Robust solution
What does it mean and why should we deal with it?

PETER ODRY *, IMRE RUDAS **
* Institute of Informatics
University of Dunaújváros
Táncsics Mihály u. 1/A Pf.: 152, 2401. Dunaujváros
HUNGARY
podry@uniduna.hu

** Antal Bejczy Center for Intelligent Robotics
Óbuda University
Kiscelli utca 82, H-1032, Budapest
HUNGARY
rudas@uni-obuda.hu

Abstract: - In the implementation phase of a task, we expect it to be as reliable and safe as possible with the best possible parameters, regardless of operating conditions; for example, if we build a structure, or write a software, or simply want to get from one point to another in a big city for a given time, etc. Operation under these conditions is called robust operation. Finding a robust solution is one of the key strategic issues in today's accelerated world. There is no time today to develop a tool, to build it and test it and then modify it, then apply this cycle several times in every possible load environment, but we are looking for mathematical methods to solve this optimization process in a simulation environment. The winning strategy is not simply about the selection of the optimization method, but also about the definition of the adequate quality (fitness), the robustness of the resulting optimum or sensitivity analysis, and the uncertainty analysis of several parameters.
Modern engineering problems are often composed by objectives that must be taken into account simultaneously for better design performance. Normally, these objectives are conflicting, i.e., an improvement in one of them does not lead, necessarily, to better results for the other ones. To overcome this difficulty, many methods to solve multi-objective optimization problems (MOP) have been proposed. The simulation model includes environmental or mission parameters that are not part of the parameters to be optimized but their variation creates different scenarios. A multi-scenario simulation can be created with the typical values of these parameters where the optimum is searched for all scenarios at the same time. Such optimum is more robust than one achieved through a process using separate scenarios since the intended use of the robot is represented by the multi-scenarios. A common goal for system design is robustness: the ability of a system to operate correctly in various conditions and fail gracefully in an unexpected situation. This paper deals with two different research domains, where the goal of finding the robust solution is presented. First, Szabad(ka)-II hexapod walker robot as a complex mechanical structure characterized by three motors per leg is analyzed. This robot is particularly suitable for testing the robustness of the closed loop system. During the design process, the minimization of the mechanical complexity was carefully addressed with the aim to reduce the unwieldy appearance. In the second part of the paper, robust methods applied in tomography are discussed. Achieving robustness in tomographic methods is very important due to the presence of different measurement noises. The applied inverse calculation methods are sensitive to noise effects, moreover, the obtained results are characterized by uncertainties due to the unknown dynamics of different noise sources.

Key-Words: - optimal solution, linear and heuristic optimalization, robust solution, robotics, tomography

1 Introduction

Robust optimization (RO) is a young research field, the basic parts originate in 1974. Optimal solution had two types, first the global optimum- this is a solution to the overall optimization problem. Its objective value is as good as any other point in the feasible region. Unlike to the local optimum, it is optimal only with respect to feasible solutions close to that point. Points far removed from a local optimum play no role in its definition and may actually be preferred to the local optimum. In the process of optimization, the aims are to minimize the mean response, but usually the variation is not minimized. Such a design method does not necessarily correspond to the optimum design in terms of safety and reliability. In order to obtain a more realistic optimum design a robust design (RD) must be adopted, which optimizes a performance index expressed in terms of mean value. At the same time, it minimizes its variability resulting from environmental uncertainty, this solution is less sensitive to the variation of parameters.

Exploring the difference between notions „stable" and „robust" touches essentially every aspect about robustness in nature, engineering, and social systems. It is argued here that robustness is a measure of feature persistence in systems that compels us to focus on perturbations.

What's the difference between stable and robust? It's the first question that comes to mind, for researchers and developers, who work with quantitative models. The robustness is a measure of feature persistence in systems where the perturbations to be considered are not fluctuating in external inputs or internal system parameters. But instead it represent changes in system composition, system topology, or in the fundamental assumptions regarding the environment in which the system operates, until in stability theory to postulate a single perturbation; from the robustness perspective it is often ineluctably necessary to consider instead multiple perturbations in multiple dimensions.

Robust optimization is a mathematical discipline that takes exactly uncertainties in the problem parameters into account - by finding solutions that are still "good", when things happen to turn out in other way as we expected. The quality depends on the applied robustness concept.

