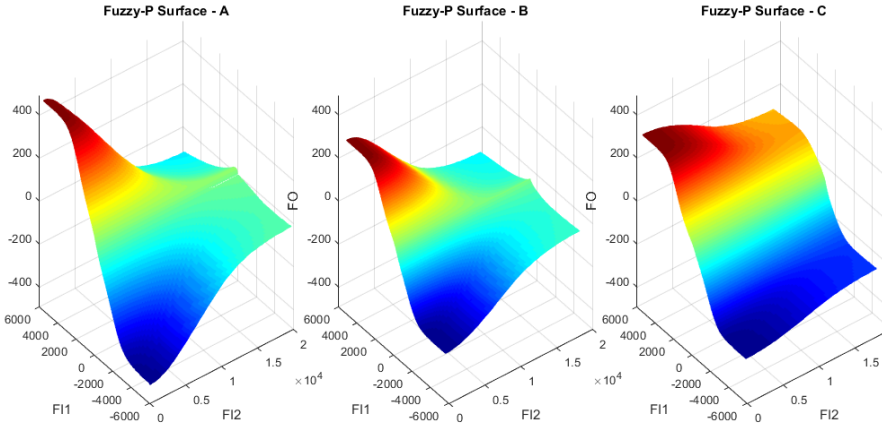


Figure 9

Optimal fuzzy-P surfaces for the A, B, and C optimization cases

4.3 Robustness Comparison

Robust optimization refers to the process of finding optimal solutions for a particular problem that have the least variability to probable uncertainties [50]. The robustness index introduced by [51] is used to evaluate and compare our optimums, because the robustness is aimed as the secondary performance metric in addition to the primary multi-objectives, similar to in [28].

Robustness index (Ri) is determined to be the largest eigenvalue of the sensitivity Jacobian matrix ($J_k \in R^{N \times M}$), which calculated between the design variables ($X \in R^N$) and the objective function values ($F \in R^M$, $F = \{f_1, \dots, f_M\}$), for one scenario (k), see equation 3.

$$Ri(X, k) = \max \left(\sqrt{\text{diag}(J_k \cdot J_k^T)} \right) = \max \left(\sqrt{\text{diag} \left(\frac{\partial F}{\partial X} \cdot \frac{\partial F^T}{\partial X} \right)} \right) \quad (3)$$

The robustness indices are calculated for each of the six scenarios in each of the three cases. The values and distribution of these robustness indices are illustrated in the left panel of Fig. 10. The smaller values represent more robust or less sensitive optimums for the changes of the design variables. Solution A shows a slightly higher median value (red line) but with the lowest maximum, solution B has the highest maximum, and solution C has the lowest median value. It is difficult to select the most robust solution because the robustness properties deviate between the scenarios.

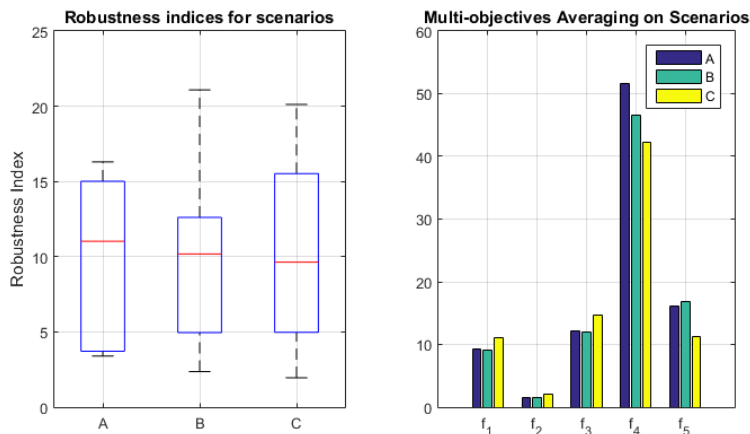


Figure 10

Comparison statistics of the three optimization cases: the left graph shows the distribution of the robustness indices (lower is more robust), the right graph shows the average values of the multi-objectives (lower is better)

The right panel of Fig. 10 illustrates the how the five objectives (defined in Table 1) differ between the three cases. The plotted values are averaged over the six scenarios. This analysis confirms that the preferences weights are influenced by the relation between the objectives. For example, in case A, the exponent for f_4 was small ($e_4=0.1$), while it was high in case C ($e_4=1.0$), furthermore f_4 shows a better value in case C.

Conclusions

We studied the issues related to the heuristic optimization of motor controllers for a walker robot, using a dynamic simulation model. Defining and quantifying the quality of the hexapod robot walking as a multi-objective problem and how to aggregate the multi-objectives into a scalar fitness value using preference weights were explored. Integrating the multi-scenario simulation approach was also examined in this optimization system.

