Optimization in Predictive Control Algorithm

JAN ANTOŠ, MAREK KUBALČÍK
Tomas Bata University in Zlín, Faculty of Applied Informatics
Nám. T. G. Masaryka 5555, 760 01 Zlín
CZECH REPUBLIC
antos@fai.utb.cz, kubalcik@fai.utb.cz
http://web.fai.utb.cz

The term predictive control designates a class of control methods which are suitable for control of various kinds of systems. Predictive control is essentially based on discrete or sampled models of a process. It is possible to use various models which describe a real system sufficiently accurately. The model is used for computation of predictions of the systems output. On the basis of the predictions and past values of the input and output of the system we can optimize the control process. A cost function which is optimized is a function of difference of the manipulated variable. An analytical solution can be obtained by differentiation of the cost function. One of the major advantages of predictive control is its ability to do on-line constraints handling in a systematic way. For a constrained case the analytical solution may not be located in an allowed area and we need to use another way of optimization. The optimization problem is necessary to be solved in each sampling period, which is computationally demanding. Various kinds of optimization algorithm can be used. The contribution is focused on an analysis of the cost function and effects of constraints to an allowed area. The analysis can help with choosing of suitable methods and algorithms.

Key-Words: - predictive control, cost function, constraints, optimization, simulation

1 Introduction

The essential idea of predictive control [1], [2], [3], [4] is based on the possibility to predict behaviour of a system using its model. Predictive control can be divided into several parts.

A predictor defines relation between past and future values. There are a lot of methods how to obtain prediction equations. These methods are based on a model of the controlled system. A range of various models can be used (for instance transfer function, ARMA, neural network etc.). A widely used model in predictive control is the CARIMA model [5] which directly contains a difference of the manipulated variable. This model can be written in the following form

$$Ay(k) = Bu(k-1) + \frac{C}{\Delta} n(k) \tag{1}$$

where polynomials A and B describes a transfer function of the system. $\Delta = 1 - z^{-1}$, y is the output variable, u is the manipulated value and n is a nonmeasurable noise which is assumed to have zero mean value and constant covariance. C is a colouring polynomial.

On the basis of this model the predictor can be calculated. The predictor [6] can be divided into two parts: a free response, which is that part of the systems response which is determined by past values of the systems inputs and outputs, and a forced response, which is determined by future control increments.

$$\vec{\boldsymbol{y}} = \boldsymbol{G}\Delta\vec{\boldsymbol{u}} + \boldsymbol{X}\begin{pmatrix} \overleftarrow{\boldsymbol{y}} \\ \overleftarrow{\boldsymbol{u}} \end{pmatrix} \tag{2}$$

where $\vec{\boldsymbol{y}}$ and $\vec{\boldsymbol{u}}$ are future values, $\overleftarrow{\boldsymbol{y}}$ and $\overleftarrow{\boldsymbol{u}}$ are past values. Matrices $\boldsymbol{G}$ and $\boldsymbol{X}$ contain coefficients which are necessary to be calculated in order to obtain the predictor.

The predictor can be also written in the form shown in eq (3)

$$\hat{\boldsymbol{y}} = \boldsymbol{G}\Delta\tilde{\boldsymbol{u}} + \boldsymbol{y_0} \tag{3}$$

where $\boldsymbol{y_0}$ is the free response of the process.

Matrix $\boldsymbol{G}$ contains values of the step sequence and it can be written in the following form

$$\boldsymbol{G} = \begin{pmatrix} g_0 & 0 & 0 & \cdots & 0 \\ g_1 & g_0 & 0 & \cdots & 0 \\ g_2 & g_1 & g_0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ g_{N_2-1} & g_{N_2-2} & g_{N_2-3} & \cdots & g_0 \end{pmatrix} \tag{4}$$

Another part of predictive control is a cost function which can be defined as a sum of squares of control errors and squares of differences of the

ISBN: 978-960-474-374-2

manipulated variable. The cost function can be written according to eq. (5) as

$$\boldsymbol{J} = (\hat{\boldsymbol{y}} - \boldsymbol{w})^T (\hat{\boldsymbol{y}} - \boldsymbol{w}) + \lambda \tilde{\boldsymbol{u}}^T \tilde{\boldsymbol{u}}$$
$$\boldsymbol{J} = (\boldsymbol{G}\Delta\tilde{\boldsymbol{u}} + \boldsymbol{y_0} - \boldsymbol{w})^T (\boldsymbol{G}\Delta\tilde{\boldsymbol{u}} + \boldsymbol{y_0} - \boldsymbol{w}) + \lambda \tilde{\boldsymbol{u}}^T \tilde{\boldsymbol{u}} \tag{5}$$

where λ is a weighting factor which is another degree of freedom.