In general, quality control needs to respond to a number of requirements (such as low power consumption, accuracy, speed, battery saving), so the system is multi-objective. The simulation model includes environmental or mission parameters that are not part of the parameters to be optimized but their variation creates different scenarios. A multi-scenario simulation can be created with the typical values of these parameters where the optimum is searched for all scenarios at the same time. Such optimum is more robust than one achieved through a process using separate scenarios since the intended use of the robot is represented by the multi-scenarios.

2 Theoretical background

Basically, two types of robust optimization approaches exist:

1. more, theoretically driven one - these tend to produce models that can hardly be used for real-world problems, and
2. application driven ones - these are often so narrowly tailored, that they can barely applied to other problems.

While solving many problem our aim is obtain solutions that in terms of objectives and feasibility are as good as possible and at the same time are at least sensitive to the parameter variations. Such solutions are said to be robust optimum solutions. In this process we investigate the trade-off between the performance and robustness of optimum solutions. The fitness value is a measure of performance of design solutions with respect to multiple objectives and feasibility of the original optimization problem. The robustness index represents a parameter sensitivity estimation approach, it is a measure that quantitatively evaluates the robustness of design solutions.

We can approach robustness in two ways and while defining the optimum value there are also two ways we can approach this as well. The first way is to search for an analytic solution, the gradient is an effective indicator of the design robustness, two problems need to be resolved before the robust design optimization can be carried out using the gradient based approach. The gradient is an n-dimensional vector, and because the units of noise factors are different, the mathematical operation of this vector can be a challenge. The gradient is a vector rather than a scalar, and thus, it is not convenient to be used as a screening index in a robust design optimization.

The second approach in robustness and while defining the optimum value is to apply heuristic optimization methods.

These methods, don't use the gradient or Hessian matrix of the objective function, they have several parameters that are fitted to the problem.

2.1 Basic principles of uncertainty and robustness

We now introduce the general framework of uncertain problems form the starting point for robust optimization. Almost every optimization problem comes from some degree of uncertainty, even if it is not visible at first. Two types of uncertainty can be distinguished: microscopic and macroscopic uncertainty.

Microscopic uncertainty includes:

- In case of numerical errors, storing any number on a computer system is only possible up to a certain exactness, resulting in so-called floating-point errors that may propagate.

- In case of the measurement errors: the mathematical models are applied to real world problems, they need to be supplied with data that was measured in some way, these measurements may be intrinsically inexact.
The macroscopic uncertainty includes:
-The *forecast errors*, knowledge about the future is seldom exact.
- *Changing environments*, due to long-term setting, the environment naturally changes over the course of time.
The parameters of the optimization models are uncertain for various reasons. Measurement errors can occur if parameter values are determined by physical experiments.
The other reason is that the models often contain parameter values for which the value will only be known in the future.
The uncertainty set is a major impact not only on the type of robustness that we consider, but also on the computational complexity of the robustness models, also, the way the functions $\boldsymbol{f}$ and $\boldsymbol{F}$ depend on $\boldsymbol{\xi}$ (from scenario parameter). In order to accommodate such uncertainties, instead of $P(\boldsymbol{\xi})$, the following parameterized family of problems is considered, and the general robust optimization RO formulation is:

$$\begin{aligned} &\textit{minimum of } f(x, \boldsymbol{\xi}) \\ &\textit{subject to } F(x, \boldsymbol{\xi}) \leq 0 \end{aligned} \tag{1}$$

In practice often not known exactly which values such a scenario ξ may take for an optimization problem $P(\xi)$. We assume that it is known that ξ lies within a given uncertainty set $\mathcal{U} \subset R^M$, M that represents the scenarios we assume to be likely enough to be considered in our analysis, then we denotes as:

$$P(\xi),\ \xi \in U \tag{2}$$

The choice of the uncertainty set is a major impact not only on the type of robustness that we consider. Also, the computational complexity of the robustness models is interesting, and should be made carefully by the modeler. The functions f and F depend on ξ leaves some freedom to the modeler's decision - in the simplest case, ξ coincides with the uncertain parameters of the given optimization problem.
The optimization problems are problems that cannot be solved to optimality, or to any guaranteed bound, by any exact (deterministic) method within a ‘‘reasonable’’ time limit.
The aim of optimization and heuristic solutions is the same – to provide the best possible solution to a given supply chain problem – but their outcomes are often dramatically different.
Here we examine the differences between optimization and heuristics, and explore the pros and cons of each approach.
The main advantage of the optimization approach is that it produces the best possible solution to a given planning and scheduling problem.
Indeed, optimization algorithms are guaranteed to generate optimal solutions, which outperform their heuristic counterparts and enable businesses to maximize cost- and operational-efficiency.
One of the chief benefits of optimization models is their flexibility, as they can automatically adjust and adapt to take into account the myriad decision variables and changing goals, constraints, and complexities in any business environment and generate the best possible planning and scheduling solutions.
The optimization models are highly sophisticated, and specific expertise and technologies are required to devise and deploy optimization solutions. For example, in order to generate an optimization solution, a thorough understanding of mathematical programming concepts and utilization of special solvers are necessary. Some real-world processes cannot be adequately modelled using linear optimization techniques, and it is difficult to model objectives such as “fairness” in an optimization model.

2.2 Linear and nonlinear optimization problem

While RO can be applied to many optimization problems, let us demonstrate its use on a linear optimization problem. The “general” formulation of an uncertain linear optimization problem is as follows:

$$\begin{aligned} &\textit{minimum of } f(x, \boldsymbol{\xi}) = c(\xi)^T x \\ &\textit{subject to } F(x, \boldsymbol{\xi}) = A(\xi)x - b(\xi) \leq 0 \end{aligned} \tag{3}$$

However, it is also possible that the unknown parameters $(A;b;c)$ may depend on (other) uncertain parameters $\xi = R^M$ where M needs not be the number of uncertain parameters of the given problem. The robust linear optimization problems relative easily are tractable.
In case of robust quadratic optimization, the quadratically constrained quadratic programs have defining functions $f(x,\xi)$ in next form:

$$f(x, \xi) = \|\boldsymbol{A}(\boldsymbol{\xi}), \boldsymbol{x}\|^2 + \boldsymbol{b}^T(\boldsymbol{\xi})\boldsymbol{x} + \boldsymbol{c} \tag{4}$$

Reaching a higher level of robustness will greatly increase the complexity of the model description. There are few partial solutions in the Matlab package. The computational demand is much more complex for complex systems.
We will present some of the current approaches to robust optimization that are given. Due to the high

conservatism of strict robustness, further research in robust optimization focused to a high degree on ways to relax this concept. We now shortly describe some of these approaches.

Thus far this paper has addressed optimization in the static or one-shot case: the decision-maker considers a single-stage optimization problem affected by uncertainty. In dynamic (or sequential) decision-making problems are single-shot assumption is restrictive and conservative ways.

Sequential decision-making appears in a broad range of applications in many areas of engineering and beyond. There has been extensive work in optimal and robust control and approximate and exact dynamic programming. We consider modelling approaches to incorporate sequential decision-making into the RO framework. One interesting representant of sequential decision-making is robust adaptable optimization.

In an adjustable robust optimization model, first we must make here-and-now decisions $x \in R^{n_x}$ before we observe the realization of an uncertain parameter within a convex uncertainty set $\mathcal{U}$. After that, we can choose our wait-and-see decision $y \in R^{n_y}$ to satisfy all the constraints. Similar to one dimensional robust model:

$$\text{minimum of } f(x, \boldsymbol{\xi})$$
$$\text{subject to } A(\xi) + By \geq D\xi + d;\ \forall \xi \in U,\ \exists y \qquad (5)$$

For ease of exposition, we did not include any wait-and see decisions in the objective. Nevertheless, model (5) can also be used to describe models with uncertainty or wait-and-see decisions in the objective by replacing the objective by an auxiliary variable $F \in R$ and adding the constraint:

$$c^T + b^T \leq F \qquad (6)$$

Substituting linear decision rule in (5) we get:

$$\text{minimum of } f(x, \boldsymbol{\xi})$$
$$\text{subject to } A\boldsymbol{\xi} + B(\bar{y} + Y\xi) \geq D\xi + d;$$
$$\forall \xi \in U,\ \exists y \qquad (7)$$

which is a standard robust optimization model without wait-and-see decisions.

2.3 The heuristic method for optimisation

The main advantage of adopting a heuristic approach is that it offers a quick solution, which is easy to understand and implement. Heuristic algorithms are practical, serving as fast and feasible short-term solutions to planning and scheduling problems, but are not capable of serving as viable solutions that deliver the best possible results.

