Chapter 1
Introduction to Model Predictive Control

Please cite the book:
Maciej Ławryńczuk: Nonlinear Predictive Control Using Wiener Models:
Computationally Efficient Approaches for Polynomial and Neural Structures.
Studies in Systems, Decision and Control, vol. 389, Springer, Cham, 2022.

Abstract This Chapter is an introduction to the field of MPC. Its basic idea and the rudimentary MPC optimisation problems are defined, at first for Single-Input Single-Output (SISO) processes and next for Multiple-Input Multiple-Output (MIMO) ones. A method to cope with infeasibility problems caused by constraints imposed on the predicted controlled variables is presented. Next, parameterisation of the decision variables using Laguerre functions in order to reduce the number of actually optimised variables is described. Classification of MPC algorithms is given and computational complexity issues are discussed. Finally, some example applications of MPC algorithms in different fields are reported.

1.1 Formulation of the Basic MPC Problem

The objective of *a good control algorithm* is to calculate repeatedly on-line the value of the manipulated variable (or the values of many manipulated variables) that leads to *good process behaviour* [36]. Let us discuss the term *good process behaviour* using two examples.

The first process example is a residential building equipped with an underfloor radiant heating system based on electric heating foils [99]. From the point of view of control engineering, the process is very simple since it has only one manipulated variable (process input) which is the value of the current (or the voltage) applied to the foils and only one controlled variable (process output) which is the average temperature inside the building. There are two objectives of the controller:

a) it must increase the temperature quickly when the user increases the temperature set-point, i.e. the value of the required temperature,
b) it must stabilise the temperature when the outside temperature drops.

The first objective is set-point tracking, i.e. the process output must follow changes of its set-point. The second objective is compensation of disturbances, i.e. the process output must be (approximately) constant when the process is affected by external disturbances, also called uncontrolled process inputs. In our simple example, it is only possible to increase the temperature by increasing the current (or the voltage), but it is impossible to reduce the temperature. It means that it works fine in the two above situations, but when the user wants to reduce the set-point or the outside temperature increases, the only possible action is to reduce heating, switch it off or simply ventilate the building. Of course, in more advanced solutions, it is possible to both heat and cool. Furthermore, it may be necessary to stabilise not only temperature but also humidity. An important application of such a control system may be found in greenhouses, where it is necessary to maintain constant temperature and humidity values for the proper growth of plants. Different parts of the greenhouse may be heated separately to obtain different local temperature conditions. In such a case, there are many manipulated, controlled and disturbance variables. In addition to set-point tracking and compensation of disturbances, the calculated values of the manipulated signals must satisfy some constraints. Typically, they have limited values and rates of change caused by the physical limits of actuators. Moreover, one may imagine that some constraints are imposed on the controlled variables, e.g. temperature and humidity should be in some ranges.

The second process example is a car. Its control is significantly much more complicated than the simple temperature control task discussed above. It is because a driver must manipulate numerous variables, such as the accelerator, clutch and brake pedals, the wheel and the gear lever. There are many controlled variables, such as position on the road, speed, acceleration. The driver controls the car in such a way that position, speed and acceleration set-point trajectories are followed. Moreover, the influence of many external disturbances is compensated, e.g. variable road slope, type of surface, side wind. Unlike the first process example, the driver not only controls the process but also calculates the set-point trajectories on-line, i.e. adjusts them to the current road conditions. Of course, there are numerous constraints which must be taken into account during calculation of the values of the manipulated variables and adjusting the trajectories. Both manipulated and controlled variables must be constrained in this example.

The classical Proportional-Integral-Derivative (PID) controller in continuous-time domain is described by the following rule

$$u(t) = u_0 + K\left(e(t) + \frac{1}{T_{\mathrm{i}}}\int_0^t e(\tau)\mathrm{d}\tau + T_{\mathrm{d}}\frac{\mathrm{d}e(t)}{\mathrm{d}t}\right) \tag{1.1}$$

The control error is defined as the difference between the set-point and the current measured value of the controlled variable, i.e. $e(t) = y^{\mathrm{sp}}(t) - y(t)$. The value of the manipulated variable u for the current time t is a linear function of three parts:

the proportional part, which takes into account the current control error, $e(t)$, the integral part, which takes into account the past errors, and the derivative part, which takes into account the rate of change of the error. The tuning parameters are: the proportional gain K, the integration time-constant T_{i} and the derivative time-constant T_{d}. Using Euler's backward differentiation and trapezoidal integration, in discrete-time domain, the value of the manipulated variable for the current sampling instant k is

$$u(k) = u(k-1) + r_0e(k) + r_1e(k-1) + r_2e(k-2) \tag{1.2}$$

where $e(k)$, $e(k-1)$ and $e(k-2)$ denote the values of the control error at the sampling instants k, $k-1$ and $k-2$, respectively, $u(k-1)$ is the value of the manipulated variable at the sampling instant $k-1$, r_0, r_1, r_2 are parameters. They are calculated for the settings K, T_{i}, T_{d} and the chosen sampling time of the controller. If properties of the process are (approximately) linear, the PID controller proves to be very efficient in numerous applications. Nevertheless, the PID controller has the following limitations:

1. The PID control law (1.1) or (1.2) is linear. In the case of nonlinear processes, the possible quality of control may be not satisfactory, in particular when the set-point changes are significant and fast or the external disturbances are strong.
2. The PID controller works fine when the process delay is not significant. Conversely, PID control of delayed dynamical systems is usually not good.
3. In its basic version, the PID controller does not include constraints. Although simple limiters may easily enforce limits of the manipulated variable and constraints on its rate of change, there is no systematic way to enforce satisfaction of constraints imposed on the controlled variable.
4. The PID controller is a natural choice when the controlled process has one manipulated variable and one controlled one. In the case of a dynamical process with many inputs and many outputs, the basic problem is finding out which manipulated variable has the strongest influence on each controlled one. Next, several classical single-loop PID controllers are used. Such an approach works correctly when the consecutive manipulated variables strongly impact the consecutive controlled ones, but when one process input impacts two or more outputs, such a control structure does not work. Moreover, the number of process inputs and outputs must be equal.
5. It is interesting that the current value of the manipulated variable generated by the PID controller depends on the current and past errors. It is clear when we consider the discrete-time implementation (1.2). The derivative part tries to use some information of the future control error but using only its current and previous measurements.
6. The PID controller is tuned in practice using some simple rules, e.g. the famous Ziegler and Nichols procedure, or simply by the trial and error approach. Although interpretation of the continuous-time parameters K, T_{i} and T_{d} is straightforward, the parameters r_0, r_1 and r_2 of the discrete-time controller have no physical interpretation.

Having discussed the objectives of a good control algorithm and properties of the PID structure, we will discuss the basic formulation of MPC. Let us recall the problem of controlling a car by a driver. Humans do not use mathematical equations to calculate values of the manipulated variables. Conversely, in our mind, we repeatedly do the following:

1. We collect all possible information, i.e. we observe the road and monitor the car dashboard.
2. Using *a model* of the car, i.e. knowing how the car reacts, we *predict* behaviour of the car, i.e. its position, speed, acceleration, over some *time horizon*.
3. We *optimise* behaviour of the car, i.e. we find out how the car should be controlled in order to satisfy all control objectives. We find not only the current values of the manipulated variables, but we also assess their future values.
4. Prediction of the future car state, as well as optimisation of the current and future control actions, are coupled, i.e. we have many possible control policies, we assess how they are successful and we choose the best one.
5. We constantly repeat the above steps as we receive new information, we assess the results of our actions and how the disturbances change. The traffic and road conditions are never constant. The horizon is moved each time we start prediction and optimisation.

Fig. 1.1 illustrates the above. Let us consider information collected by the driver A (measurements) and the decisions taken for three different time instants denoted as t_1, t_2 and t_3, respectively. Initially, at time t_1, for the prediction horizon used, the driver A is able to see the speed limit sign and his or her decision is to deduce speed to 50 km/h. The prediction horizon is too short to notice the cars B and C. Next, at time t_2, the prediction horizon makes it possible to notice the car B that is approaching from the right-side road. Because the car B moves very slowly, the driver A decides to continue driving with constant speed, he or she does not wait for the car B to give way to it. For the prediction horizon used, the driver A does not notice the car C. Finally, at time t_3, the driver A sees the car C. He or she is unable to overtake it since the car D moves from the opposite direction, in the second lane. Probably, provided that no other obstacles exist, overtaking will be possible shortly. Let us point out that all decisions are made using predictions of future behaviour of all drivers, possible drivers' actions are predicted using some models, all existing constraints are taken into account.

Now, it is time to formulate the basic MPC problem using mathematics. At first, let us consider a SISO process. The input of the controlled process is denoted by u, the output is denoted by y. At each consecutive sampling instant k, $k = 1, 2, 3, \ldots$, the vector of the future increments of the manipulated variable

$$\triangle \boldsymbol{u}(k) = \begin{bmatrix} \triangle u(k|k) \\ \vdots \\ \triangle u(k + N_{\mathrm{u}} - 1|k) \end{bmatrix} \tag{1.3}$$