It can be also written in the form, as presented in eq. (6)

$$\boldsymbol{J} = c_0 + 2\boldsymbol{g}^T \tilde{\boldsymbol{u}} + \tilde{\boldsymbol{u}}^T \boldsymbol{H} \tilde{\boldsymbol{u}} \tag{6}$$

where $\boldsymbol{g}$ is a gradient of the cost function and $\boldsymbol{H}$ is the Hessian matrix. The gradient and Hessian matrix can be written in the form shown in eq. (7)

$$\boldsymbol{g}^T = \boldsymbol{G}^T (\boldsymbol{y_0} - \boldsymbol{w})$$
$$\boldsymbol{H} = \boldsymbol{G}^T \boldsymbol{G} + \lambda \boldsymbol{I} \tag{7}$$

The solution can simply be obtained by derivative of the cost function. Equation (8) considers the vector of the manipulated variable

$$\tilde{\boldsymbol{u}} = -\boldsymbol{H}^{-1} \boldsymbol{g}$$
$$\tilde{\boldsymbol{u}} = \boldsymbol{K}(\boldsymbol{w} - \boldsymbol{y_0}) \tag{8}$$

where $\boldsymbol{K} = (\boldsymbol{G}^T \boldsymbol{G} + \lambda \boldsymbol{I})^{-1} \boldsymbol{G}^T$.

Only the first element is used and the whole procedure is repeated in the next sampling period. It is called the Receding Horizon concept.

There are three important horizons in predictive control. N_1 and N_2 are minimum and maximum prediction horizons. It means horizons over which are predicted output values. N_u is called a control horizon and it defines length of the vector of control increments.

In technical practice often occur constraints of variables which causes that the analytical solution of the cost function is located outside the allowed area. In a constrained case alternative methods of optimization must be used to obtain the solution. The optimization must be as effective as possible.

In the past, the predictive control was mostly applied for control of systems with large time constants. Nowadays it is also possible to apply it for control of systems with faster dynamics because of increasing computational power. But the optimization problem must be solved in each sampling period and computationally effective algorithms are required. The aim of the paper is to analyze the cost function and effects of constraints to an allowed area.

2 Constraints and cost function

Computational costs of solving the optimization problem is dependent on a shape of the cost function and the allowed area.

2.1 Cost function

As it is presented in eq. (5) the cost function is a quadratic function with the shape depicted in Fig. 1.

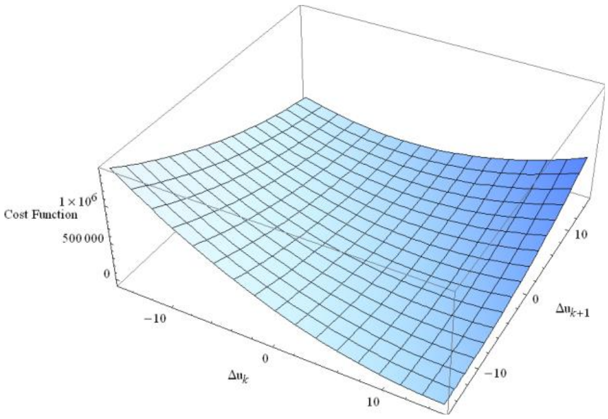

Fig. 1. Cost function

Dimension of the cost function is equal to $N_u + 1$. It is not possible to show the n-dimensional function in 3D space in case $N_u > 2$. But it is possible to depict its cuts with fixed values in other axes. In Fig. 1 is depicted the cost function for the first sampling step of the control process. In the following steps the cost function looks similarly. According to its shape it can be solved by the derivative. The cost function is unimodal with one local minimum.

2.1 Constraints

The values of variables in a real system are usually constrained. The constraints are mostly given by security conditions or technical limits. In this case the cost function is limited by a specific area.