The optimization results (three example solutions of the fuzzy-PI controller) show high divergences between the optimums for defining different preferences between the objectives. The preferences are implemented in a utility function with bias and exponent pair weights for each objective.

The manual selection of these weights opens another optimization issue, which we believe, is the important part of the entire system. The sensitivity or robustness analysis can be used as an external quality aspect to select the appropriate preferences. The method proposes an automatic definition of the preference weights by the optimum robustness against the design variables, design parameters or multi-scenarios.

This robust optimization approach could be applied, not only to mobile robots (legged, wheeled or train structures), but in other fields of engineering systems, such as, optimization problems of smart-grids, turbo-jet engines, or bridge structures. In the aforementioned systems, quality can be defined using more objectives; moreover, the intended use could both include additional scenarios and be simulated with multi-scenarios. The robustness of optimum against situation variability can be considered as a secondary objective, since in reality, several scenarios may occur.

Acknowledgements

This work/publication is supported by the EFOP-3.6.1-16-2016-00003 project and is co-financed by the European Union.

References

- [1] I. Kecskés, E. Burkus, F. Bazso, P. Odry: "Model validation of a hexapod walker robot", *Robotica*, Vol. 35, No. 2, 2017, pp. 419-462, DOI: 10.1017/S0263574715000673
- [2] D. Grzelczyk, et al.: "Kinematics, Dynamics and Power Consumption Analysis of the Hexapod Robot During Walking with Tripod Gait", *International Journal of Structural Stability and Dynamics*, Vol. 17, No. 9, (17 pages), 2017, DOI: 10.1142/S0219455417400107
- [3] I. Kecskés, L. Szekacs, J. C. Fodor, P. Odry: "PSO and GA optimization methods comparison on simulation model of a real hexapod robot", *IEEE 9th International Conference on Computational Cybernetics (ICCC)*, 2013, pp. 125-130, DOI: 10.1109/ICCCyber.2013.6617574
- [4] I. Kecskés and P. Odry: "Optimization of PI and Fuzzy-PI Controllers on Simulation Model of Szabad (ka)-II walking robot", *Int. J. Adv. Robot. Syst.*, Vol. 11:186, 2014, DOI: 10.5772/59102
- [5] I. Kecskés, L. Székács, P. Odry "Lookup table based fuzzy controller implementation in low-power microcontrollers of hexapod robot Szabad (ka)-II." *3rd International Conference & Workshop Mechatronics in Practice and Education–MECHEDU*, 2015

-
- [6] Srinivas Nidamarthi and Kalyanmoy Deb: "Multiobjective optimization using nondominated sorting in genetic algorithms", *Evolutionary Computation*, Vol. 2, No. 3, 1994, pp. 221-248
- [7] Kalyanmoy Deb, et al.: "A fast and elitist multiobjective genetic algorithm: NSGA-II", *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 2, 2002, pp. 182-197
- [8] Konstantinos E. Parsopoulos, Michael N. Vrahatis: "Multi-objective particles swarm optimization approaches", *Multi-objective optimization in computational intelligence: Theory and practice, Chapter II. IGI Global*, 2008, pp. 20-42
- [9] Alexander Teske, Rafael Falcon, Amiya Nayak: "Efficient detection of faulty nodes with cuckoo search in t-diagnosable systems", *Applied Soft Computing*, Vol. 29, 2015, pp. 52-64
- [10] Ines Alaya, Christine Solnon, Khaled Ghedira. "Ant colony optimization for multi-objective optimization problems", *19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2007) 2007*, DOI: 10.1109/ICTAI.2007.108
- [11] Awad Noor H., Mostafa Z. Ali, Rehab M. Duwairi: "Multi-objective differential evolution based on normalization and improved mutation strategy", *Natural Computing*, Vol. 16, No. 4, 2017, pp. 661-675
- [12] J. Brank, et al. eds.: "Multiobjective optimization: Interactive and evolutionary approaches", *Theoretical Computer Science and General Issues, Springer Science*, Vol. 5252, 2008, DOI: 10.1007/978-3-540-88908-3
- [13] R. T. Marler, J. S. Arora: "Survey of multi-objective optimization methods for engineering", *Structural and multidisciplinary optimization*, Vol. 26, No. 6, 2004, pp. 369-395
- [14] Eckart Zitzler: "Evolutionary algorithms for multiobjective optimization: Methods and applications", Vol. 63, Ithaca: Shaker (1999)
- [15] Andrew L. Nelson, Gregory J. Barlow, Lefteris Doitsidis: "Fitness functions in evolutionary robotics: A survey and analysis." *Robotics and Autonomous Systems*, Vol. 57, No. 4, 2009, pp. 345-370
- [16] R. Timothy Marler, Jasbir S. Arora: "Survey of multi-objective optimization methods for engineering", *Structural and multidisciplinary optimization*, Vol. 26, No. 6, 2004, pp. 369-395
- [17] Michael A. Gennert, Alan L. Yuille: "Determining the optimal weights in multiple objective function optimization", *Second International Conference on Computer Vision (ICCV) 1988*, DOI: 10.1109/CCV.1988.589974

