Scientific Journals
of the Maritime University of Szczecin

Zeszyty Naukowe
Akademii Morskiej w Szczecinie

2015, 43 (115), 22–28
ISSN 1733-8670 (Printed)
ISSN 2392-0378 (Online)

Methods of controller synthesis using linear matrix inequalities and model predictive control

Anna Miller[1], Monika Rybczak[2]

Gdynia Maritime University, Department of Ship Automation
81–87 Morska St., 81-225 Gdynia, Poland
e-mails: [1]anwas@am.gdynia.pl, [2]rybczak.monika@gmail.com

Key words: controller synthesis, LMI, MPC, predictive controller, controller synthesis methods, optimization methods

Abstract
Controllers based on linear matrix inequalities (LMI) and model predictive control (MPC) both use optimization methods; there are however significant differences between them. In case of LMI controllers, optimization is carried out during controller synthesis, because LMI's are an optimization tool that requires a linear programming problem being solved. With MPC controllers, however, optimization methods are not used as much in controller synthesis as in controller algorithm operation, to determine optimal control signal values based on the found minimum of the criteria function. A square function is used with boundaries from above and below, which requires a square programming problem, with boundaries for decision variables, being solved. In this paper controller synthesis methods using LMI and MPC are shown, with a focus on the steps that need to be performed, and a comparison of both methods.

Introduction

Vessel movement control is becoming an increasingly important part of modern automation systems installed on ships. Increasingly, multivariable controllers are used, which are a completely different standard compared to the widely used monovariable controllers (used for a ship's course). There are many types of multivariable controllers used in marine automation systems and many methods for their synthesis. This paper focuses on comparing controller synthesis using linear matrix inequalities (LMI) and model predictive control (MPC) methods. The most frequently quoted publications about LMI are (Boyd et al., 1994) and (Weiland & Scherer, 2005). In Poland, two of the first papers on control theory with LMI were "Controller synthesis, selected classical and optimization methods" by Koziński (Koziński, 2004) and "Analysis and synthesis of multidimensional system classes using linear matrix inequality methods" by Paszke (Paszke, 2005). MPC is not one, specific, clearly defined algorithm. It is more of an expanded strategy of optimal (or suboptimal) control used in many branches of industry. Controllers based on predictive algorithms were first used, in practice, in the chemical and petrochemical industry (Xi, Li & Lin, 2013), where controlled processes are usually slow changing. It is a control strategy that allows implementing boundaries to both input and control variables. MPC is a control method based on controlling object (or process) models, and using them to predict future object behaviour based on determined control signals and known starting conditions. Archetypes for this now growing control strategy were the publications (Clarke, Mohtadi & Tuffs, 1987a; 1987b). Our current paper presents the specific stages of controller synthesis using LMI and MPC methods, with a summary that highlights the difficulties and benefits of both methods.

Method of controller synthesis using linear matrix inequalities

The method of LMI is one of a number of methods of convex optimization used for controller design. LMI conditions create a set of boundaries called a convex set; each of these boundaries is described by an inequality that affects system dynamics. LMI are used for:

– Analysis of control systems, as in stability and quality control;
– Controller synthesis with various controlled object connection possibilities e.g. serial, parallel or static state space controllers.

Controller synthesis using LMI requires several steps, which are described below and shown in Figure 2:

1. Identification of the linear controlled object model.
2. Controller synthesis is divided into the following sub-steps:
   a) Finding a symmetric positive definite matrix P, also known as the feasibility problem;
   b) User defining boundaries for the convex set, placed in the left half-plane of complex variable plane s;
   c) Calculation of H_∞ standard;
   d) Calculation of H_2 standard.
3. Minimization of objective function, which means selecting values of H_∞ and H_2 standards so that a weighted sum can be calculated.

After identifying the linear controlled object model, which is done by analyzing the relations between the input and output signals, state equations are formed. Figure 1 shows a standard control system structure used for optimization (Boyd et al., 1994; Weiland & Scherer, 2005).

