
Proceedings Paper:
Salomon, S., Avigad, G., Fleming, P.J. orcid.org/0000-0001-9837-8404 et al. (1 more author) (2013) Optimization of Adaptation - A Multi-objective Approach for Optimizing Changes to Design Parameters. In: Evolutionary Multi-Criterion Optimization. 7th International Conference, EMO 2013, March 19-22, 2013, Sheffield, UK. Lecture Notes in Computer Science, 7811 . Springer Berlin Heidelberg , pp. 21-35. ISBN 978-3-642-37139-4

https://doi.org/10.1007/978-3-642-37140-0_6

Optimization of Adaptation - a Multi-Objective Approach for Optimizing Changes to Design Parameters

Shaul Salomon[1], Gideon Avigad[2], Peter J. Fleming[1], and Robin C. Purshouse[1]

[1] Department of Automatic Control and Systems Engineering
University of Sheffield
Mappin Street, Sheffield S1 3JD, UK
{s.salomon,p.fleming,r.purshouse}@sheffield.ac.uk
[2] Department of Mechanical Engineering
ORT Braude College of Engineering, Karmiel, Israel
gideona@braude.ac.il

Abstract. Dynamic optimization problems require constant tracking of the optimum. A solution for such a problem has to be adjustable in order to remain optimal as the optimum changes. The manner of changing design parameters to predefined values is dealt with in the field of control. Common control approaches do not consider the optimality of the design, in terms of the objective function, while adjusting to the new solution. This study highlights the issue of the optimality of adaptation, and defines a new optimization problem – "Optimization of Adaptation". It is a multiobjective problem that considers the cost of the adaptation and the optimality while the adaptation takes place. An evolutionary algorithm is proposed in order to solve this problem, and it is demonstrated, first, with an academic example, and then with a real life application of a robotic arm control.

Keywords: dynamic optimization, adaptation, optimal control

1 Introduction

The competency of living creatures to adapt to a changing environment is a crucial virtue. Adaptation is the evolutionary process that allows species to survive. The properties of a species gradually changes to meet the demands of the changing environment. It is a slow mechanism, and it may take hundreds of generations for a trait to establish among the population [1]. Adaptation is also related to the ability of a specimen to change some of its physical properties during its lifetime. These changes are not genetic changes, and they are not passed on to any offspring. Nevertheless, the ability to adapt is inherent within the species' genotype. Examples of fast adaptation are the changing colours of a chameleon [2], the expansion and contraction of the pupil as light changes [3], and the increasing number of red blood cells as a reaction to low percentage of oxygen in the air in high altitude [4].

The necessity for adaptation may be also related to engineering (e.g., a product should be adapted to changes in market demands), companies (e.g., change of personnel or facilities, as costs change), politics, and many other fields of interest. In engineering design, adaptation may be ensured by including tunable parameters that can be altered when changes are required.

In the context of *single objective optimization*, an initial design might lose its optimality as time passes due to changes that influence the objective function, or it might become infeasible with the changing of constraints. A tunable design may be able to adapt in order to maintain satisfactory performance, or to remain feasible, when such changes occur. This virtue avoids the need for producing a totally new design whenever an environmental change occurs, and it enables prolonging the lifetime of a product.

These kinds of optimization problems, where the optimal solution changes with time, are known as *dynamic optimization problems* (DOPs). Mathematically, a *dynamic single objective optimization problem* is formulated as follows:

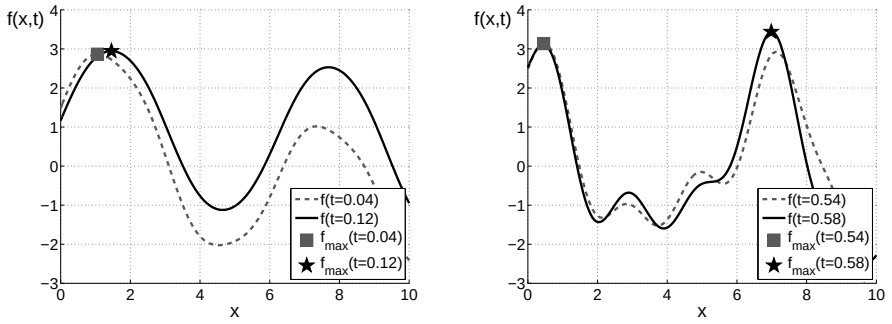
Definition 1. *Dynamic single objective optimization problem:*

$$\begin{aligned} & \max_{\mathbf{x} \in Q} f(\mathbf{x}, t) \\ s.t. \quad & g_i(\mathbf{x}, t) \geq 0 \ , && (i = 1, \ldots, I) \\ & h_j(\mathbf{x}, t) = 0 \ , && (j = 1, \ldots, J) \end{aligned}$$

where $Q \subset \mathbb{R}^M$ is the search domain, f is the objective function, and g and h are the inequality and equality constraints, respectively. Since the objective function and the constraints are time dependent, the optimal solution $\tilde{\mathbf{x}}$ also changes with time: $\tilde{\mathbf{x}} \Rightarrow \tilde{\mathbf{x}}(t)$.