-
- [18] L. Kubler, C. Henninger, P. Eberhard: "Multi-criteria optimization of a hexapod machine", *Advances in Computational Multibody Systems*, Volume 14, Issues 3-4, pp. 225-250, DOI:10.1007/s11044-005-4311-z
- [19] E. Burkus, J. C. Fodor, P. Odry, "Structural and gait optimization of a hexapod robot with particle swarm optimization", *IEEE 11th International Symposium on Intelligent Systems and Informatics (SISY) 2013*, pp. 147-152 DOI:10.1109/SISY.2013.6662559
- [20] Rudolf Andoga, et al.: "Innovative approaches in modeling control and diagnostics of small turbojet engines", *Acta Polytechnica Hungarica*, Vol. 10, No. 5, 2013, pp. 81-99
- [21] Hanz Richter: "A multi-regulator sliding mode control strategy for output-constrained systems", *Automatica* Vol. 47, No. 10, 2011, pp. 2251-2259
- [22] Moulay Rachid Douiri, Mohamed Cherkaoui: "Evolutionary Multi-objective Optimization Based Proportional Integral Controller Design for Induction Motor Drive", *Multi-disciplinary Trends in Artificial Intelligence, (MIWAI 2012), Lecture Notes in Computer Science*, Vol. 7694. Springer, Berlin, Heidelberg, 2012
- [23] N. Katal, S. K. Singh: "Multi-objective optimisation of PID controller for DC servo motor using genetic algorithm", *Eng. Intell. Syst.*, Vol. 23, No. 1, 2015, pp. 7-16
- [24] Nasseer K. Bachache, Jinyu Wen: "Multi Objective Swarm Optimization Design Fuzzy Controller to Adjust Speed of AC Motor Drive", *International Conference in Swarm Intelligence*, Springer, Berlin, Heidelberg, 2013
- [25] Stefano Leonori, et al.: "Multi objective optimization of a fuzzy logic controller for energy management in microgrids", *IEEE Congress on Evolutionary Computation (CEC) 2016*, DOI: 10.1109/CEC.2016.7743811
- [26] Longda Wang, et al.: "Multi-objective Optimization Improved GA Algorithm and Fuzzy PID Control of ATO System for Train Operation", *Intelligent Computing, Networked Control, and Their Engineering Applications, (ICSEE 2017), (LSMS 2017) Communications in Computer and Information Science*, Vol. 762. Springer, Singapore 2017
- [27] G. Fadel, I. Haque, V. Blouin, M. Wiecek, "Multi-criteria multi-scenario approaches in the design of vehicles", *6th World Congresses of Structural and Multidisciplinary Optimization (2005)* DOI:10.1.1.74.3818
- [28] R. Marijt: "Multi-objective robust optimization algorithms for improving energy consumption and thermal comfort of buildings", *Master's thesis*, Leiden Institute for Advanced Computer Science, Leiden University, 2009 <http://liacs.leidenuniv.nl/assets/Masterscripties/2009-20RobertMarijt.pdf>