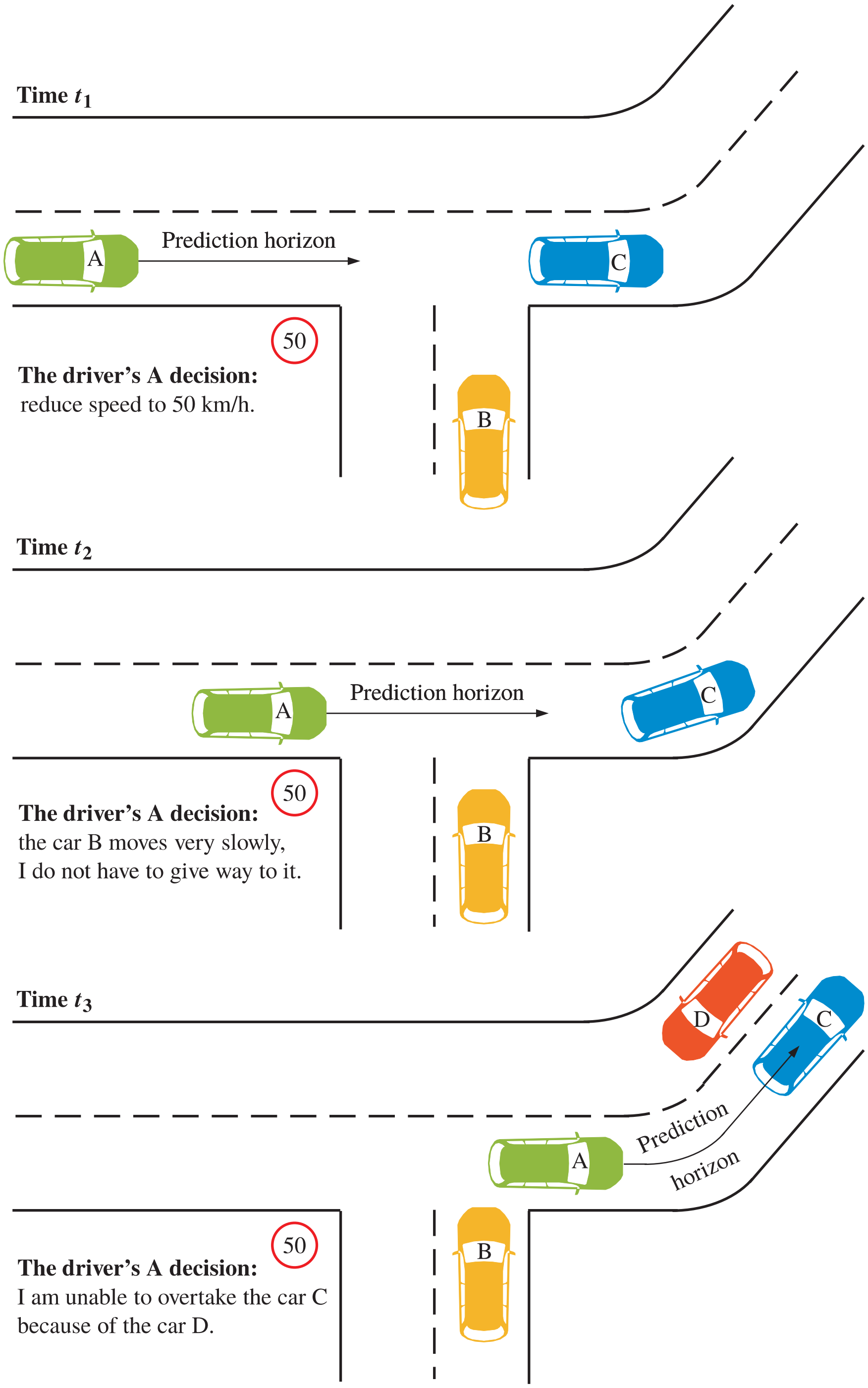

Fig. 1.1 Situations on the road and the driver's A decisions for three example time instants t_1, t_2, t_3

is calculated on-line. The symbol $\triangle u(k+p|k)$ denotes the increment of the manipulated variable for the sampling instant $k+p$ calculated at the current sampling instant k, N_{u} is the control horizon which defines the number of decision variables (1.3). The first increment is

$$\triangle u(k|k) = u(k|k) - u(k-1) \tag{1.4}$$

and the following ones are

$$\triangle u(k+p|k) = u(k+p|k) - u(k+p-1|k) \tag{1.5}$$

for $p = 1, \ldots, N_{\mathrm{u}} - 1$. The symbol $u(k+p|k)$ denotes the value of the manipulated variable for the sampling instant $k+p$ calculated at the current sampling instant k, $u(k-1)$ is the value of the manipulated variable used (applied to the process) at the previous sampling instant. In the simplest case, the vector of decision variables (1.3) is calculated on-line from an unconstrained optimisation problem

$$\min_{\triangle \boldsymbol{u}(k)} \{J(k)\} \tag{1.6}$$

Typically, the minimised objective function (the cost-function) consists of two parts

$$J(k) = \sum_{p=1}^{N} (y^{\mathrm{sp}}(k+p|k) - \hat{y}(k+p|k))^2 + \lambda \sum_{p=0}^{N_{\mathrm{u}}-1} (\triangle u(k+p|k))^2 \tag{1.7}$$

The first part of the MPC cost-function measures the predicted quality of control since the differences between the set-point trajectory and the predicted trajectory of the output variable (i.e. the predicted control errors) over the prediction horizon $N \geq N_{\mathrm{u}}$ are taken into account. The set-point value for the sampling instant $k+p$ known at the current sampling instant k is denoted by $y^{\mathrm{sp}}(k+p|k)$, the predicted value of the output variable for the sampling instant $k+p$ calculated at the current instant is denoted by $\hat{y}(k+p|k)$. The future values of the set-point are usually not known, hence only the scalar set-point value for the current sampling instant, denoted by $y^{\mathrm{sp}}(k)$, is used, i.e. $y^{\mathrm{sp}}(k+1|k) = \ldots = y^{\mathrm{sp}}(k+N|k) = y^{\mathrm{sp}}(k)$. Such an approach is typically used in control of industrial processes in which changes of the set-point are very rare, but the controller must compensate for changes of the disturbances. However, in some applications, e.g. in autonomous vehicles and robotics, the set-point trajectory may be not constant over the prediction horizon. The second part of the MPC cost-function is a penalty term. It is used to reduce excessive changes of the manipulated variable; $\lambda > 0$ is a weighting coefficient. The greater its value, the lower the increments of the manipulated variable and, hence, the slower control. Because in practice the control horizon is shorter than the prediction one, it is assumed that $u(k+p|k) = u(k+N_{\mathrm{u}}-1|k)$ for $p = N_{\mathrm{u}}, \ldots, N$, which means that $\triangle u(k+N_{\mathrm{u}}|k) = \ldots = \triangle u(k+N|k) = 0$.

Although at each sampling instant as many as N_{u} future increments of the manipulated variable (1.3) are calculated, only the first element of this sequence is actually applied to the process, i.e. the increment for the current sampling instant k.