A metaheuristic is an algorithm designed to solve approximately a wide range of hard optimization problems without having to deeply adapt to each problem. Indeed, the greek prefix ''meta'', which is present in the name, is used to indicate that these algorithms are ''higher level'' heuristics, in contrast with problem specific heuristics. Metaheuristics are generally applied to problems for which there are no satisfactory problem-specific algorithms to solve them. The metaheuristics has the following characteristics: they are nature/inspired, use of stochastic components. Numerous multi-variable evolutionary optimization methods exists and it is generally difficult to choose the best because the performance of each method is problem-dependent, the heuristic and hybrid methods are promising for a non-linear, multivariable problems.

First, we present single-solution based metaheuristics, also called trajectory methods. Unlike population-based metaheuristics, they start with a single initial solution and move away from it, describing a trajectory in the search space. Some of them can be seen as ''intelligent'' extensions of local search algorithms. Trajectory methods mainly encompass the simulated annealing method, the tabu search, the GRASP method, the variable neighborhood search, the guided local search, the iterated local search, and their variants.

Secondly, population based metaheuristics deal with a set of solutions rather than with a single solution. The most studied population-based methods are related to Evolutionary Computation (EC) and Swarm Intelligence (SI). EC algorithms are inspired by Darwin's evolutionary theory, where population of individuals is modified through recombination and mutation operators. In SI, the idea is to produce computational intelligence by exploiting simple analogies of social interaction, rather than purely individual cognitive abilities.

We continue with presentation of two dominant heuristic methods in the sequel.

Genetic Algorithm (GA) can be applied to solve problems that are not well suited for standard optimization algorithms, including problems in which the objective function is discontinuous, non-differentiable, stochastic, or highly nonlinear.

Being a member of the family of evolutionary computation, the first step of GA is population initialization which is usually done stochastically. The GA usually uses three simple operators called selection, recombination (usually called crossover) and mutation.

Selection is the step of a genetic algorithm in which a certain number of individuals are chosen from the current population for later breeding (recombination or crossover), the choosing rate is normally proportional to individual's fitness value. There are several general selection techniques. Tournament selection and fitness proportionality selection (also known as roulette-wheel selection) considering all given individuals. Other methods only choose those individuals with a fitness value greater than a given arbitrary constant. Crossover and mutation taken together is called reproduction. They are analogous to biological crossover and mutation respectively. The most important operator in GA is crossover, which refers to the recombination of genetic information during sexual reproduction. The child has a lot of common characteristics with one's parents. Therefore, in GAs, the offspring has an equal chance of receiving any given gene from either one parent because the parents' chromosomes are combined randomly.

Particle Swarm Optimization (PSO) is one of the most important swarm intelligence paradigms. The PSO uses a simple mechanism that mimics swarm behaviour in bird flocking and fish schooling to guide the particles to search for globally optimal solutions. The PSO is a stochastic global optimization method which is based on simulation of social behavior, exploits a population of potential solutions to probe the search space. In contrast to the GA and ES, in PSO no operators inspired by natural evolution are applied to extract a new generation of candidate solutions. The PSO relies on the exchange of information between individuals, called particles, of the population, called swarm. As a result, each particle adjusts its trajectory towards its own previous best position. In next step use the best previous position attained by any member of its neighborhood, in the global variant of PSO, the whole swarm is considered as the neighborhood.

The particles are manipulated according to the following equations:

$$V_{id}^{n+1} = V_{id}^{n} + C_1\varphi_1(P_{id}^{n} - X_{id}^{n}) + C_2\varphi_2\left(P_{gd}^{n} - X_{id}^{n}\right) \tag{8a}$$

where *n = 1, 2,...,N*, and *N* is the size of the swarm; $\varphi_1 and\, \varphi_2$ are two random numbers uniformly distributed in the range *[0, 1]*, C_1 and C_2 are constant multiplier terms known as acceleration coefficients, and the position of each particle is also updated in each iteration by adding the velocity vector to the position vector:

$$X_{id}^{n+1} = X_{id}^{n} + V_{id}^{n+1} \tag{8b}$$

The GA and the PSO must also have a fitness evaluation function that takes the agent's position and assigns to it a fitness value. The position with the highest fitness value in the entire run is called the global best. Each agent also keeps track of its highest fitness value. The location of this value is called its personal best. Each agent is initialized with a random position and random velocity. The fitness function is at the heart of an evolutionary computing application. It is responsible for determining which solutions (controllers in the case of ER) within a population are better at solving the particular problem at hand.