We can consider three types of constraints. The first one is constraint of difference of the manipulated variable. It can be written in the form represented in eq. (9).

$$\Delta\boldsymbol{u_{min}} \le \Delta\boldsymbol{u} \le \Delta\boldsymbol{u_{max}} \tag{9}$$

It can be also expressed by a set of equations (10)

$$\begin{aligned} \Delta u(k) &\ge \Delta u_{\min} \\ \Delta u(k+1) &\ge \Delta u_{\min} \\ &\vdots \\ -\Delta u(k) &\ge -\Delta u_{\max} \\ -\Delta u(k+1) &\ge -\Delta u_{\max} \\ &\vdots \end{aligned} \tag{10}$$

The system of equations can be written in a matrix form, as presented in eq. (11)

$$\begin{pmatrix} \boldsymbol{I} \\ -\boldsymbol{I} \end{pmatrix} \Delta \boldsymbol{u} \geq \begin{pmatrix} \mathbf{1}\Delta \boldsymbol{u}_{min} \\ -\mathbf{1}\Delta \boldsymbol{u}_{max} \end{pmatrix} \tag{11}$$
$$\boldsymbol{A}\Delta\boldsymbol{u} \geq \boldsymbol{b}$$

where $\boldsymbol{I}$ is an identity matrix of dimension $N_u \times N_u$.

Constraints of difference of the manipulated variable leads to the shape of the allowed area in the form of nD-cube which is depicted in Fig. 2.

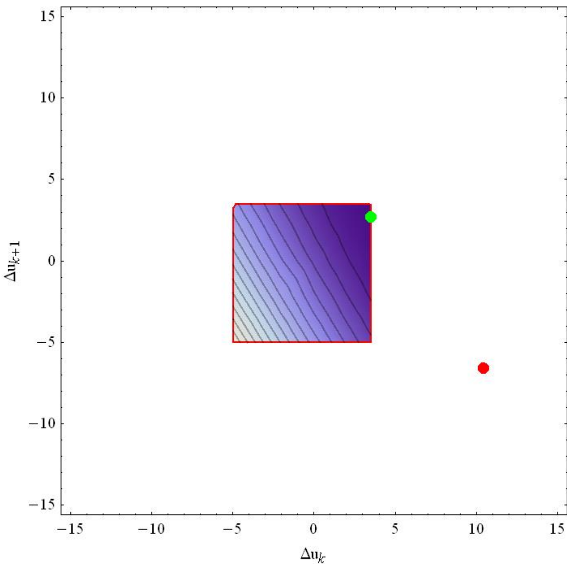

Fig. 2. Constraints of difference of manipulated variable

The red point is a minimum obtained by derivative of the cost function. The green point is the global minimum of the allowed area.

The next type is the constraint of the manipulated variable. It can be expressed by equation (12).

$$\boldsymbol{u}_{min} \leq \boldsymbol{u} \leq \boldsymbol{u}_{max} \tag{12}$$

A set of equations is in form (13).

$$\begin{aligned} \Delta u(k) &\geq u_{\min} - u(k-1) \\ \Delta u(k) + \Delta u(k+1) &\geq u_{\min} - u(k-1) \\ &\vdots \\ -\Delta u(k) &\geq -u_{\max} + u(k-1) \\ -\Delta u(k) - \Delta u(k+1) &\geq -u_{\max} + u(k-1) \\ &\vdots \end{aligned} \tag{13}$$

And the matrix form can be expressed as follows

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{T} \\ -\boldsymbol{T} \end{pmatrix}, \boldsymbol{b} = \begin{pmatrix} \mathbf{1}\boldsymbol{u}_{min} - \mathbf{1}\boldsymbol{u}(\boldsymbol{k}-\mathbf{1}) \\ -\mathbf{1}\boldsymbol{u}_{max} + \mathbf{1}\boldsymbol{u}(\boldsymbol{k}-\mathbf{1}) \end{pmatrix} \tag{14}$$

where $\boldsymbol{T}$ is a lower triangular matrix with ones in the non-zero positions.

The shape of the allowed area is depicted in Fig. 3.