Controller state space equations:

$$\begin{cases} \dot{x} = Ax + B_w w + B_u u \\ z = C_z + D_{zw} w + D_{zu} u \\ y = C_y + D_{yw} w + D_{yu} u \\ z_2 = C_{z2} + D_{z2w} w + D_{z2u} u \\ z_\infty = C_{z\infty} + D_{z\infty w} w + D_{z\infty u} u \end{cases} \tag{1}$$

where: x – dependent variables vector, state vector; z, y – output value vectors; u, w – response vectors, input function vectors; A – state matrix; B_w, B_u – control matrix for w and u signals; C_z, C_y, C_{z2}, $C_{z\infty}$ – output matrix for z, z_2, z_∞ and y signals; D_{zw}, D_{zu}, D_{yw}, D_{yu}, $D_{z\infty w}$, D_{z2w}, $D_{z\infty u}$, D_{z2u} – transmission matrix for specific signals; and where it is assumed that D_{yu}, D_{yw}, $D_{z2w} = 0$, which means that the output signals y and z_2 are not directly related to the response signal u. The identified model of the controlled object is defined as linear and stationary with the transmittance of a closed loop system $G(s)_2$ for the H_2 standard, which describes the relation between input signal w and output signal z_2; and a second transmittance of a closed loop system $G(s)_\infty$ for the standard H_∞, which describes the relation between input signal w and output signal z_∞.

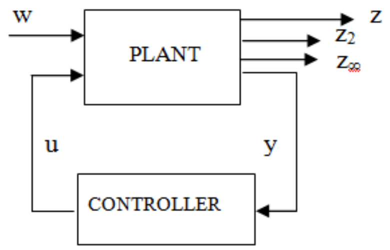

Figure 1. Control structure for LMI

The following state space equations describe the controller:

$$\begin{cases} \dot{x} = A_c x_c + B_c y \\ u = C_c x_c + D_c y \end{cases} \tag{2}$$

where: x_c – controller state vector; y – measured signal vector; u – control signal vector; A_c, B_c, C_c, D_c – controller matrices.

A closed loop system, where w is the input signal and z is the output signal, is described by the state equations below. The specific matrices of the closed loop system are as follows:

$$A_{cl} = \begin{bmatrix} A + B_u D_c C_y & B_u B_c \\ B_c C_y & A_c \end{bmatrix} \qquad B_{cl} = \begin{bmatrix} B_w + D_c D_{yw} \\ B_c D_{yw} \end{bmatrix} \tag{3}$$

$$\begin{cases} \dot{x}_{cl} = A_{cl} x + B_{cl} w \\ u = \begin{bmatrix} D_c C_y & C_c \end{bmatrix} \begin{bmatrix} x \\ x_c \end{bmatrix} + \begin{bmatrix} D_c D_{yw} \end{bmatrix} w \\ z = \begin{bmatrix} C_z + D_{zu} D_c C_y & D_{zw} C_c \end{bmatrix} \begin{bmatrix} x \\ x_c \end{bmatrix} + \\ \quad + \begin{bmatrix} D_{zu} D_c D_{yw} + D_{zw} \end{bmatrix} w \\ z_2 = \begin{bmatrix} C_{z2} + D_{z2u} D_c C_y & D_{2u} C_c \end{bmatrix} \begin{bmatrix} x \\ x_c \end{bmatrix} + \\ \quad + \begin{bmatrix} D_{z2u} D_c D_{yw} + D_{z2w} \end{bmatrix} w \\ z_\infty = \begin{bmatrix} C_{z\infty} + D_{z\infty u} D_c C_y & D_{z\infty w} C_c \end{bmatrix} \begin{bmatrix} x \\ x_c \end{bmatrix} + \\ \quad + \begin{bmatrix} D_{z\infty u} D_c D_{yw} + D_{z\infty w} \end{bmatrix} w \end{cases} \tag{4}$$

For clarity of formulas the following standard designations are used:

$$A_{cl} = A,\ B_{cl} = B,\ C_z,\ C_{z2},\ C_{z\infty} =$$
$$= C,\ D_{zw},\ D_{zu},\ D_{yw},\ D_{yu},\ D_{z\infty w},\ D_{z2w},\ D_{z\infty u},\ D_{z2u} = D.$$

After defining state equalities for a closed loop system, the following goal of control synthesis can be formulated (Koziński, 2004):

For a given controlled object and a given transmittance standard of a closed loop system, from all controllers for which the closed loop system is stable, find a controller that will minimize the transmittance standard of a closed loop system between "w" and "z" signals.