2.4 Robust optimal solution

Instead of one optimal solution, a multi-criteria optimization gives rise to a set of optimal solutions. In this solution set, not one solution can be considered to be better than any other with respect to all objective functions. These optimal solutions are known as Pareto Optima also known as non-inferior solutions or non-dominated solutions.

The RO problems may have multiple optimal solutions, and that not all of these solutions are Pareto robustly optimal, this solution is called Pareto robustly optimal. The following two-steps can find as so-called Pareto robustly optimal solutions:

- Solve the original model, which gives a solution with minimal worst-case costs.
- Change the objective into minimizing the costs for the nominal demand trajectory, and add a constraint that ensures that the worst-case costs do not exceed the costs found in the previous step.

A common practice in solving multi-criteria optimization problems is to convert the multiple objectives into one objective function and thus a substitute scalar optimization problem is constructed, which can be handled using standard optimization routines (exist a number of methods).

Each problem has its own fitness function. The fitness function that should be used depends on the given problem. When it comes to formulating a problem using genetic or PSO algorithms then coming up with a fitness function for the given problem is the hardest part. There is no hard and fast rule that a particular function should be used in a particular problem. However, certain functions have been adopted by data scientists regarding certain types of problems. Typically, for classification tasks where supervised learning is used, error measures such as Euclidean and Manhattan distance have been widely used as the fitness function.

Similarly, when using gradient-based multi-objective optimization methods, gradients are needed to be calculated for the optimization problem solver, existing gradient information can be used to control and direct the optimization process.

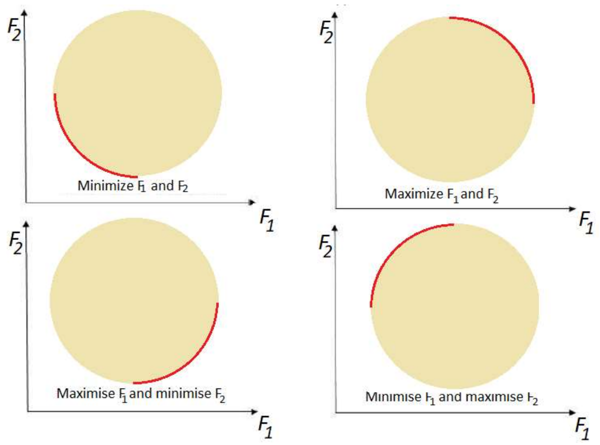

***Fig. 1.** Alternatives for Pareto robust solutions*

The Pareto set is for four different scenarios for the minimisation and/or maximisation of two objectives. The solid curves represent the optimal Pareto sets. In all cases the Pareto set consists of a particular edge of the feasible search region. As can be seen in this figure, the Pareto front could be convex, non-convex or discontinuous. Another characteristic of multi-objective problems are on finding global and local Pareto sets.

3 Real Problems for Robust Analysis

For a longer time, we have been researching and developing the following fields:

- Hexapod robot walker (theoretical mechanics) [1] and [2].
- Tomography (theoretical electromagnetics) [4] and [5].

In both cases, we are solving inverse tasks in a particularly nonlinear environment with noise. In these cases, only robust solution is the good solution.

The hexapod robot contains not only noise and nonlinearity but also disruption and integers, which means that only heuristic methods can be applied while searching for robust optimum solutions.

When it comes to solving tomographic inverse problem, then initial solution is expected to be differentiable so we can use gradient-based multi-objective optimization methods. In the following results emphasis will be on the refinement where different gradient based methods can be used. Based on our publications and research we can say that in order to get the best results we should use heuristic based methods to find the solutions for tomographic tasks.

3.1 Tomography, robust solution

In the field of theoretical electromagnetics, we are working on tomography researches, in essence in the area of medical-biological electric impedance tomography (we have a patent in this topic – [5]) and in . We are also doing magnetic impedance tomography researches which specifically means testing the internal structure of „high-thickness" ferromagnetic materials (we have a patent – [4]). For example, in the case of depth analysis of nuclear reactor walls or for any kind of other, industrial-like uses where other types of nondestructive analyzes cannot be applied. In both of these cases of tomography research, we used our own high sensitivity electronics/sensor systems and self-developed calculating processes.