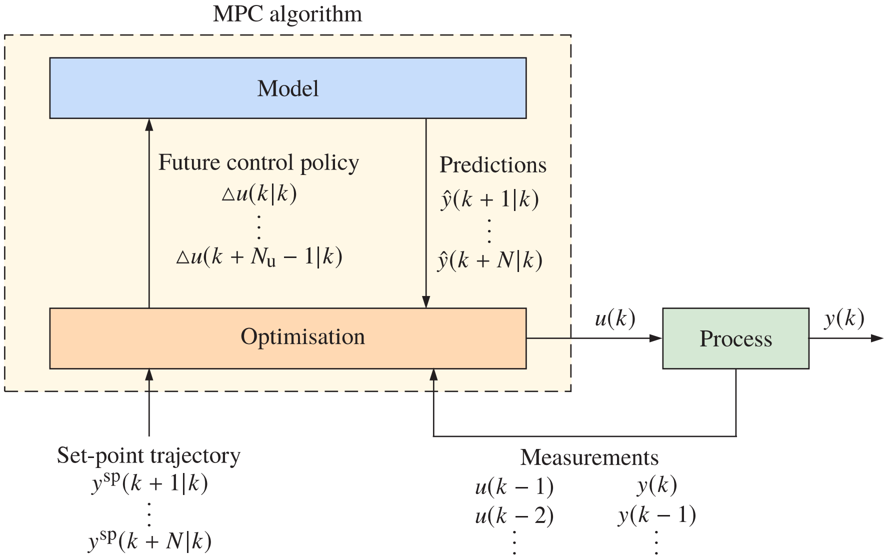

Fig. 1.2 The general structure of the MPC algorithm

Let the optimal vector calculated from the MPC optimisation problem be denoted by $\triangle \boldsymbol{u}^{\mathrm{opt}}(k)$. The current optimal value of the manipulated variable is applied to the process

$$u(k) = \triangle u^{\mathrm{opt}}(k|k) + u(k-1) \tag{1.8}$$

where $\triangle u^{\mathrm{opt}}(k|k)$ is the first element of the vector $\triangle \boldsymbol{u}^{\mathrm{opt}}(k)$. In the next sampling instant $(k+1)$ the output value of the process is measured (the state variables may also be measured or estimated), the prediction horizon is shifted one step forward and the whole procedure described above is repeated. As a result, the MPC algorithm works in the closed-loop, i.e. with feedback from the measured process output. Fig. 1.2 depicts the general structure of the MPC algorithm. It is assumed that the time necessary to solve the MPC optimisation problem is much shorter than the sampling time.

In practical applications, it is necessary to take into account existing constraints. First of all, the magnitude of the manipulated variable may be constrained. Such constraints result from the physical limits of the actuator

$$u^{\min} \leq u(k+p|k) \leq u^{\max},\ p = 0, \ldots, N_{\mathrm{u}}-1 \tag{1.9}$$

where $u^{\min}$ and $u^{\max}$ are the minimal and maximal values of the manipulated variable, respectively. It is interesting to notice the fact that all calculated values of the manipulated variable over the whole control horizon are limited, not only the value for the current sampling instant, i.e. $u(k|k)$. Secondly, the rate of change of the manipulated variable may be constrained

$$\triangle u^{\min} \leq \triangle u(k+p|k) \leq \triangle u^{\max},\ p=0,\ldots,N_{\mathrm{u}}-1 \tag{1.10}$$

where $\triangle u^{\min}$ and $\triangle u^{\max}$ are the maximal negative and maximal (positive) changes of the manipulated variable, respectively (usually $\triangle u^{\min} = -\triangle u^{\max}$). All calculated increments of the manipulated variable over the whole control horizon are limited, not only the increment for the current sampling instant, i.e. $\triangle u(k|k)$. Thirdly, the predicted values of the process output variable may also be limited, which is usually enforced by some technological reasons

$$y^{\min} \leq \hat{y}(k+p|k) \leq y^{\max},\ p=1,\ldots,N \tag{1.11}$$

where $y^{\min}$ and $y^{\max}$ are the minimal and maximal values of the predicted output variable, respectively. All predictions over the prediction horizon N are constrained. When the constraints are present, the vector of decision variables (1.3) is calculated at each sampling instant from an optimisation problem in which the cost-function (1.7) is minimised and all the constraints (1.9), (1.10) and (1.11) are taken into account. Hence, the rudimentary MPC constrained optimisation problem is

$$\begin{aligned}
&\min_{\triangle \boldsymbol{u}(k)} \left\{ J(k) = \sum_{p=1}^{N} \left(y^{\mathrm{sp}}(k+p|k) - \hat{y}(k+p|k)\right)^2 + \lambda \sum_{p=0}^{N_{\mathrm{u}}-1} \left(\triangle u(k+p|k)\right)^2 \right\}\\
&\text{subject to} \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (1.12)\\
&u^{\min} \leq u(k+p|k) \leq u^{\max},\ p=0,\ldots,N_{\mathrm{u}}-1\\
&\triangle u^{\min} \leq \triangle u(k+p|k) \leq \triangle u^{\max},\ p=0,\ldots,N_{\mathrm{u}}-1\\
&y^{\min} \leq \hat{y}(k+p|k) \leq y^{\max},\ p=1,\ldots,N
\end{aligned}$$

The number of decision variables of the optimisation problem (1.12) is N_{u}, the number of constraints is $4N_{\mathrm{u}} + 2N$.