Many methods, including evolutionary algorithms, have been developed in order to solve DOPs, and to determine what changes should be performed in order to achieve optimal or satisfactory performance over time [5]. In many cases, the required changes in design over time are continuous and relatively small. In other cases, the optimum might "jump" to a different region of the design space. This can happen when a totally different solution becomes the optimum, or when the current optimal solution becomes infeasible. The former case is illustrated in Figure 1. The dynamic function $f(x,t) = 2sin(x+2t) + tsin(3x-t) + cos(3xt)$ is presented at several time instants. In Figure 1(a) the optimal solution changes slightly between $t = 0.04$ and $t = 0.12$. On the other hand, in Figure 1(b), a new optimal solution is developed in the region of $x = 7$, and a "jump" of the optimum occurs between the time instants $t = 0.54$ and $t = 0.58$. Since the required changes in design at times of an optimum "jump" are significant, some issues regarding the manner of performing these changes should be considered.

When the required change is a mechanical change (as opposed to tuning an electric signal, for example, which is rapid), the adaptation time might be non-negligible. During that time, the function value of the system changes according to the change in design. Considering the example of Figure 1(b), the function values between the old optimum and the new one are much lower than the optimal value. If the change is continuous (i.e., x goes through all the values in between



(a) The optimum changes slightly within the same region.

(b) The optimum "jumps" to a different region.

Fig. 1. A dynamic function at four time instants. The optimum changes from the grey square to the black star.

the old and the new solutions), the function values during the adaptation are not optimal.

Although existing research has addressed finding efficient ways to track the moving optimum, no research known to the authors has addressed the optimality during the change itself. This study comes to shed some light on this issue. The aim here is not to suggest a new method for solving DOPs, but to optimize the adaptation of a solution when it requires significant changes. It is assumed here that the DOP is already solved, and the new location of the optimum is known. The aim is at designing a trajectory in phase plane (high level control design), according to its related performances in objective space. This is in contrast to the common approach in control theory, where the optimization is based on the trajectories in phase plane (e.g., minimizing the states' deviation from the new set-point).

Figure 2 depicts two possible trajectories to trace the moving optimum of a single objective DOP with two design variables. Different trajectories in design space are possible in the example, since it consists of two variables. The values of the objective function are illustrated by contour lines. The location of the old optimum $\mathbf{x}_0$ is marked with a square, and the location of the new optimum $\mathbf{x}_f$ is marked with a star. In this example, Trajectory 1 (marked with triangles) is along a straight line and passes through a region with very low function values, while Trajectory 2 (marked with diamonds) bypasses this region, and passes through regions with high function values. The trajectories of the objective function over time, for the same example, are shown in Figure 3. Note that the function values along Trajectory 1 are lower than those along Trajectory 2. Therefore, assuming the function is static during the traverse, it can be stated that the optimality of Trajectory 2 is better than the optimality of Trajectory 1 in terms of the function values.

In many cases, changing a system's design has a cost, in terms of money, energy or other resources. The way parameters are changed, especially when more than one parameter is to be adjusted, may affect the cost of the change. The possible conflict between the optimality of the performance during the change, and the cost of the adaptation defines a new optimization problem which is termed here as- *Optimization of Adaptation*. It is a *multi-objective optimization problem* (MOP) by its nature, since the function value during the adaptation process, and the cost of the change are to be optimized at the same time.

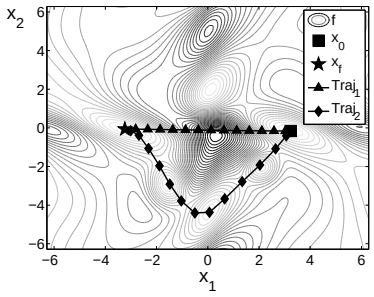


Fig. 2. Two possible trajectories to adapt from $\mathbf{x_0}$ to $\mathbf{x_f}$. Brighter contour lines represent higher function values.

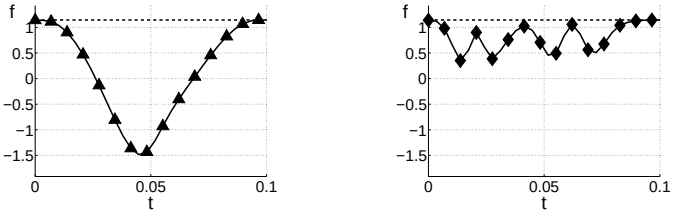


Fig. 3. The trajectories $f(t)$ for the two trajectories of Figure 2. Figure 3(a) refers to trajectory 1, and Figure 3(b) refers to trajectory 2.

The remainder of this paper is organized as follows. In Section 2 the *Optimization of Adaptation Problem* is defined, and a method to assess the objective functions is suggested. In Section 3 an evolutionary algorithm is suggested to solve the problem. In section 4, an academic example and a real-world application for the proposed optimization method are presented. Finally, the paper ends with a discussion in Section 5.