There is another area where we are solving complex inverse tasks. By using optimization processes, we can define those robot structural parameters from the measurement results, which we are unfamiliar with (not published yet or cannot be measured). By using the estimated structural parameters we can overcome those nonstationary events' effects, which cannot be predicted in the development of regulatory drive mechanisms [7]. This way we can assure the hexapod walking robot's robust controlling on a wider scenario platform. In this case, the modeling, included the robot's inverse kinematics and inverse dynamics. The object of the research was the physical hexapod walking robot, which is our own development, it involves more generations of robots. The [1] publication discusses the validation process of the robot in detail.

There is a high degree of uncertainty in both research topics, but the reasons are different. In case of tomography researches, while doing measurement processes, we can encounter systematic errors, which most of the time are irreducible. The noise, that occurs while measuring, is the biggest problem in further processing. While solving the electromagnetic inverse task, not only the measurement interference signals cause issues but the "mathematical" noise (quantization error) also adds up to the uncertainty. A dimensional "depth" information content should be created in case of tomography, this is the main problem. We can only make measurements on the surface of the body, we cannot do inner measurements, because this is a nondestructive process. In the field of theoretical electromagnetics, direct task means that we know the examined body's material structure. While using Maxwell equation and exciting the body, we can calculate the examined body's surface's voltage conditions. This is also a complex task (nowadays, there are finite element FEM packs, for example, COMSOL). Compared to this, inverse task solving is a lot more complex. This is because there are only finite measuring results on the surface

and this is how we need to determine the material's structure in depth.
This non-local property of conductivity imaging, which still applies at the moderate frequencies used in EIT, is one of the principal reasons that EIT is difficult. It means that to find the conductivity image one must solve a system of simultaneous equations relating every voxel to every measurement. Non-locality in itself is not such a big problem provided we attempt to recover a modest number of unknown conductivity parameters from a modest number of measurements. Worse than that is the ill-posed nature of the problem. Small errors in measurement can violate consistency conditions, such as reciprocity [4]. The starting point for consideration of EIT should be Maxwell's equations. But for simplicity let us assume direct current or sufficiently low a frequency current that the magnetic field can be neglected.
The time harmonic Maxwell's equations, with combining the conductivity and permittivity as a complex admittivity $\sigma + i\omega\varepsilon$, the solution is in next form:

$$E = -\nabla\phi \tag{9}$$

In the mathematical literatures we will often see the assumption that ϕ lies in the Sobolev Space $H^1(\Omega)$, in our application these spaces are easily understood on an intuitive level and have a natural physical meaning. While doing the research, we were heading to two directions (patent code – [4]). In terms of research, we were defining the material magnetic properties while using magnetic space characteristics measurements. In the case of (patent code – [5]) research, we were defining electric characteristics on the surface with electric voltage/current values. Next, are going to discuss a so-called electric impedance tomography EIT problems.

There are going to be similar problems in the field of magnetic material characteristics. The inverse problem, as formulated by Calderon, is to recover σ from Λ_σ, for one set of Dirichlet and Neumann boundary data, provided it contains enough frequency components, is enough to determine the boundary between two homogeneous materials with differing admittivity.
Electrical imaging system uses a system of conducting electrodes attached to the surface of the body under investigation. One can apply current or voltage to these electrodes and measure voltage or current respectively, measurement principle presented on Fig. 2.