All things considered, in the case of the SISO constrained MPC algorithm, at each sampling instant k, the following steps are performed on-line:

1. The current value of the controlled variable, $y(k)$, is measured; the state variables may be measured or estimated when necessary.
2. The future sequence of increments of the manipulated variable is calculated from the optimisation problem (1.12).
3. The first element of the determined sequence is applied to the process (Eq. (1.8)).

Having discussed the MPC formulation for the SISO case, we will concentrate on a more general MIMO problem. Let us assume that the number of process inputs is denoted by n_{u} and the number of process outputs is denoted by n_{y}. In this book we use two notation methods: scalars and vectors. When possible, it is very convenient to use vectors, but sometimes the consecutive scalar signals must be used. The vector of manipulated variables is $u = \begin{bmatrix} u_1 & \ldots & u_{n_{\mathrm{u}}} \end{bmatrix}^{\mathrm{T}}$ and the vector of controlled variables is $y = \begin{bmatrix} y_1 & \ldots & y_{n_{\mathrm{y}}} \end{bmatrix}^{\mathrm{T}}$. The vector of decision variables of the MPC algorithm (1.3) is hence of length $n_\mathrm{u}N_\mathrm{u}$. The minimised MPC cost-function for the MIMO case is

$$\begin{aligned} J(k) = & \sum_{p=1}^{N} \sum_{m=1}^{n_\mathrm{y}} \mu_{p,m} \left(y_m^\mathrm{sp}(k+p|k) - \hat{y}_m(k+p|k) \right)^2 \\ & + \sum_{p=0}^{N_\mathrm{u}-1} \sum_{n=1}^{n_\mathrm{u}} \lambda_{p,n} \left(\triangle u_n(k+p|k) \right)^2 \end{aligned} \tag{1.13}$$

In comparison with the SISO case (Eq. (1.7)), in the first part of the cost-function (1.13), we consider the predicted control errors for all n_y controlled variables over the whole prediction horizon. Similarly, in the second part of the cost-function, increments of all n_u manipulated variables are taken into account over the whole control horizon. The weighting coefficients $\mu_{p,m} \geq 0$ make it possible to differentiate the influence of the predicted control errors of the consecutive outputs within the prediction horizon. The coefficients $\lambda_{p,n} > 0$ are used not only to differentiate the influence of the control increments of the consecutive inputs of the process within the control horizon but to establish the necessary scale between both parts of the cost-function.

The MPC cost-function and the resulting optimisation problems may be conveniently and compactly derived, formulated and implemented using vector-matrix notation rather than scalars. The cost-function (1.13) may be expressed in the following form

$$J(k) = \sum_{p=1}^{N} \left\| y^\mathrm{sp}(k+p|k) - \hat{y}(k+p|k) \right\|_{\boldsymbol{M}_p}^2 + \sum_{p=0}^{N_\mathrm{u}-1} \left\| \triangle u(k+p|k) \right\|_{\boldsymbol{\Lambda}_p}^2 \tag{1.14}$$

Now, the set-point vector for the sampling instant $k+p$ known at the current sampling instant k is denoted by $y^\mathrm{sp}(k+p|k)$, the predicted vector of the output variables for the sampling instant $k+p$ calculated at the current sampling instant k is denoted by $\hat{y}(k+p|k)$, both vectors are of length n_y. The matrix $\boldsymbol{M}_p = \mathrm{diag}(\mu_{p,1}, \ldots, \mu_{p,n_\mathrm{y}}) \geq 0$ is of dimensionality $n_\mathrm{y} \times n_\mathrm{y}$, the matrix $\boldsymbol{\Lambda}_p = \mathrm{diag}(\lambda_{p,1}, \ldots, \lambda_{p,n_\mathrm{u}}) > 0$ is of dimensionality $n_\mathrm{u} \times n_\mathrm{u}$.

For the process with n_u manipulated variables, the magnitude constraints are

$$u_n^\mathrm{min} \leq u_n(k+p|k) \leq u_n^\mathrm{max}, \ p = 0, \ldots, N_\mathrm{u}-1, \ n = 1, \ldots, n_\mathrm{u} \tag{1.15}$$

where u_n^min and u_n^max are the minimal and maximal values of the manipulated variable u_n, respectively. The constraints imposed on the rate of change of the manipulated variables are

$$\triangle u_n^\mathrm{min} \leq \triangle u_n(k+p|k) \leq \triangle u_n^\mathrm{max}, \ p = 0, \ldots, N_\mathrm{u}-1, \ n = 1, \ldots, n_\mathrm{u} \tag{1.16}$$

where $\triangle u_n^\mathrm{min}$ and $\triangle u_n^\mathrm{max}$ are the maximal negative and maximal (positive) changes of the manipulated variable u_n, respectively. The constraints imposed on the predicted values of the process output variables are

$$y_m^{\min} \leq \hat{y}_m(k+p|k) \leq y_m^{\max},\ p = 1,\ldots,N,\ m = 1,\ldots,n_{\mathrm{y}} \tag{1.17}$$

where $y_m^{\min}$ and $y_m^{\max}$ are the minimal and maximal values of the predicted variable y_m, respectively. If we use the vector notation, the constraints are defined by the following vectors of length n_{u}

$$u^{\min} = \begin{bmatrix} u_1^{\min} \\ \vdots \\ u_{n_{\mathrm{u}}}^{\min} \end{bmatrix},\ u^{\max} = \begin{bmatrix} u_1^{\max} \\ \vdots \\ u_{n_{\mathrm{u}}}^{\max} \end{bmatrix},\ \triangle u^{\min} = \begin{bmatrix} \triangle u_1^{\min} \\ \vdots \\ \triangle u_{n_{\mathrm{u}}}^{\min} \end{bmatrix},\ \triangle u^{\max} = \begin{bmatrix} \triangle u_1^{\max} \\ \vdots \\ \triangle u_{n_{\mathrm{u}}}^{\max} \end{bmatrix} \tag{1.18}$$

and the following vectors of length n_{y}

$$y^{\min} = \begin{bmatrix} y_1^{\min} \\ \vdots \\ y_{n_{\mathrm{y}}}^{\min} \end{bmatrix},\ y^{\max} = \begin{bmatrix} y_1^{\max} \\ \vdots \\ y_{n_{\mathrm{y}}}^{\max} \end{bmatrix} \tag{1.19}$$