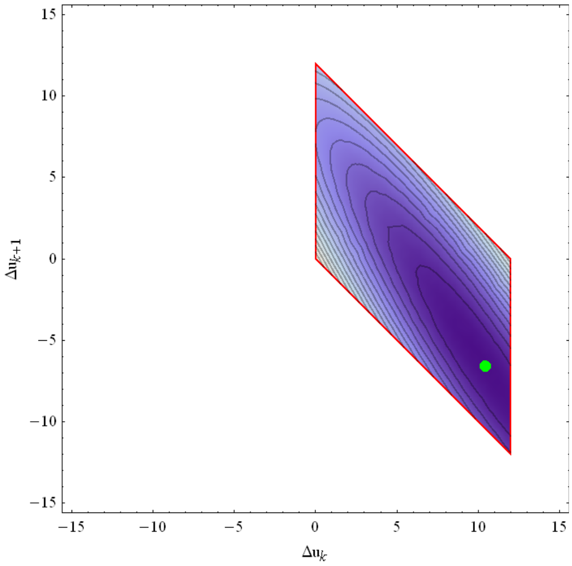

Fig. 3. Constraint of manipulated variable

As it can be seen in figure 3 the range of $\Delta u(k)$ is in the interval form. But the next difference value $\Delta u(k+1)$ is dependent on the previous value of $\Delta u(k)$. The interval of the vertical axis is being continuously moved which causes the shape depicted in Fig. 3. Every next interval of other axes in case of a multidimensional problem is also moved according to previous values of differences of the manipulated variable. The allowed area is formed as a subspace in the nD space.

The third type of constraint is defined as an interval of the output variable which can be written in the following form

$$\boldsymbol{y}_{min} \leq \boldsymbol{y} \leq \boldsymbol{y}_{max} \tag{15}$$

There must be defined a relation between $\boldsymbol{y}$ and $\Delta u(k)$. Equation (3) can be substituted to equation (15) and the constraint is expressed as follows

$$\boldsymbol{y}_{min} \leq \boldsymbol{G}\Delta\tilde{\boldsymbol{u}} + \boldsymbol{y}_0 \leq \boldsymbol{y}_{max} \tag{16}$$

Equation (16) can be modified

$$\begin{aligned} \boldsymbol{G}\Delta\tilde{\boldsymbol{u}} &\geq \boldsymbol{y}_{min} - \boldsymbol{y}_0 \\ -\boldsymbol{G}\Delta\tilde{\boldsymbol{u}} &\geq -\boldsymbol{y}_{max} + \boldsymbol{y}_0 \end{aligned} \tag{17}$$

The matrices $\boldsymbol{A}$ and $\boldsymbol{b}$ take the following form

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{G} \\ -\boldsymbol{G} \end{pmatrix}, \boldsymbol{b} = \begin{pmatrix} \mathbf{1}\boldsymbol{y}_{min} - \boldsymbol{y}_0 \\ -\mathbf{1}\boldsymbol{y}_{max} + \boldsymbol{y}_0 \end{pmatrix} \tag{18}$$

If the mapping between manipulated and output values is feasible then the interval of output constraint corresponds to the interval of the manipulated variable constraint. In this case the

ISBN: 978-960-474-374-2

shape of the allowed area, which is depicted in Fig. 4, is similar to the shape depicted in Fig. 3.

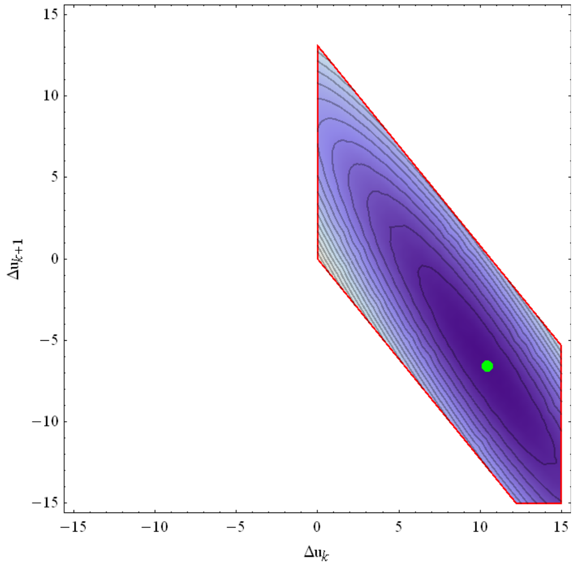

Fig. 4. Constraint of output variable

It is obvious that positions of the local and global minimums can be different and they are dependent on the cost function and constrains. The constraints can be combined each other and the allowed area is in the intersection of them, which is depicted in Fig. 5.