The first step is finding an answer to the question of whether there exists a solution x to the following inequality:

$$A(x) \succ 0$$

where: the symbol $A \succ 0$ means that matrix A is symmetric $A \in R^{nxn}$ and for every vector $x \in R^n$ it is true that $x^T Ax > 0$, which is called positive determination, which means, if the eigenvalues of matrix $A(x)$ are negative and if they are placed in the left half-plane of complex variable plane s. Additionally, a positively symmetrical matrix ($P = P^T \succ 0$) must be found for an inequality known as the Lyapunov inequality ($A^T P + PA < 0$) which is closely related, by matrix A, to the controlled object described by the state equalities. Fulfilling this condition is the first LMI problem, and is called the feasibility problem.

The second step requires the user to define a bounded convex set area of solutions placed in the left half-plane of complex variable plane s, in order to determine dynamic parameters of the controlled object. The shape and parameters of the convex set area are empirically determined by the user (Rybczak, 2014).

The third step is the approximation of the $H\infty$ standard. A positively symmetrical matrix P must be found which fulfils the condition below:

$$\begin{bmatrix} A^T P + PA & PB & C^T \\ B^T P & -\gamma_\infty \mathrm{I} & D^T \\ C & D & -\gamma_\infty \mathrm{I} \end{bmatrix} \prec 0 \tag{5}$$

The user must empirically determine the value of γ_∞ coefficient which fulfils the inequality

$$\|G(s)\|_\infty^2 < \gamma_\infty{}^2 \tag{6}$$

The synthesis method of control is shown below in Figure 2.

The user must look for the smallest difference between the output response signal and the input signal, which relates to the control error z_∞, formulated as: $z_\infty = w - z$, where: w – input signal; z – output response signal.