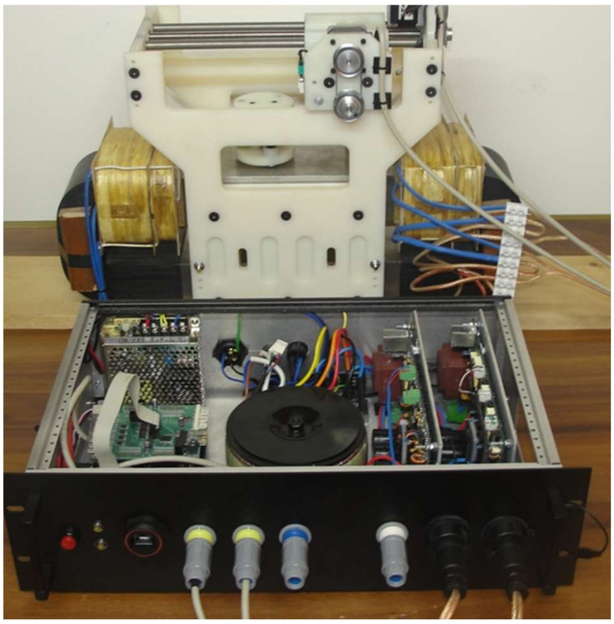

Fig. 2 *Realized tomograph*

The total variation functional is assuming an important role in the regularization of inverse problems in the image restoration context. The use of such a functional as a regularization penalty term allows the reconstruction of discontinuous profiles.
Newton-like algorithms is one of usual methods to solve inverse problem of (EIT), but it is sensitive to initial values. The PSO, genetic algorithm, Markov Chain Monte Carlo (MCMC) and many other methods for solving inverse problem are presented in literature. These methods are less sensitive for initial value but there are other problems that need to be resolved. The stable inversion method requires regularization. The efficient regularization is Tikhonov regularization. With applied the Tikhonov regulation a nonlinear total variation functional regularized inversion in a short time.

3.2. Hexapod robot, robust realization

In case of Hexapod walking robot, direct task means that we have to completely describe the simulations which includes every nonlinear, nonstationary etc. phenomenon. With the help of this simulation, every property can be calculated. When we have 18 joints (with 18 drives) it is also not an easy task. The "inverse" task means the following: we can measure a finite number of points, here we need to mention that we are not able to measure every single, important parameter (these properties can be about structural, like mechanical properties). We need to emphasize that this is a real structure and the system is especially nonlinear and nonstationary.

The basis of searching for the robust solution is that the optimal solutions need to be found. Not every optimal solution is a robust solution, it can even be different. On the multi-dimensional Pareto surface, the solutions exclude each other by their behavior on other surfaces. This means that we should be prudent when choosing the optimal solution because it needs to be robust as well.

Within our article we are going to represent robustness on a very simple example in case of the Hexapod walking robot: on the simplified robot leg system. Fuzzy is usually independent of any kind of use which requires controlling technique task solutions. There can be another case, when next to the not properly chosen [3] parameters, due to its basic philosophy, we get the robust solution as a result, not like other methods (in this article we do not get into proving this fact). This is why in the following chapters we only deal with the devices which are mentioned in the chapter.

One of our basic expectation was to have a reliably working system while doing the system's research and realizing process, in hardware and software environment as well. We wanted it to function under any scenario. In theory, the number of scenarios is infinite if the system is for free use. In case of an endless number of scenarios, the test cannot be done. This means we need to create a set of simple scenarios that mostly cover opportunities arising in real situations on their own. Or if they have any other realizable combination, then they are combined. The main goal is to have robustness in the system's wider area, let it be foreseeable or unforeseeable environmental effects. However, it is difficult to realize this in most cases.

The precise modelling and controlling of driving DC motors are essential parts for the optimization procedure of the Szabad(ka)-II hexapod robot. The simulation model of Szabad(ka)-II hexapod was already built and validated [1]. It includes the detailed model of Faulhaber coreless DC micromotor and gear as engine of robot arms [1]. Model is implemented in Simulink environment, which forms the connection between the electrical and mechanical sides. The electrical side contains the model of embedded motor controller (PID or fuzzy controller), the model of sensors (encoder and current sensor, accelerometer, gyroscope) and the model of PWM amplifier. The mechanical side contains the all mechanical parts of the motor and gear system, its efficiency and the three dimensional kinematics and dynamics of the 18 DOF hexapod robot.

The PI and fuzzy-PI controllers have been already developed and optimized for this robot, where the objective function was the quality of robot walking. In this paper the objective function is independent from the robot walking, therefore it can be used for different purposes. The quality evaluation by the fitness or objective function focuses to reduce the high peaks and jerks in the motor current and torque and the robustness against the motor and load parameters.