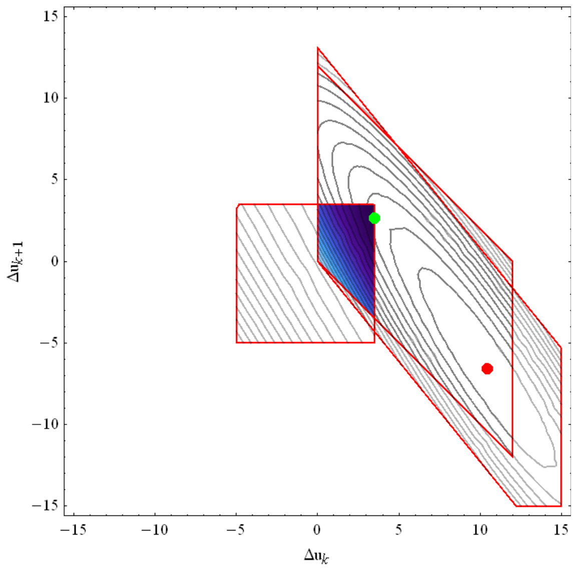

Fig. 5. Combination of constraints

The allowed area is continuous but we can consider other types of constraints e.g. discrete values, multi-intervals etc. In these cases a solution of the optimization problem may be more complicated because of the shape of the allowed area.

3 Simulation and algorithms

2.1 Simulation

As a simulation example is presented control of the following system of the second order

$$G(s) = \frac{35.88}{(519.29s+1)(40.80s+1)} \quad (19)$$

The horizons were set as N_1 = 1, N_2 = 3 and N_u = 2 in order to have a possibility to show graphs (3D cost function and 2D allowed area).

The constraints were set to the values shown in eq. (20).

$$\begin{aligned} -5 &\le \Delta u \le 3.5 \\ 0 &\le u \le 12 \\ 0 &\le y \le 310 \end{aligned} \quad (20)$$

The optimization problem was solved by Hill Climbing algorithm [7]. Time responses of control are depicted in Fig. 6 and Fig. 7.

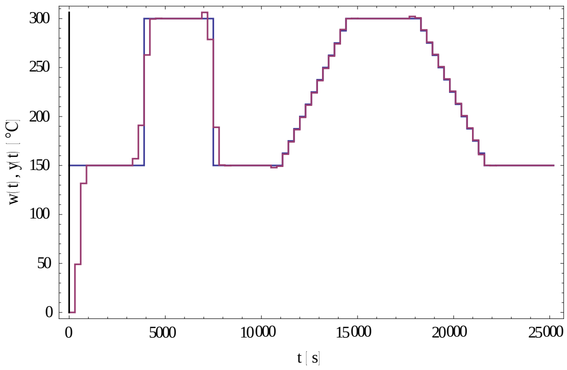

Fig. 6. Setpoint and output variables of control process

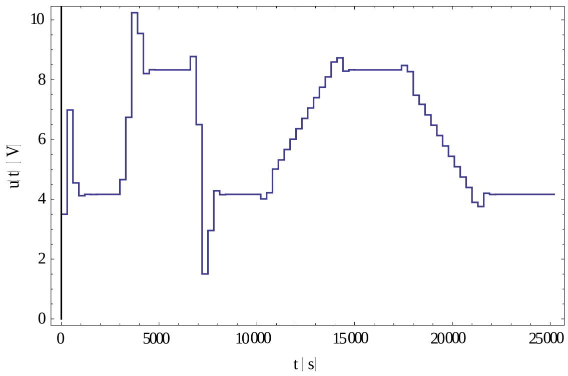

Fig. 7. Manipulated variable of control process

We can study the shape of the allowed area and values of the manipulated variables. It is obvious that the control error for k = 1 is large and there are also defined constraints of difference of the manipulated variable, manipulated variable and output variable. The shape of the allowed area is depicted in Fig. 8.

ISBN: 978-960-474-374-2

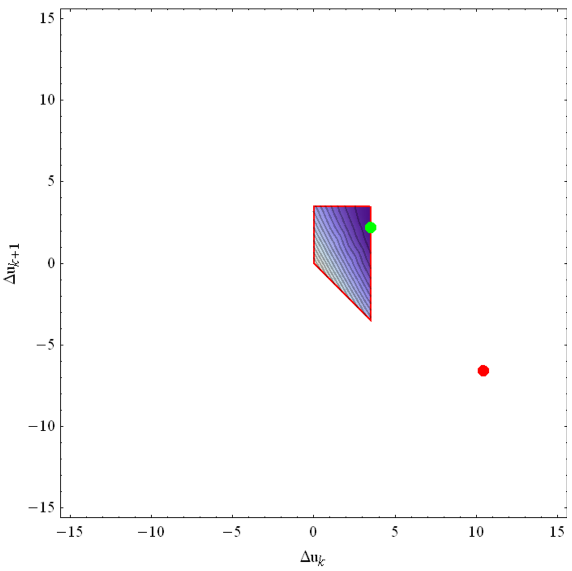

Fig. 8. Allowed area for k = 1

It is obvious that the analytical solution is located outside the allowed area. For the unconstrained case the analytically computed value is $\Delta u(k) = 10.4437$. But the solution which was found within the allowed area is 3.5 because constraint of difference of the manipulated variable is $\Delta u_{\max} = 3.5$.