2 Methodology

2.1 Problem Definition

In order to optimize the adaptation of a system at a time instant t_{jump}, when a significant change in design is required, the following assumptions are made:

1. The change takes place over a time interval $[t_0, t_f]$ which is significantly shorter than the time constant of the DOP. Hence, the function value and the constraints can be considered as static for the duration of the change:
   $f(\mathbf{x}, t_{jump}) = f(\mathbf{x})$
   $g_i(\mathbf{x}, t_{jump}) = g_i(\mathbf{x})$
   $h_j(\mathbf{x}, t_{jump}) = h_j(\mathbf{x})$
2. The state of the design parameters prior to the change is $\mathbf{x}(t_0) = \mathbf{x_0}$.
3. The state of the design parameters at the end of the change is the new optimal solution of the DOP: $\mathbf{x}(t_f) = \mathbf{x_f} \quad , \quad f(\mathbf{x_f}) = \max f(\mathbf{x})$ (assuming maximization).
4. The system is at a steady state at the beginning and at the end of the adaptation.

Considering the above assumptions, the new optimization problem is defined as follows:

Definition 2. *Optimal adaptation problem (OAP):*

$$\min_{\mathbf{x}(t) \in Q} \{Error(\mathbf{x}(t)), Cost(\mathbf{x}(t))\} \ , \qquad t \in [t_0, t_f] \tag{1}$$

$$s.t. \quad \mathbf{x}(t_0) = \mathbf{x_0} \quad , \quad \mathbf{x}(t_f) = \mathbf{x_f} \tag{2}$$

$$\left.\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t}\right|_{t_0} = \left.\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t}\right|_{t_f} = 0 \tag{3}$$

$$g_i(\mathbf{x}(t)) \geq 0 \ , \qquad (i = 1, \ldots, I) \tag{4}$$

$$h_j(\mathbf{x}(t)) = 0 \ , \qquad (j = 1, \ldots, J) \tag{5}$$

where $\mathbf{x}(t)$ is a trajectory in the design space for the entire time interval, $Error(\mathbf{x}(t))$ is the difference between the function value at time t and the optimal function value $f(\mathbf{x_f})$: $Error(\mathbf{x}(t)) = f(\mathbf{x_f}) - f(\mathbf{x}(t))$, and $Cost(\mathbf{x}(t))$ represents the invested resources for changing $\mathbf{x}$ from $\mathbf{x_0}$ to $\mathbf{x_f}$. The constraints of the original DOP (Eq. (4) and (5)) must be met at all times.

Note that in spite of the fact that the original DOP is a single objective problem, the new proposed OAP is inherently a MOP. In order to maintain high function values along the trajectory, one might need to invest more energy to the adaptation process. For that reason, the two objectives of the OAP may be conflicting, and the solution of the OAP might be a set of optimal trajectories known as the *Pareto optimal set* (POS).

2.2 Assessment of the Objective Functions

Each solution for the proposed problem is a trajectory, and its resulting performances in the objective space $(Error, Cost)$, as defined previously, are trajectories as well. Some research exists on the optimization of trajectories (e.g. [6]), but the common approach is to represent each trajectory by an auxiliary function (e.g. [7],[8]). In this paper we adopt this approach, and assess the objective functions by using a common concept from optimal control theory:

- The error function $Error\left(\mathbf{x}(t)\right)$ is represented by its *integral absolute* (IAE), which is a well known measure of optimality in control theory:
  $IAE = \int_{t_0}^{t_f} \left| f(\mathbf{x_f}) - f\left(\mathbf{x}(t)\right) \right| \, \mathrm{d}t$
  Recalling the trajectories in Figure 3, the IAE of each trajectory is equal to the area in between the function value and the optimal value marked with the dashed line. It is clear that the IAE of the trajectory in Figure 3(b) is smaller than the IAE of the trajectory in Figure 3(a).
- The cost function $Cost\left(\mathbf{x}(t)\right)$ is represented by the overall cost of the control forces required to follow the trajectory $\mathbf{x}(t)$. It is assumed here that the design variables are controlled by a vector of control forces $\mathbf{u} = [u_1, \ldots, u_p]^T$, where each design variable is controlled by one or more control variables. Every control force has its cost, and the total cost C is calculated as follows:
  $C = \int_{t_0}^{t_f} \left| \mathbf{w}^T \mathbf{u}(t) \right| \, \mathrm{d}t$
  where $\mathbf{w}$ is a cost vector with w_i represents the cost of u_i. Each control force variable u_i has its related saturation values, which define the domain of the control forces.