We may notice that the above 3 scalar constraints given by Eqs. (1.15), (1.16) and (1.17) may be rewritten in the same way it is done for the SISO case, i.e. by Eqs. (1.9), (1.10) and (1.11).

Now we may formulate the general MPC optimisation problem for MIMO processes. Using the cost-function (1.14), the scalar constraints (1.15), (1.16), (1.17) and the definitions (1.18)-(1.19), we have

$$\begin{aligned} &\min_{\triangle \boldsymbol{u}(k)} \left\{ J(k) = \sum_{p=1}^{N} \|y^{\mathrm{sp}}(k+p|k) - \hat{y}(k+p|k)\|_{\boldsymbol{M}_p}^2 + \sum_{p=0}^{N_{\mathrm{u}}-1} \|\triangle u(k+p|k)\|_{\boldsymbol{\Lambda}_p}^2 \right\} \\ &\text{subject to} \qquad\qquad (1.20) \\ &u^{\min} \leq u(k+p|k) \leq u^{\max},\ p = 0,\ldots,N_{\mathrm{u}}-1 \\ &\triangle u^{\min} \leq \triangle u(k+p|k) \leq \triangle u^{\max},\ p = 0,\ldots,N_{\mathrm{u}}-1 \\ &y^{\min} \leq \hat{y}(k+p|k) \leq y^{\max},\ p = 1,\ldots,N \end{aligned}$$

where the norm is defined as $\|\boldsymbol{x}\|_{\boldsymbol{A}}^2 = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x}$ (the matrix $\boldsymbol{A}$ is square). The above optimisation problem corresponds with the task (1.12) for the SISO case. The number of decision variables of the optimisation problem (1.20) is $n_{\mathrm{u}}N_{\mathrm{u}}$, the number of constraints is $4n_{\mathrm{u}}N_{\mathrm{u}} + 2n_{\mathrm{y}}N$.

Although at each sampling instant as many as $n_{\mathrm{u}}N_{\mathrm{u}}$ future increments of the manipulated variables (1.3) are calculated, only the first n_{u} elements of this sequence are actually applied to the process, i.e. the increments for the current sampling instant k. The current optimal values of the manipulated variables applied to the process are calculated from Eq. (1.8), the same which is used in the SISO case, but now all vectors, i.e. $u(k)$, $\triangle u^{\mathrm{opt}}(k|k)$ and $u(k-1)$, are of length n_{u}.

In the case of the MIMO constrained MPC algorithm, at each sampling instant k the following steps are performed on-line:

1. The current values of the controlled variables, $y_1(k), \ldots, y_{n_{\text{y}}}(k)$, are measured; the state variables may be measured or estimated when necessary.
2. The future sequence of increments of the manipulated variables is calculated from the optimisation problem (1.20).
3. The first n_{u} elements of the determined sequence are applied to the process (Eq. (1.8)).

Now, let us find a more compact representation of the rudimentary MIMO MPC optimisation problem (1.20). Let us define the set-point trajectory vector

$$\boldsymbol{y}^{\text{sp}}(k) = \begin{bmatrix} y^{\text{sp}}(k+1|k) \\ \vdots \\ y^{\text{sp}}(k+N|k) \end{bmatrix} \tag{1.21}$$

and the predicted output trajectory vector

$$\hat{\boldsymbol{y}}(k) = \begin{bmatrix} \hat{y}(k+1|k) \\ \vdots \\ \hat{y}(k+N|k) \end{bmatrix} \tag{1.22}$$

Both vectors are of length $n_{\text{y}}N$. The MPC cost-function (1.14) may be rewritten in the following compact form

$$J(k) = \|\boldsymbol{y}^{\text{sp}}(k) - \hat{\boldsymbol{y}}(k)\|^2_{\boldsymbol{M}} + \|\triangle \boldsymbol{u}(k)\|^2_{\boldsymbol{\Lambda}} \tag{1.23}$$

The matrices $\boldsymbol{M} = \text{diag}(\boldsymbol{M}_1, \ldots, \boldsymbol{M}_N) \geq 0$ and $\boldsymbol{\Lambda} = \text{diag}(\boldsymbol{\Lambda}_0, \ldots, \boldsymbol{\Lambda}_{N_{\text{u}}-1}) > 0$ are of dimensionality $n_{\text{y}}N \times n_{\text{y}}N$ and $n_{\text{u}}N_{\text{u}} \times n_{\text{u}}N_{\text{u}}$, respectively.