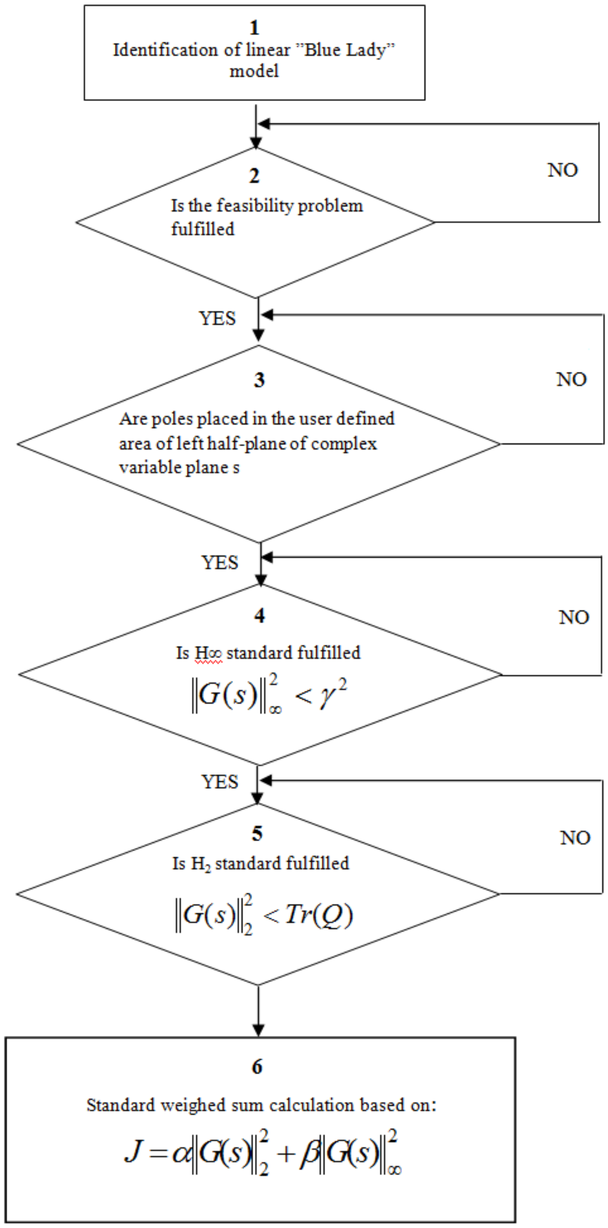

Figure 2. Synthesis method of control using LMI graph

The fourth step is the calculation of the H_2 standard. Positively symmetrical matrices P and $Q = Q^T$ must be found that fulfil the condition below:

$$\begin{bmatrix} Q & C \\ C^T & P \end{bmatrix} \succ 0 \tag{7}$$

The user must empirically determine the value of the γ_2 coefficient needed to minimize the value of the control signal. During simulations, an inequality must be fulfilled for the γ_2 coefficient as shown below:

$$\|G(s)\|_2^2 < \gamma^2, Tr(Q) < \gamma^2 \tag{8}$$

It means that the trace of matrix Q cannot exceed the upper boundary γ_2 of the H_2 standard.

The final step is the minimization of the above defined standards. The minimization of the H_2

standard, which is $\|G(s)\|_2$ for transmittance $G(s)_2$, and the H_∞ standard, which is $\|G(s)\|_\infty$ for transmittance $G(s)_\infty$, allows a weighed sum of both standards in the form of the J coefficient described below to be formulated:

$$J = \alpha \|H\|_2^2 + \beta \|H\|_\infty^2 \tag{9}$$

where coefficients $\alpha > 0$ and $\beta > 0$ are empirically determined by the user. In this paper, it was agreed that $\alpha = 1$, $\beta = 1$, which might cause the results to be more conservative.

After all the steps have been performed controller synthesis is finished. The above steps belong to a general scheme of controller set point selection for the designed controller. Full controller synthesis depends on:

- Controller to controlled object configuration, which impacts controlled object state equalities;
- Taking into consideration the input and output signals of the system.

Method of controller synthesis using model predictive control

Predictive control is an advanced optimal control strategy created in 1987 by Clarke and others (Clarke, Mohtadi & Tuffs, 1987a; 1987b). It is based on finding the values of future control signals that will minimise the criteria function given below:

$$J = \sum_{n=N_1}^{N_2} \delta(n)\left[y(n+k) - w(n+k)\right]^2 + \sum_{n=1}^{N_2} \lambda(n)\left[\Delta u(n+k-1)\right]^2 \tag{10}$$

where:

$\delta(n)$, $\lambda(n)$ – input and output signal weight increase vectors;

$y(n+k)$ – predicted output signal values for $(n+k)$ time step;

$w(n+k)$ – predicted reference trajectory for $(n+k)$ time step;

$\Delta u(n+k)$ – predicted control signal deviations for $(n+k)$ time step.

The above quality control criteria, which are minimised in every time step of the algorithm operation, can be modified according to the requirements of the controlled object. In case of MPC controllers, there is a possibility to control both mono and multivariable objects. With multivariable objects, predicted values are converted to matrices of predicted values according to the scheme below:

$$\delta(n) \to \Delta(n), \lambda(n) \to \Lambda(n), y(n+k) \to \to Y(n+k), w(n+k) \to W(n+k), \Delta u(n+k) \to \Delta U(n+k)$$

Controller synthesis using MPC requires several steps which are described below and shown in Figure 3:

1. Controller structure selection – defining control, output and reference signals.
2. Creating a linear controlled object model or linearization of a nonlinear object model close to the working point – which in case of nonlinear models will have the same structure as the controller.
3. Selecting boundaries of input signals and change speed of control signals.
4. Selecting prediction and control horizon lengths.
5. Selecting the control signal weight increase and output signal weights.