The OAP may now be reformulated by changing the objectives at Eq. 1 in Definition 2, and by considering the saturation values as additional constraints. Because the objective functions are evaluated based on utility functions, the reformulated problem is presented here as:

Definition 3. *Utility based Optimal Adaptation Problem (UOAP):*

$$\min_{\mathbf{x}(t) \in Q} \{IAE, C\} \ , \qquad t \in [t_0, t_f] \tag{6}$$

$$s.t. \quad \mathbf{x}(t_0) = \mathbf{x_0} \quad , \quad \mathbf{x}(t_f) = \mathbf{x_f} \tag{7}$$

$$\left.\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t}\right|_{t_0} = \left.\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t}\right|_{t_f} = 0 \tag{8}$$

$$\mathbf{u}_{min} \leq \mathbf{u}(t) \leq \mathbf{u}_{max} \tag{9}$$

$$g_i\left(\mathbf{x}(t)\right) \geq 0 \ , \qquad (i = 1, \ldots, I) \tag{10}$$

$$h_j\left(\mathbf{x}(t)\right) = 0 \ , \qquad (j = 1, \ldots, J) \tag{11}$$

where $\mathbf{x}(t) = f(\mathbf{u}(t))$.

As is common in control, IAE and C are in conflict with one another. Therefore, the optimal solution for the UOAP, as is the case with the OAP, is expected to be a set rather than a single solution. Note that other reformulations for the OAP are possible.

3 Solving the Problem with an Evolutionary Algorithm

In this section, an evolutionary algorithm (EA) is presented for solving the UOAP, defined in Definition 3.

3.1 The Genotype

For a better visualisation of the problem, the value of the design variables are considered here as their positions. The first and second derivatives in time of the variables are considered as their velocity $\mathbf{v}(t)$ and acceleration $\mathbf{a}(t)$, respectively. Although the solution of the UOAP is the trajectory $\mathbf{x}(t)$, it is coded by $\mathbf{a}(t)$.

The time interval is defined at K time instants- $\mathbf{t} = [t_1, \ldots, t_k, \ldots, t_K]$, where $t_1 = t_0$ and $t_K = t_f$. Each solution is defined by a matrix A of size $M \times K$ of real coded variables. A_{mk} represents the acceleration of the m^{th} design variable at the k^{th} time sample.

Considering the constraints in Eq. (7) and (8), the speed $v_m(t)$ and position $x_m(t)$ can be derived as follows:

$$v_m(t) = \int_{t_0}^{t} a_m(t)\,\mathrm{d}t \tag{12}$$

$$x_m(t) = x_{0m} + \int_{t_0}^{t} v_m(t)\,\mathrm{d}t \tag{13}$$

Of course, the integrals are evaluated by a discrete approximation method such as the trapezoidal rule, for example.

3.2 Constraint Satisfaction: Repair Method

In order to satisfy the constraints regarding t_f in Eq. (7) and (8), two modifications of the acceleration trajectory are made. The first results in an intermediate acceleration trajectory $\mathbf{a}(t)^{**}$ which satisfies Eq. (8), and the second results in the repaired acceleration trajectory $\mathbf{a}(t)$ which satisfies Eq. (7) as well.

Let a pre-repair acceleration trajectory be $\mathbf{a}^*(t)$. It is clear from Eq. (12) that in order to force the final speed $\mathbf{v}_m(t_f) = 0$, the mean acceleration has to be zero. This can be realized by subtracting from $\mathbf{a}^*(t)$ its mean value $\bar{\mathbf{a}}^*$. The resulting intermediate acceleration trajectory is $\mathbf{a}^{**}(t) = \mathbf{a}^*(t) - \bar{\mathbf{a}}^*$. The

intermediate velocity trajectory $\mathbf{v}^{**}(t)$, using Eq. (12), satisfies the constraint of Eq. (8). Then the intermediate final position of the trajectory $\mathbf{x}^{**}(t_f)$ according to Eq. (13) is:

$$\mathbf{x}^{**}(t_f) = \mathbf{x}_0 + \int_{t_0}^{t_f} \mathbf{v}^{**}(t)\,\mathrm{d}t$$

At the second stage, in order to satisfy Eq. (7), $\mathbf{a}^{**}(t)$ is scaled by a scaling vector $\mathbf{l}$:

$$\mathbf{l} = [l_1, \ldots, l_m, \ldots, l_M]^T\,, \qquad l_m = \frac{x_{fm} - x_{0m}}{x_m^{**}(t_f) - x_{0m}}$$

The repaired final acceleration variable that satisfies both constraints is $\mathbf{a}(t) = \mathbf{a}^{**}(t) \cdot \mathbf{l}$.

Although the suggested gene manipulation may result in violation of the constraint in Eq. (9), it speeds up the evolutionary process since it eliminates the two equality constraints in Eq. (7) and (8).

3.3 The Evolutionary Algorithm

All of the problems of Section 4 are solved using NSGA-II with constraint domination [9]. The genetic operators used are the simulated binary crossover (SBX) operator and polynomial mutation [10], with distribution indexes of $\eta_c = 15$ and $\eta_m = 20$ respectively. A cross-over probability of $p_c = 1$ and a mutation probability of $p_m = 1/MK$ are used. The stopping criterion is a maximal number of generations. A schema of the EA is presented in Algorithm 1.