It is necessary to find the relation between the future values of the manipulated variables and their increments, which are calculated on-line in MPC. From the definitions of increments (Eqs. (1.4) and (1.5)), we have

$$\begin{aligned} u(k|k) &= \triangle u(k|k) + u(k-1) \\ u(k+1|k) &= \triangle u(k|k) + \triangle u(k+1|k) + u(k-1) \\ &\vdots \\ u(k+N_{\text{u}}-1|k) &= \triangle u(k|k) + \ldots + \triangle u(k+N_{\text{u}}-1|k) + u(k-1) \end{aligned} \tag{1.24}$$

which may be expressed as a general rule

$$u(k+p|k) = \sum_{i=0}^{p} \triangle u(k+i|k) + u(k-1) \tag{1.25}$$

for $p = 0, \ldots, N_\mathrm{u} - 1$. The above observation may be rewritten compactly

$$\boldsymbol{u}(k) = \boldsymbol{J} \triangle \boldsymbol{u}(k) + \boldsymbol{u}(k-1) \tag{1.26}$$

where

$$\boldsymbol{u}(k) = \begin{bmatrix} u(k|k) \\ \vdots \\ u(k + N_\mathrm{u} - 1|k) \end{bmatrix} \tag{1.27}$$

is a vector of length $n_\mathrm{u} N_\mathrm{u}$ that corresponds to the vector of increments $\triangle \boldsymbol{u}(k)$. Using Eq. (1.26), the scalar constraints (1.15) may be expressed compactly

$$\boldsymbol{u}^\mathrm{min} \leq \boldsymbol{J} \triangle \boldsymbol{u}(k) + \boldsymbol{u}(k-1) \leq \boldsymbol{u}^\mathrm{max} \tag{1.28}$$

where the vectors

$$\boldsymbol{u}^\mathrm{min} = \begin{bmatrix} u^\mathrm{min} \\ \vdots \\ u^\mathrm{min} \end{bmatrix}, \ \boldsymbol{u}^\mathrm{max} = \begin{bmatrix} u^\mathrm{max} \\ \vdots \\ u^\mathrm{max} \end{bmatrix} \tag{1.29}$$

and

$$\boldsymbol{u}(k-1) = \begin{bmatrix} u(k-1) \\ \vdots \\ u(k-1) \end{bmatrix} \tag{1.30}$$

are of length $n_\mathrm{u} N_\mathrm{u}$, the matrix

$$\boldsymbol{J} = \begin{bmatrix} \boldsymbol{I}_{n_\mathrm{u} \times n_\mathrm{u}} & \boldsymbol{0}_{n_\mathrm{u} \times n_\mathrm{u}} & \boldsymbol{0}_{n_\mathrm{u} \times n_\mathrm{u}} & \ldots & \boldsymbol{0}_{n_\mathrm{u} \times n_\mathrm{u}} \\ \boldsymbol{I}_{n_\mathrm{u} \times n_\mathrm{u}} & \boldsymbol{I}_{n_\mathrm{u} \times n_\mathrm{u}} & \boldsymbol{0}_{n_\mathrm{u} \times n_\mathrm{u}} & \ldots & \boldsymbol{0}_{n_\mathrm{u} \times n_\mathrm{u}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{I}_{n_\mathrm{u} \times n_\mathrm{u}} & \boldsymbol{I}_{n_\mathrm{u} \times n_\mathrm{u}} & \boldsymbol{I}_{n_\mathrm{u} \times n_\mathrm{u}} & \ldots & \boldsymbol{I}_{n_\mathrm{u} \times n_\mathrm{u}} \end{bmatrix}$$

is of dimensionality $n_\mathrm{u} N_\mathrm{u} \times n_\mathrm{u} N_\mathrm{u}$. The scalar constraints (1.16) may be expressed compactly

$$\triangle \boldsymbol{u}^\mathrm{min} \leq \triangle \boldsymbol{u}(k) \leq \triangle \boldsymbol{u}^\mathrm{max} \tag{1.31}$$

where the vectors

$$\triangle \boldsymbol{u}^\mathrm{min} = \begin{bmatrix} \triangle u^\mathrm{min} \\ \vdots \\ \triangle u^\mathrm{min} \end{bmatrix}, \ \triangle \boldsymbol{u}^\mathrm{max} = \begin{bmatrix} \triangle u^\mathrm{max} \\ \vdots \\ \triangle u^\mathrm{max} \end{bmatrix} \tag{1.32}$$

are of length $n_\mathrm{u} N_\mathrm{u}$. The scalar constraints (1.17) may be expressed compactly as

$$-\boldsymbol{y}^\mathrm{min} \leq \hat{\boldsymbol{y}}(k) \leq \boldsymbol{y}^\mathrm{max} \tag{1.33}$$

where the vectors

$$\boldsymbol{y}^{\min} = \begin{bmatrix} y^{\min} \\ \vdots \\ y^{\min} \end{bmatrix}, \ \boldsymbol{y}^{\max} = \begin{bmatrix} y^{\max} \\ \vdots \\ y^{\max} \end{bmatrix} \tag{1.34}$$

are of length $n_{\mathrm{y}}N_{\mathrm{y}}$. Taking into account the minimised cost-function (1.23) and the constraints (1.28), (1.31), (1.33), the general MIMO MPC optimisation problem (1.20) is rewritten in a very compact vector-matrix form

$$\begin{aligned} & \min_{\triangle \boldsymbol{u}(k)} \left\{ J(k) = \left\| \boldsymbol{y}^{\mathrm{sp}}(k) - \hat{\boldsymbol{y}}(k) \right\|_{\boldsymbol{M}}^2 + \left\| \triangle \boldsymbol{u}(k) \right\|_{\boldsymbol{\Lambda}}^2 \right\} \\ & \text{subject to} \\ & \boldsymbol{u}^{\min} \leq \boldsymbol{J} \triangle \boldsymbol{u}(k) + \boldsymbol{u}(k-1) \leq \boldsymbol{u}^{\max} \\ & \triangle \boldsymbol{u}^{\min} \leq \triangle \boldsymbol{u}(k) \leq \triangle \boldsymbol{u}^{\max} \\ & \boldsymbol{y}^{\min} \leq \hat{\boldsymbol{y}}(k) \leq \boldsymbol{y}^{\max} \end{aligned} \tag{1.35}$$