- [29] L. F. Wang: "Controller design for DC motor drives using multi-objective optimization evolutionary algorithms", Proceedings of the 2004 International Symposium on Micro-Nanomechatronics and Human Science and The Fourth Symposium Micro-Nanomechatronics for Information-Based Society, 2004, DOI: 10.1109/MHS.2004.1421273
- [30] Sanjay Kr Singh, et al.: "Multi-objective optimization of PID controller for temperature control in centrifugal machines using genetic algorithm", *Research Journal of Applied Sciences, Engineering and Technology*, Vol. 7, No. 9, 2014, pp. 1794-1802
- [31] Yazmin Maldonado, Oscar Castillo, Patricia Melin: "A multi-objective optimization of type-2 fuzzy control speed in FPGAs", *Applied Soft Computing*, Vol. 24, 2014, pp. 1164-1174
- [32] Margaret M. Wiecek, Vijay Singh, Vincent Blouin: "Multi-scenario multi-criteria optimization in engineering design", *Report Documentation Page*, Clemson Univ Sc Dept Of Mathematical Sciences, Form Approved: OMB No. 0704-0188, 2007
- [33] Francesco Rossi, et al.: "Multi- scenario robust online optimization and control of fed- batch systems via dynamic model- based scenario selection" *AIChE Journal*, Vol. 62, No. 9, 2016, pp. 3264-3284
- [34] Marco C. Campi, Simone Garatti, Maria Prandini: "The scenario approach for systems and control design", *Annual Reviews in Control*, Vol. 33, No. 2, 2009, pp. 149-157
- [35] Ling Zhu, Kalyanmoy Deb, Sandeep Kulkarni: "Optimization of Multi-Scenario Problems Using Multi-Criterion Methods: A Case Study on Byzantine Agreement Problem", IEEE Congress on Evolutionary Computation (CEC 2014) 2014, DOI: 10.1109/CEC.2014.6900637
- [36] Karl Iagnemma, Steven Dubowsky: "Traction Control of Wheeled Robotic Vehicles in Rough Terrain with Application to Planetary Rovers", *The International Journal of Robotics Research*, Vol. 23, No. 10-11, 2004, pp. 1029-1040
- [37] G. Carbone: "Stiffness analysis and experimental validation of robotic systems", *Frontiers of Mechanical Engineering* vol Vol. 6, No. 2, 2011, pp. 182-196
- [38] Irfan Ullah et al: "Integrated tracking and accident avoidance system for mobile robots", *International Journal of Control Automation and Systems*, Vol. 11, No. 6, 2013, pp. 1253-1265
- [39] Zoltan Pap, Istvan Kecskes, Ervin Burkus, Fulop Bazso, Peter Odry: "Optimization of the hexapod robot walking by genetic algorithm" *IEEE 8th International Symposium on Intelligent Systems and Informatics (SISY)* 2010, DOI: 10.1109/SISY.2010.5647242

-
- [40] I. Kecskés, P. Odry: “Fuzzy Route Control of Dynamic Model of Four-Wheeled Mobile Robot”, *4th IEEE International Symposium on Logistics and Industrial Informatics (LINDI)* 2012, pp. 215-220, DOI: 10.1109/LINDI.2012.6319490
- [41] I. Kecskés, E. Burkus, Z. Király, Á. Odry, P. Odry: “Competition of motor controllers using a simplified robot leg PID vs fuzzy logic”, *4th International Conference on Mathematics and Computers in Sciences and Industry (MCSI)* Corfu, 2017, DOI: 10.1109/MCSI.2017.16
- [42] I. Kecskés, E. Burkus, P. Odry: “Swarm-based optimizations in hexapod robot walking”, in: *IEEE 9th International Symposium on Applied Computational Intelligence and Informatics (SACI)* 2014, pp. 123-127, DOI:10.1109/SACI.2014.6840048
- [43] Particle swarm toolbox for MATLAB, <http://code.google.com/p/psomatlab/>
- [44] Matlab code of MSMO optimization framework, <http://appl-dsp.com/wp-content/uploads/2014/03/optimization-methods.zip>
- [45] Maurizio Melluso: “A Theoretical and Experimental Approach of Fuzzy Adaptive Motion Control for Wheeled Autonomous Nonholonomic Vehicles”, *ISRN Robotics*, Volume 2013, 2013, 18 pages, Article ID 621067, DOI: 10.5402/2013/621067
- [46] Á. Odry, J. Fodor, and P. Odry: “Stabilization of a two-wheeled mobile pendulum system using LQG and fuzzy control techniques”, *International Journal On Advances in Intelligent Systems*, Vol. 9, No. 1-2, 2016, pp. 223-232
- [47] I. Kecskés, P. Odry: "Protective Fuzzy Control of Hexapod Walking Robot Driver in Case of Walking and Dropping." *Computational Intelligence in Engineering*, Springer Berlin Heidelberg, 2010, pp. 205-217, DOI: 10.1007/978-3-642-15220-7_17
- [48] Stanley Rylander, Gotshall Bart: "Optimal population size and the genetic algorithm", *Population*, Vol. 100, No. 400, 2002, p. 900
- [49] Tianshi Chen, et al.: "A large population size can be unhelpful in evolutionary algorithms" *Theoretical Computer Science*, Vol. 436, 2012, pp. 54-70
- [50] S. Mirjalili, A. Lewis: “Novel frameworks for creating robust multi-objective benchmark problems”, *Information Sciences*, Vol. 300, 2015, pp. 158-192, DOI:10.1016/j.ins.2014.12.037
- [51] O. B. Augusto, F. Bennis, S. Caro, “Multiobjective engineering design optimization problems: a sensitivity analysis approach”, *Pesquisa Operacional*, Vol. 32, No. 3, 2012, pp. 575-596, DOI: 10.1590/S0101-74382012005000028