We defined 17 rule in the core of fuzzy inference system, which are intended to decrease the speed error *dn(t)[rpm]* while hold down the current of motor *I(t)[A]* and especially the changing of current *dI(t)[A/s]*. We have tested the motor with constant speed rotations as desired input (*7000[rpm]*). In this test the gear is ignored yet. In the comparison three different control scenarios have been done in order to check the difference between them:

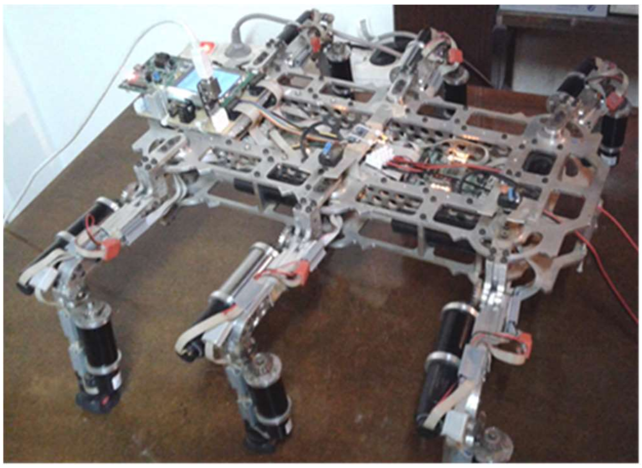

Fig. 3. *Hexapod robot Szabad(ka)-II.*

The quality or fitness evaluation of a controlled process is the key to be able to quantify the performance and optimize the system parameters. Genetic Algorithm (GA) was used for the optimization, because it has been already researched, and the convenient developed programs were available for this purpose. For the equal opportunity competition the parameters were optimized both for PID and Fuzzy controller separately, while the input conditions, fitness evaluation and the optimization algorithm with its own parameters were the same.

The details of the controller optimization, the selected fuzzy parameters to be optimized, and the setting of the optimization algorithm are not detailed here (in detail, see [1]).

4 Conclusion

In this article, we represent two problems at inverse task model creation. In case of these systems, there are seemingly no common characteristics. But there is one common problem, which is the need to find one robust solution. There are two ways for finding the solution for robustness. The first option is when we know the whole description (in form of differential equation) of the examined system. In this case, the linear solution can give a good result. It is used a lot of times (in both problems) with bigger and smaller or success as well. We present not only the use of linear solution mechanism but the robot with examples, how does the heuristic solving mechanism work.

Another issue arises when we cannot describe the whole system, they are nonlinear (we do not know the nonlinear measure and type). In a lot of cases, there are systematic errors, that cannot be eliminated. The problem solvers are disturbed by high noise level, the noise itself and the correlation of system's parameters with noise. Our experience shows that the linear solution works until a certain level, but by combining the heuristic solution, even more at the same time, it can help a lot in cases of making the result even better.

References:

[1] I. Kecskes, E. Burkus, F. Bazso, P. Odry, Model validation of a hexapod walker robot, *Robotica* Vol. 35 No. 2, 2015, pp. 419-462. doi:10.1017/600 S0263574715000673.

[2] I. Kecskes, P. Odry, Optimization of PI and fuzzy-PI controllers on simulation model of Szabad(ka)-II walking robot, Int. J. Adv. Robot. Syst. 11 (2014) 186. doi:10.5772/59102.

[3] Laslo Tadic, Miodrag Popovic, Peter Odry: "Fuzzified Gabor Filter for License Plate Detection" *International Scientific Journal Engineering Applications of Artificial Intelligence*, 2016, DOI: 10.1016/j.engappai.2015.09.009

[4] Magnetic tomography of ferromagnetic materials dept structure, PN°: 1600165

[5] Data acquisition and processing method to assembly soft tomography examinations: PN°: 1500616

[6] István Kecskés, Ervin Burkus, Zoltán Király, Ákos Odry, Péter Odry: Comparatition of motor controllers using a simplified robot leg: PID vs fuzzy logic; MCSI, august 2017; Corfu

[7] Cariou C., Berducat M., Gobor Z. and Seiferth B.: (2017), Mobile Robot Trajectory Planning Under Kinematic and Dynamic Constraints for Partial and Full Field Coverage. *J. Field Robotics.* doi:10.1002/rob.21707

[8] Ákos Odry; Róbert Fullér; Imre J Rudas; Péter Odry: Kalman filter for mobile-robot attitude estimation: Novel optimized and adaptive solutions, *Mechanical System and Signal Processing* (ISSN: 0888-3270) 110: pp. 569-589. (2018); DOI: 10.1016/j.ymssp.2018.03.053;