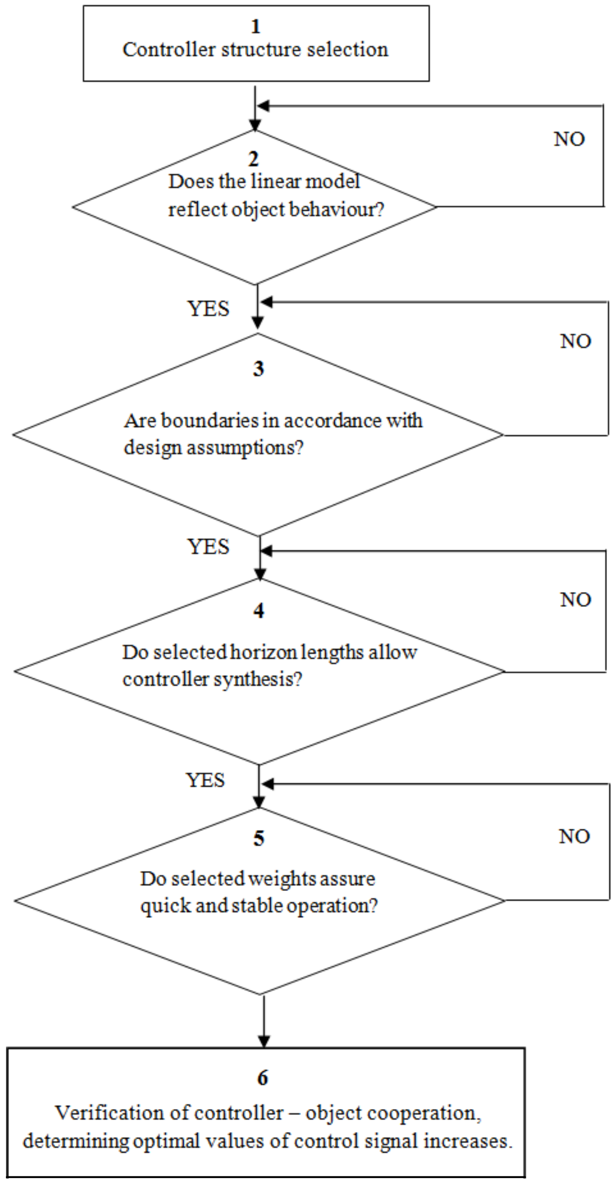

Figure 3. Synthesis method of controller using MPC graph

In case of a predictive controller, as in all controller design, it is necessary to precisely define its tasks. This is done by determining control signal vectors $u(n)$, output signal vectors $y(n)$ and discrete state space variables $x(n)$. Vector elements are discrete because MPC controllers are discrete. Additionally, in the case of nonlinear controlled objects it is necessary to determine a working point in the vicinity of which the controller will work. It means that the mathematical model of a nonlinear controlled object must be linearized.

The predictive controller algorithm is based on the minimization of the target function (equation number J=...) in the space of the known prediction and control horizons. To estimate the output signals $y_{est}(n + k)$ required to achieve the target function value and to perform the optimization to determine control signal value vector (matrix), it is necessary to use the controlled object model that is shown in Figure 3.

Due to placing the model inside the controller block, it is necessary to adjust its structure to the controller tasks. Input signals must have the same form as control signals ($u(n)$) determined in a closed loop control system, while the object output signals should have the same form as the estimated output signals ($y_{est}(n)$). In order to assure quick calculations and to find a global optimal solution in real life (industrial) applications, a linear (or linearized in the vicinity of the working point) model is used.

The creation or linearization of a controlled object model is a complex process, in which it is possible to use different methods. The most popular and most frequently used is model identification in state space (Brunton, Dawson & Rowley, 2014), which is described by the equations below:

$$\begin{aligned} \dot{x}(n) &= A \cdot x(n) + B \cdot u(n) \\ y(n) &= C \cdot x(n) \end{aligned} \tag{11}$$

where: A, B, C – state, control and output matrices, $x(n)$ – state vector, $y(n)$ – output vector, $u(n)$ – control vector in discrete time steps n.