Algorithm 1 The evolutionary algorithm for solving the UOAP

1: $R_1^* \leftarrow$ generate a random set of solutions of size $2N$
2: $R_1 \leftarrow$ modify R_1^* according to the procedure in Section 3.2
3: $g \leftarrow 1$
4: **while** $g \leq$ number of generations **do**
5: evaluate R_g
6: $P_{g+1} \leftarrow$ select N solutions from R_g
7: $Q_{g+1}^* \leftarrow$ evolve from P_{g+1} (cross-over and mutation)
8: $Q_{g+1} \leftarrow$ modify Q_{g+1}^* according to the procedure in Section 3.2
9: $R_{g+1} \leftarrow P_{g+1} \cup Q_{g+1}$
10: $g \leftarrow g + 1$
11: **end while**

4 Test Cases

4.1 Academic Example

Consider the following unconstrained DOP:

$$\max_{\mathbf{x}} f(\mathbf{x}, t) = \frac{\sin(x_1 + x_2 + t/20)}{{x_2}^2 + 1} - \frac{\cos\left(x_1 - x_2 + (t/10000)^2 + 4\right)}{{x_2}^2 + 1} + \frac{{x_1}^2}{80}$$
$$x_i \in [-2\pi, 2\pi] \quad , \quad i = \{1, 2\}.$$

When $t = 98$, the optimum "jumps" from $\mathbf{x} = \mathbf{x_0} = [3.24, -0.16]^T$ to $\mathbf{x} = \mathbf{x_f} = [-3.24, -0.12]^T$. The contour of the function at $t = 98$, and the old and new optima were previously depicted in Figure 2. The UOAP for the adaptation in that time instant is solved by the EA described in Algorithm 1. The adaptation's time interval is set to 0.1, and it is defined for $K = 30$ time instants. The population consists of 250 members, and it is evolved over 300 generations.

For this example, the design variables are considered as simple mechanical components that react to the control force according to Newton's 2^{nd} law: $u_m(t) = i_m a_m(t)$, where i_m is a constant number representing the inertia of the m^{th} design variable. Here $i_1 = 20$ and $i_2 = 10$.

The set of non-dominated solutions in the final population is shown in Figure 4. In Figure 4(a), the trajectories of all the non-dominated solutions are illustrated with gray lines. Three different trajectories from this set are marked with diamonds, triangles and circles. The trajectory marked with diamonds does not require investment of much control force, since it is aimed at the new optimum with minimal changes possible. As a result, its cost is the lowest from all solutions. Since it runs through a region with very low function values, its ISE is very high. The trajectory marked with circles passes along the highest function values on the way to the new optimum. As a result, it has a low ISE value. In order to pass through these local optima, a significant force is required to be applied. Therefore, this solution has a high C value. The trajectory marked with diamonds is a compromise between the two objectives. Its path is shorter than the one of the circles, but it passes through only one local optimum. These insights can be depicted from the Pareto front shown in Figure 4(b). All of the non-dominated solutions are marked with gray dots, and the above three solutions are marked with black markers.

The control forces, positions and function values over time of these three solutions are shown in Figure 5. As mentioned in Section 2.2, the area under the control force function indicates the cost of the adaptation, and the area between the function value in time and the optimal function value indicates the ISE. The trade-off between optimality over time and the cost of the adaptation can be seen here. A larger area of the control force is associated with a smaller area of the error, and vice versa.

In order to evaluate the consistency of the algorithm in finding similar approximated sets, the UOAP was simulated for 100 times using the above genetic

settings. For three of the 100 simulations the algorithm failed to find the trajectories with the lowest ISE values, such as the triangle associated solution in Figure 4. The lowest ISE value in these cases was 0.04.

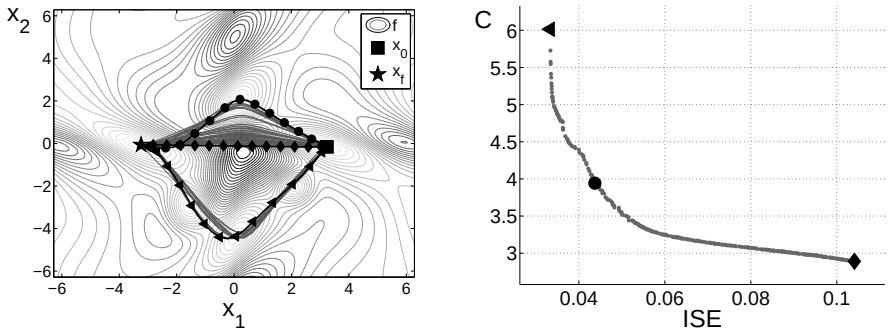


Fig. 4. The final approximated set of the UOAP. Three different trajectories are marked in Figure 4(a), and their associated objective values are marked in Figure 4(b).