The allowed area for k = 2 is depicted in Fig. 9.

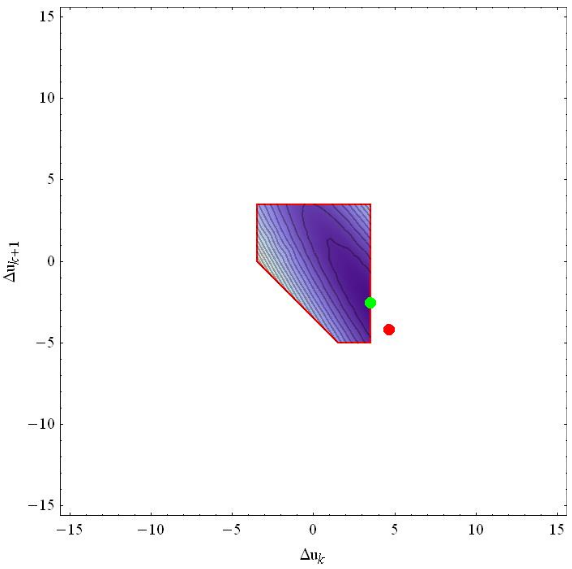

Fig. 9. Allowed area for k = 2

In this step the analytical solution is much closer to the solution obtained inside the allowed area. The allowed area for k = 3 is depicted in Fig. 10.

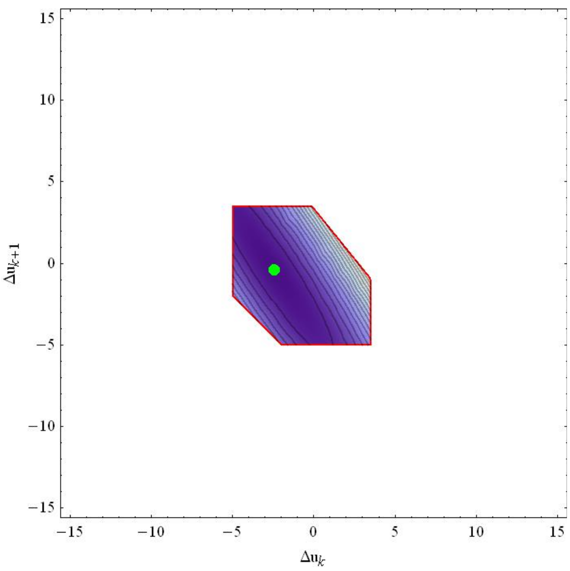

Fig. 10. Allowed area for k = 3

In the third step the analytical solution is inside the allowed area. In most cases the analytical solution is inside the allowed area. In these cases the solution can be obtained by derivative of the cost function. In other cases must be used a suitable algorithm in order to find a solution inside the allowed area. These cases are mainly located around sudden changes of the reference signal.

2.1 Algorithms

The searching of the solution can be divided into two phases. In the first stage is being searched an arbitrary solution which is placed inside the allowed area. The second phase consists of searching the best solution within the allowed area.

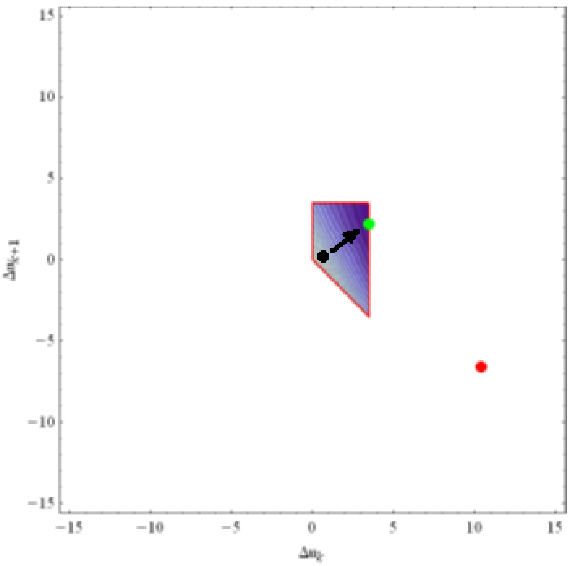

Fig. 11. Searching solution inside the allowed area

The simplest way how to find the allowed area is the random walks method [7]. This method is not suitable for solving of higher dimensional problems because its efficiency rapidly decreases with increasing dimension.