Based on the above, a discrete model in state space is achieved. Before it can be included into the controller structure, model verification is necessary. Precisely how the model predicts output signals $y_{est}(n)$ based on future control signals $u(n - k)$, output signals $y(n - k)$ and current control signals $u(n)$ must be checked. Additionally, model stability must be checked, because only a stable predictive model allows synthesis of a stable controller. Finally, a residual analysis is done and a check carried out as to whether the model will work correctly regardless of input and output signal combinations (so-called model independence). A model that fulfils the above conditions can be the basis for further predictive controller synthesis.

Predictive control, being an advanced variant of optimal control, gives the possibility to include boundaries for control signal speed change ($\Delta u(n) = u(n + 1) - u(n)$) as well as control signal ($u(n)$) and output signal ($y(n)$) values. As such, it is a case of square optimization with boundaries. According to (Kerrigan & Maciejowski, 1999) a minimum of the function must be found (10) with the boundaries below:

$$\begin{aligned} \Delta u(n+k) &\in [V_{\min}; V_{\max}] \\ u(n+k) &\in [U_{\min}; U_{\max}] \\ y(n+k) &\in [Y_{\min}; Y_{\max}] \end{aligned} \tag{12}$$

where: $V_{min;max}$, $U_{min;max}$, $Y_{min;max}$ – numerical values of boundaries.

The most frequently used method of square optimization is the Newton method, which allows for a quick solution. This method can be used because the boundaries of a MPC controller are from the top and bottom.

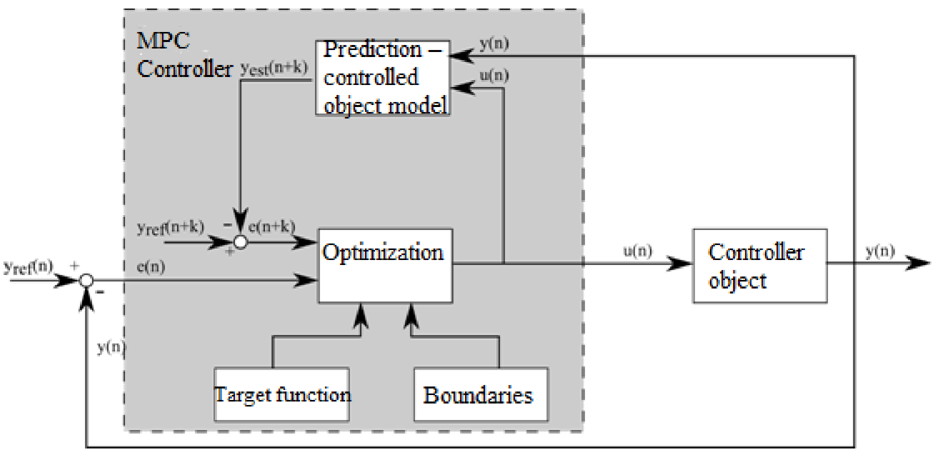

Figure 4. General structure of MPC controller

In predictive control two time frames are considered: the prediction horizon and the control horizon. The prediction horizon is a time frame from the present time (n) to time step ($n + k$) in which the final value of output signal $y_{est}(n + k)$ is estimated. The control horizon, on the other hand, is a time frame in which the algorithm takes into account different values of predicted output signals $u(n)$. In the time between the end of control horizon and the end of prediction horizon, to minimise the required calculations, it is assumed that the control signal is constant and equal to the last value of the control signal in the control horizon, as shown in Figure 5.