4.2 Real World Example

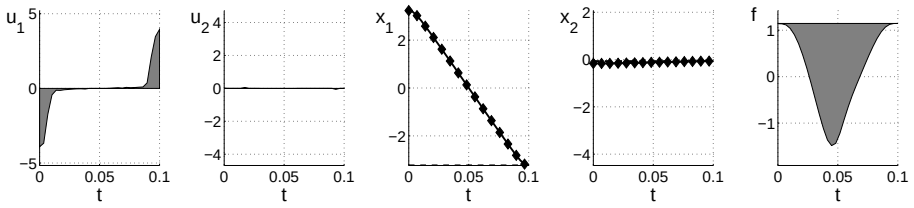
To demonstrate the scope of the proposed optimization approach, the following engineering problem is introduced. A planar robot with three rotating links of equal lengths has to carry a load of mass M over a specific route. The angles of the joints are controlled by the user with servo motors that are free to rotate at any angle. The mass of each motor is m. The robot and related parameters are depicted in Figure 6. The desired location of the load, i.e., the end of the robot's third link, is progressing very slowly. Since it has three degrees of freedom, the robot can keep the load at the desired location with an infinite number of configurations. The optimization goal is to follow the path while minimizing the robot's dimensions, i.e., keeping it as folded into itself as possible. This *dynamic optimization problem* can be formulated as follows:

$$\min_{\boldsymbol{\theta}(t)\in\mathbb{R}^3} \phi\left(\boldsymbol{\theta}(t)\right) \tag{14}$$

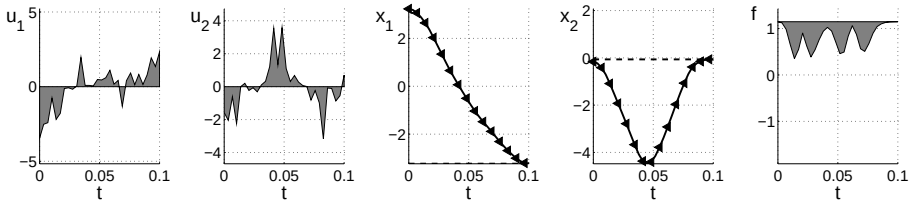
$$s.t. \quad \mathbf{r_e} = \mathbf{P}(t) \tag{15}$$

where ϕ is the stretch of the robot: $\phi = d_1 + d_2 + d_3$, $\boldsymbol{\theta} = [\theta_1, \theta_2, \theta_3]^T$, $\mathbf{r_e}$ is the location of the manipulator's end, and $\mathbf{P}(t)$ is the desired location of the load at time t. The constraint in Eq. (15) has to be satisfied at all times, other than short periods when the robot has to change a configuration for the reason below.

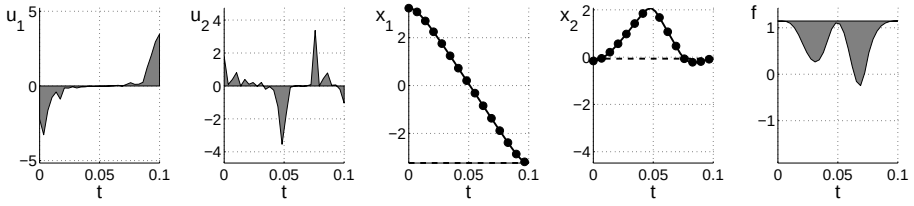
At some time instants there is a single optimal configuration, while at others, the minimal value of ϕ can be achieved by two different configurations. Figure 7



(a) First solution – a high error and low cost.



(b) Second solution – a low error and high cost.



(c) Third solution – a compromise between error and cost.

Fig. 5. The control forces, positions and function values over time for the solutions highlighted in Figure 4.

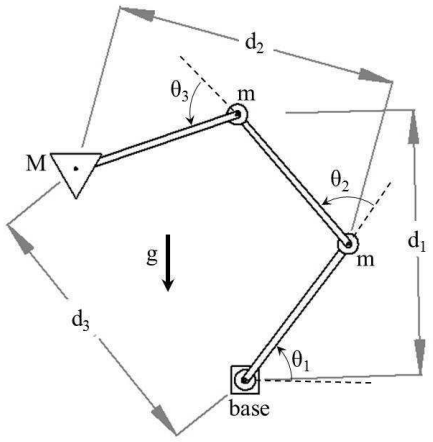


Fig. 6. The robotic manipulator and related parameters.

depicts the solution of the above DOP. The required path of the load $\mathbf{P}(t)$ is marked with arrows, and the optimal configuration is shown for five time instants. When the robot's end is located far from the base, such as in Figures 7(a) and 7(b), the configuration with minimal dimensions has a "Z" shape. When it is closer to the base, such as in Figures 7(c) and 7(d), the optimal configuration has an "N" shape. For the example given, when $t = 1000s$ the robot has to change its configuration from "Z" to "N". The duration of the change is 4 seconds. The question of how to perform that change is considered as the following *utility optimal adaptation problem*. The UOAP follows the main optimization goal, i.e. minimizing the dimensions of the robot, and it also seeks to minimize the power applied to the motors:

$$\begin{aligned} & \min_{\boldsymbol{\theta}(t)\in\mathbb{R}^3} \{E, C\} \ , && t \in [t_0, t_f] \\ s.t. \quad & \boldsymbol{\theta}(t_0) = \boldsymbol{\theta_0} \quad , \quad \boldsymbol{\theta}(t_f) = \boldsymbol{\theta_f} \\ & \left.\frac{\mathrm{d}\boldsymbol{\theta}}{\mathrm{d}t}\right|_{t_0} = \left.\frac{\mathrm{d}\boldsymbol{\theta}}{\mathrm{d}t}\right|_{t_f} = 0 \\ & -\boldsymbol{\tau}_{sat} \leq \boldsymbol{\tau}(t) \leq \boldsymbol{\tau}_{sat} \end{aligned}$$

where $\boldsymbol{\tau}(t) = [\tau_1(t), \tau_2(t), \tau_3(t)]^T$ is a vector with the required torques in the joints to follow the trajectory $\boldsymbol{\theta}(t)$, $\boldsymbol{\tau}_{sat}$ are the saturation values of the motors, E is the integral of $\phi(t)$, and C is the total applied torque over time:

$$E = \int_{t_0}^{t_f} \phi(t)\,\mathrm{d}t \quad , \quad C = \int_{t_0}^{t_f} \left(\textstyle\sum_{i=1}^{3} |\tau_i(t)| \right) \mathrm{d}t.$$

$\boldsymbol{\theta_0}$ and $\boldsymbol{\theta_f}$ are the configurations at Figure 7(b) and 7(c), respectively. $\mathbf{u}(t)$ is calculated by the iterative Newton-Euler dynamic formulation (for more information, see [11]).

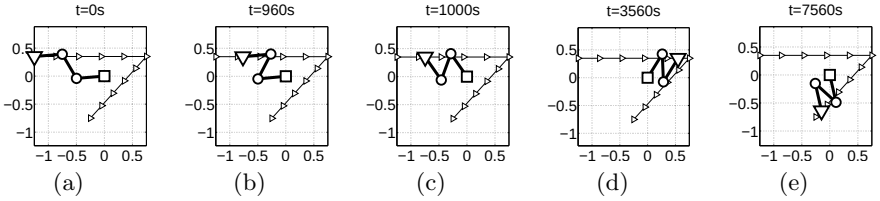


Fig. 7. The configurations with minimal dimensions for five positions of the load. The manipulator's base is marked with a square, and the load is marked with a triangle. The initial configuration of the OAP is the one at Figure 7(b), and the the final configuration is the one at Figure 7(c).

The solutions of the above UOAP are the trajectories of the links and the torques applied by the motors. A controller can be synthesized to follow these

trajectories. Note that the above UOAP is different from the common approach arising from using optimal control theory for designing optimal controllers. A controller designed according to optimal control theory considers the final state $\boldsymbol{\theta_f}$ as an optimization goal, and tries to minimize the control force and the error from $\boldsymbol{\theta_f}$. In contrast, here, the optimization goal is the original goal of the DOP, i.e., minimizing the dimensions of the robot.

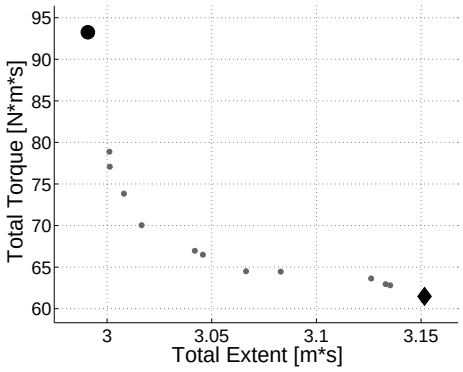
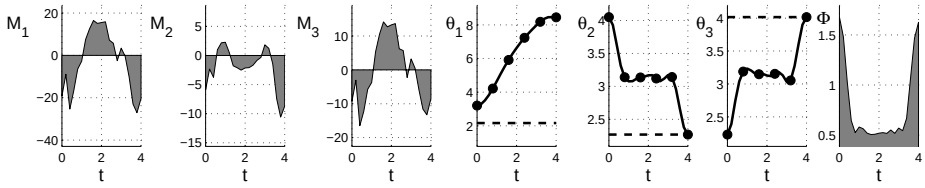


Fig. 8. The approximated set of Pareto optimal solutions for the UOAP, found by the EA.

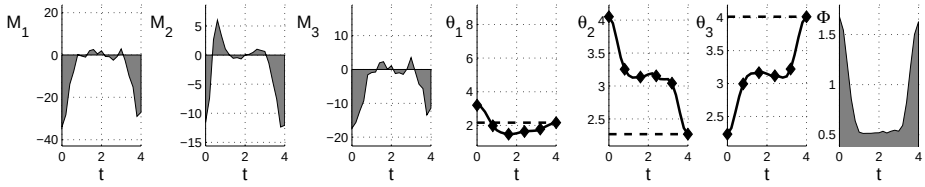
The UOAP was solved by the EA described in Algorithm 1, using the same configuration parameters as in the academic example from the previous section. The final approximated set is depicted in Figure 8. The two extreme solutions, marked with a larger circle and diamond, are shown in Figures 9 and 10. The trajectories of the torques, the joint angles and the extent are shown in Figure 9, and the dynamic behavior of the robot is shown in Figure 10. Note that in order to fold more quickly, under the limits of the saturation torque values, the solution marked with a circle rotates around its base. The solution marked with a diamond balances vertically until it needs to stretch again in the other configuration.