The allowed area can be also found by the method of penalisation of the cost function outside of the allowed area (soft-constraints[1]). The question is how to penalise the cost function in order to find the allowed area quickly. For example multiplication of the cost function only increases value of the cost function but the gradient may stay similar. In this case the gradient methods may not converge to the allowed area. The analytical solution can be used as the initial estimate. Also point [0,0] is a good candidate to be inside the allowed area. But zero values of difference of the manipulated variable correspond to the constant value of the manipulated variable and the free response of the system may not fulfil requirements on constraints of the output variable.

The next method how to find the allowed area is to solve a system of inequalities with fixed future values of difference of manipulated variables.

The second phase can be solved by gradient methods because of the shape of the cost function. A general equation of the gradient method can be written in the form shown in eq. (21)

$$x(i+1) = x(i) + f(g(x)) \tag{21}$$

where $f(g(x))$ is a function of the gradient of the cost function.

One of the most suitable methods is quadratic programming [8], [9]. This method is often used in predictive control but it is also possible to use other methods which enable to solve this problem such as evolutionary algorithms [7]. Disadvantage of quadratic programming is higher computational cost and computational time.

In cases when a model, initial conditions and a setpoint are known the problem can be solved offline [10].

4 Conclusion

In the paper were introduced basic aspects of solving the optimization problem in predictive control. In an unconstrained case it is possible to find the solution by derivate of the cost function. The cost function is a quadratic function, which is unimodal, and it has one local minimum. In a constrained case a shape of the allowed area depends on a type and values of constraints. In this case the analytical solution may be located outside of the allowed area. There are lots of algorithms which enable to solve this problem. One of the most effective methods is called quadratic programing but it is also possible to solve this problem by many other methods for example by methods based on evolutionary algorithms. The crucial issue of the optimization are computational costs because it must be solved on-line in each sampling period. Searching of the solution can be divided into two phases. In the first phase there must be found the allowed area. In the second phase there must be found the best solution inside the allowed area.

Acknowledgments:
Authors are thankful to Internal Grant Agency (IGA/FAI/2014/009) of Tomas Bata University in Zlín, Czech Republic for financial support.

References:

[1] D. W. Clarke, C. Mohtadi, P. S. Tuffs, Generalized predictive control, part I: the basic algorithm. *Automatica*, 23, 1987, 137-148.

[2] D. W. Clarke, C. Mohtadi, P. S. Tuffs, Generalized predictive control, part II: extensions and interpretations. *Automatica*, 23, 1987, 149-160.

[3] M. Morari, J. H. Lee, Model predictive control: past, present and future. *Computers and Chemical Engineering*, 23, 1999, 667-682.

[4] R. R. Bitmead, M. Gevers, V. Hertz, *Adaptive Optimal Control. The Thinking Man's GPC,* Prentice Hall, Englewood Cliffs, New Jersey, 1990.

[5] E. F. Camacho, C. Bordons, *Model Predictive Control*, Springer-Verlag, London, 2004.

[6] J. A. Rossiter, *Model Based Predictive Control: a Practical Approach,* CRC Press, 2003.

[7] Back T., Fogel D. B., Michalewicz Z., *Handbook of evolutionary algorithms*, Oxford University Press, 1997, ISBN 0750303921

[8] G.M. Lee, N.N. Tam, and N.D. Yen, Quadratic Programming and Affine Variational Inequalities: A Qualitative Study, Springer, 2005.

[9] Z. Dostál, Optimal Quadratic Programming Algorithms: With Applications to Variational Inequalities, New York: Springer, 2009.

[10] Kvasnica, M.; Rauova, I.; Fikar, M., "Automatic code generation for real-time implementation of Model Predictive Control," Computer-Aided Control System Design (CACSD), 2010 IEEE International Symposium on , vol., no., pp.993,998, 8-10 Sept. 2010

---

[1] In case of soft-constraints cost functions of points which are located outside of the allowed area are penalized.