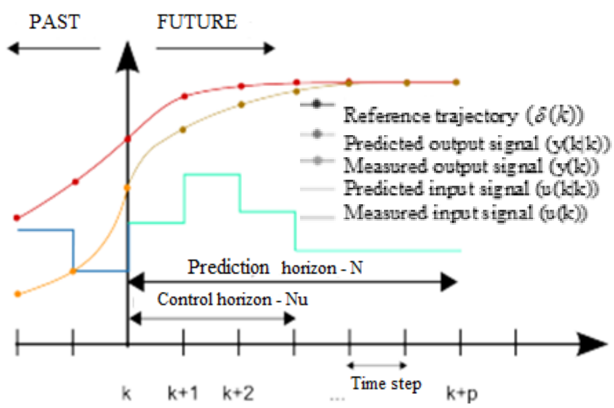

Figure 5. Predictive controller concept

The selected lengths of both horizons impact the stability, robustness to external interference and the working speed of the closed loop control system. It is found that the longer the prediction horizon, the better the controller performance. It decreases oscillations in output signal (Kerrigan & Maciejowski, 1999), but significantly slows down controller operation due to the increased size of the vectors and matrices used in the target function. In the predictive control of objects with large inertia and slow dynamics, the prediction horizon needs to be increased to include non-minimum phase behaviour of the controlled object. In the control of nonlinear objects with large inertia, such as ships, the prediction horizon needs to be longer than the time required to achieve turning at the maximum swing of the thruster (e.g. rudder) and with the longitudinal speed matching the object working point. Shortening the prediction horizon significantly increases robustness to object parameter changes and external interference (Miller, 2014). Thus the prediction horizon should be as short as possible, but long enough to allow stable controller operation.

The control horizon length should match the prediction horizon length. Its value is determined by iterations to be as small as possible, but long enough to allow stable and quick controller operation. The pole placement needs to be checked for synthesized system transmittance and evaluated if they all fit inside a singular circle. While performing simulations of closed loop control systems, attention must be paid if there are no oscillations in output signal and if the system is not brought to stability limit after some time of operation, which is frequently the case with incorrectly selected horizon lengths.

The final parameters that need to be selected in controller synthesis are weights $\delta(n)$, $\lambda(n)$ (10). Correct weight selection has a significant influence on control performance and system stability. The weight coefficient $\lambda(n)$ determines the penalty for control signal change, smoothes it out and minimises its step changes. In case its value drops to 0, boundaries to control signals are not considered anymore. Increasing this coefficient decreases overshoot and increases the response time. The function of coefficient $\delta(n)$ determines the penalty for control error between the output signal and setpoint value and ensures that the boundaries of the output signals are not violated. The larger the value of this coefficient, the smaller the probability of offset occurrence. When selecting values of $\delta(n)$ and $\lambda(n)$ coefficients, values ensuring closed loop system stability must be selected, and in the next step they must be modified by iterations and their influence on system operation observed.

In multidimensional systems, instead of single coefficients, coefficient matrices are used for input signals $\Delta(n)$ and control signals $\Lambda(n)$ that have the form below:

$$\Delta(n)=\begin{bmatrix}\delta_1 & 0 & \cdots & 0\\ 0 & \delta_2 & \cdots & 0\\ \vdots & 0 & \ddots & 0\\ 0 & \cdots & 0 & \delta_l\end{bmatrix};\Lambda(n)=\begin{bmatrix}\lambda_1 & 0 & \cdots & 0\\ 0 & \lambda_2 & \cdots & 0\\ \vdots & 0 & \ddots & 0\\ 0 & \cdots & 0 & \lambda_m\end{bmatrix} \quad (13)$$

These are an array of coefficients for specific channels and perform the exact same functions as the $\delta(n)$ and $\lambda(n)$ coefficients in monovariable systems. Both $\Delta(n)$ and $\Lambda(n)$ are diagonal matrices and must be positively definite. It is possible to observe significant variances between specific matrix elements, which are caused by differences in the significance of specific controlled variables as well as disproportions between output signal values and the range of their changes. For example, if the system response to the first input signal is significantly slower than that to other input signals, then the value of the δ_1 coefficient should be increased compared to the other ones (Kerrigan & Maciejowski, 1999; Miller, 2014).

The above method is a general scheme for predictive controller synthesis and simulating its operation in a closed loop system. For system operation, verification closed loop stability must be checked along with proper system operation.

Conclusions

The main goal of this paper was to describe steps of controller synthesis using LMI and MPC methods.

The LMI method, shown in Figure 1, includes four groups of conditions that have to be met in order to correctly specify the optimization goal. This includes the correct identification of the linear controlled object model which is a basis for the state equations formulation. The following stages are related to the control and measurement non-singularity.