5 Conclusions and Future Work

In this paper, a new optimization problem was introduced: *Optimization of Adaptation Problem*. The problem deals with situations when a change in design is required in order to remain optimal in a changing environment. The required change is found by solving a DOP prior to the formulation of the OAP. The issues of optimality during the adaptation process and the cost of the adaptation were discussed, and the OAP was defined as a MOP for minimizing the cost and maximizing the function value during the adaptation. A relaxed version of the OAP was defined as a *Utility based Optimization of Adaptation Problem*, and an

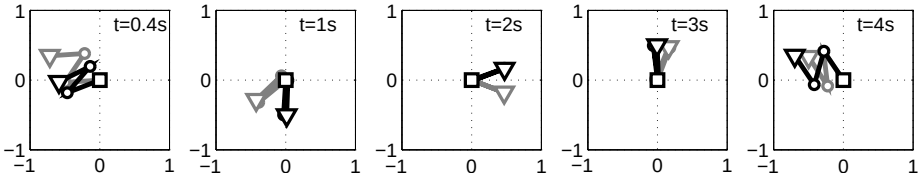


(a) One alternative – small dimensions and high cost. Note that the final θ_1 is 2π larger than the desired final value. That means the manipulator performs a full turn around its base.

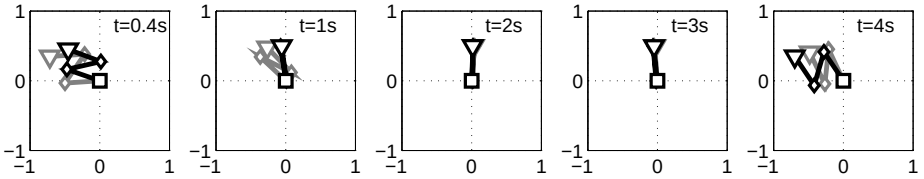


(b) Second alternative – larger dimensions and smaller cost.

Fig. 9. Trajectories of the joint torques and angles and the extent of the robot in time.



(a) One alternative – small dimensions and high cost.



(b) Second alternative – larger dimensions and smaller cost.

Fig. 10. Positions of two Pareto optimal solutions at different time samples. The lighter configuration is 0.4s prior to the black one.

EA was proposed in order to solve it. The EA was tested on two examples: a theoretic mathematical function and a problem of robotic arm control. For both examples, the EA was able to find a set of trade off solutions that enable the decision maker to choose whether to adapt in a minimum cost manner or to invest more resources in order to maintain high function values along the adaptation.

As future work, the new approach should be integrated and tested on more real life applications. This paper dealt with a single objective DOPs. The issue of optimal adaptation for multiobjective DOPs should be studied as well. The method should be also extended to deal with cases where the future function values and costs are subject to uncertainties.

Acknowledgements This research was supported by a Marie Curie International Research Staff Exchange Scheme Fellowship within the 7^{th} European Community Framework Programme.

References

1. Rose, M., Lauder, G.: Adaptation. Academic Press, San Diego, CA (1996)
2. Ferguson, G., Murphy, J., Ramanamanjato, J., Raselimanana, A., et al.: The panther chameleon: color variation, natural history, conservation, and captive management. Krieger Publishing Company (2004)
3. Ruseckaite, R., Lamb, T., Pianta, M., Cameron, A.: Human scotopic dark adaptation: Comparison of recoveries of psychophysical threshold and erg b-wave sensitivity. Journal of Vision **11**(8) (2011)
4. Jacobs, R., Lundby, C., Robach, P., Gassmann, M.: Red blood cell volume and the capacity for exercise at moderate to high altitude. Sports Medicine **42**(8) (2012) 643–663
5. Branke, J.: Evolutionary approaches to dynamic optimization problems-updated survey. In: GECCO Workshop on evolutionary algorithms for dynamic optimization problems. (2001) 27–30
6. Avigad, G., Eisenstadt, E., Goldvard, A., Salomon, S.: Transient responses optimization by means of set-based multi-objective evolution. Engineering Optimization **44**(4) (2011) 407–426
7. Réne, M., J.J.S., Ole, S.: Topology optimization for transient response of photonic crystal structures. Optical Society of America. Journal B: Optical Physics **27**(10) (2010) 2040–2050
8. Toscano, R.: A simple robust pi/pid controller design via numerical optimization approach. Journal of Process Control **15**(1) (2005) 81 – 88
9. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: Nsga-ii. Evolutionary Computation, IEEE Transactions on **6**(2) (2002) 182–197
10. Deb, K., Agrawal, R.: Simulated binary crossover for continuous search space. Complex systems **9**(2) (1995) 115–148
11. Craig, J.: Introduction to Robotics. Pearson Education, Inc. (2005)