Predictive controller synthesis needs proper linear model identification, with a structure which is in accordance with the control and output signals of the future controller. A stable model, which adequately projects object behaviour, is essential for producing a stable predictive control system. Stages of predictive controller synthesis, include iteration and the empiric determination of the remaining controller parameters, need to be followed. This means that MPC controller synthesis is based on multiple modifications of parameters and the verification of system operation by simulations, as with LMI controller synthesis. Another common feature of both methods, apart from being based on a linear controlled object model, is the need to solve the feasibility problem.

Predictive controller synthesis is based on the knowledge and experience of the designer. There are however tools that assist this task. One of these is the model predictive toolbox for Matlab, which allows for the determination of the controller matrix in state space, based on declared parameters, and for the checking of the correctness of controller operation with freely selected criteria functions. Using this tool makes the synthesis process a lot easier for the designer, because it automates mathematical calculations using predefined functions and speeds up and simplifies the iteration parameter adjustment of MPC.

Predictive controllers synthesized with the method described above are stable, and behave robustly when slight changes of controlled object parameters and interference (e.g. wind) occur. An additional benefit is the possibility to include control and output signal boundaries and control signal change speed directly in the controller algorithm, which is not the case for LMI controllers.

Both MPC and LMI controllers are based on optimization methods, but they are included in different stages of controller design. In case of LMI controllers, linear optimization is used during controller synthesis. Then, during controller operation, earlier synthesized systems with fixed parameters are used. In MPC controllers, however, nonlinear (square) optimization with boundaries is used, which is executed in every step of the algorithm operation. This means that a controller using a predictive algorithm requires a lot more calculation capacity than an LMI controller. Furthermore, an MPC controller for the same controlled object has a significantly larger dimension than an LMI controller. However, this means that the possibilities to fine-tune the MPC controller to specific system requirements are larger than with an LMI controller.

References

1. Boyd, S., Ghaoui, L., Feron, E. & Balakrishnan, V. (1994) *Linear Matrix Inequalities in System and Control Theory*. Philadelhia: Society for Industrial and Applied Mathematics (SIAM).
2. Brunton, S.L., Dawson, S.T.M. & Rowley, C.W. (2014) State space model identification and feedback control of unsteady aerodynamic forces. *Journal of Fluids and Structures*. 50. pp. 253–270.
3. Clarke, D.W., Mohtadi, C. & Tuffs, P.S. (1987a) Generalized predictive controlpart i. the basic algorithm. *Automatica*. 23, 2. pp. 137–148.
4. Clarke, D.W., Mohtadi, C. & Tuffs, P.S. (1987b) Generalized predictive controlpart i extensions and interpretations. *Automatica*. 23, 2. pp. 149–160.
5. Kerrigan, E.C. & Maciejowski, J.M. (1999) Fault-tolerant control of a ship propulsion system using model predictive control. *Proceedings of European control conference*.
6. Koziński, W. (2004) *Projektowanie regulatorów. Wybrane metody klasyczne i optymalizacyjne*. Warszawa: Oficyna Wydawnicza Politechniki Warszawskiej.
7. Miller, A. (2014) Model Predictive Control of the ship's motion in presence of wind disturbances. *Scientific Journals of the Maritime University of Szczecin*. 39 (111). pp. 107–115.
8. Paszke, W. (2005) *Analysis and synthesis of multidimensional system classes using linear matrix inequality methods*. Praca doktorska. Zielona Góra.
9. Rybczak, M. (2014) Bounded linear stability region in linear matrix inequalities for a multivariable object. Scientific Journals of the Maritime University of Szczecin. 39(111). pp. 133–139.
10. Weiland, S. & Scherer, C. (2005) *Linear Matrix Inequalities in Control*. The Netherlands: Society for Industrial and Applied Mathematics. Delft Center for Systems and Control Delft University of Technology.
11. Xi, Y.G., Li, D.W. & Lin, S. (2013) Model predictive control – status and challenges. *Acta Automatica Sinica*. 39, 3.