Since a mathematical model of the controlled process is used on-line for prediction and optimisation of the control policy, the MPC algorithms have the following advantages:

1. It is possible to control MIMO processes efficiently. When a series of classical single-loop PID controllers are used for the MIMO process, the consecutive controllers work independently; each of them has only one objective, i.e. control of only one controlled variable. When cross-couplings in the process (interactions of the consecutive manipulated variables with the consecutive controlled ones) are strong, such single-loop PID controllers do not work properly. Conversely, due to using a model for prediction, the MPC "knows" all interactions between process variables and calculates the best possible control policy.
2. The MPC algorithms may be used when the number of process inputs is different from the number of outputs. In such a case, it is practically impossible to use a set of single-loop PID controllers.
3. It is possible to take into account constraints imposed on both manipulated and predicted controlled variables in a simple way (MPC optimisation is simply carried out subject to all necessary constraints).
4. It is possible to control "difficult" processes, i.e. with significant time-delays or with the inverse step-response.

Additional advantages of MPC are:

1. Tuning of MPC algorithms is relatively easy. It is only necessary to select appropriate horizons and some weighting coefficients. All these parameters have a clear physical interpretation.
2. It is possible to take into account the measured disturbances of the process, i.e. the uncontrolled inputs (the feed-forward action).

3. Unlike the PID algorithm, future changes of the set-point trajectory over the prediction horizon may be easily taken into account.
4. The core idea of MPC is straightforward, which is important when advanced methods are introduced in industry [112, 177].

Let us emphasise the very significant role of the process model in MPC. The model is used for prediction. Intuitively, the better the model, the better (potentially) the resulting control accuracy. Moreover, without the model, it is impossible to use MPC at all. Let us also mention some other advanced model-based computational methods: fault diagnosis [81, 83, 145, 192] and fault-tolerant control [118, 145, 192].

An important question is how to assess the quality of control. In addition to typically used indicators, such as the sum of squared errors, overshoot and setting time, we can use more sophisticated indices, including fractal and entropy measures [36]. Effectiveness of such methods is discussed in [38, 39, 41] (for MPC algorithms based on linear models) and in [40, 42] (for nonlinear MPC algorithms). A review of control performance assessment methods for MPC is given in [37].

We have presented above the classical formulation of MPC. In the next parts of the book, we will detail computationally efficient nonlinear approaches. At this point we have to mention a few important extensions of MPC. In numerous industrial applications, when the objective is maximisation of production profits, set-point optimisation that cooperates with MPC [50, 91, 89, 177, 181] and economic MPC [49, 48, 107, 132] must be used. An excellent review of possible architectures for distributed and hierarchical MPC is given in [163]. MPC algorithms may also offer fault-tolerant control [118, 145, 167], which means that safe process operation is guaranteed in the case of some faults, e.g. when a sensors' or an actuators' malfunction occurs. It is also possible to take into account in MPC not only control accuracy and economic issues but also the remaining useful life of the system considered (health-aware MPC) [150]. An important direction of theoretical research is concerned with stable and robust versions of MPC algorithms [128, 129]. Different versions of such approaches are presented in [58, 117, 146, 144, 145, 174, 159, 182]. In the last years MPC schemes for fractional-order systems have gained popularity [43, 44, 45, 46, 135, 169]. The fractional-order approach makes it possible to control processes for which classical differential (or difference) equations are insufficient as models used for prediction in MPC.

1.2 How to Cope with Infeasibility Problem

In this work three different classes of constraints are taken into account in MPC optimisation taks (1.12), (1.20) and (1.35). The constraints may be imposed on: the values of the manipulated variables, the corresponding increments of those variables and on the predicted values of the controlled variables. The first two classes of constraints simply limit the feasible set of possible solutions of the MPC optimisation task. The third type of constraints may cause some important problems. Let us imagine that we require no overshoot. In order to achieve that, we use the constraints

$$\hat{y}(k+p|k) \leq y^{\mathrm{sp}}(k),\ p = 1, \ldots, N \tag{1.36}$$

If the model used for prediction is precise and there are no external disturbances, such constraints may work correctly provided that the constraints imposed on the manipulated variables are not too restrictive. It is also possible that the constraints (1.36) may be not satisfied because of the constraints imposed on the manipulated variables, even in the case of a perfect model and no disturbances. When the model is only a rough approximation of the process, which frequently happens, and/or the process is affected by a strong disturbance, it is very likely that it is impossible to calculate a decision variable vector which leads to satisfaction of the constraints (1.36). When such problems occur, the feasible set of the MPC optimisation problem is empty. In such a case, one may use for control at the current sampling instant the signals applied to the process at the previous sampling instant, i.e. $u(k-1)$, or the signals calculated at the previous sampling instant for the current sampling, i.e. $u(k|k-1)$. A more mathematically sound approach is to use soft output constraints [112, 177]. The original hard constraints (in the vector notation for a general MIMO process)

$$y^{\min} \leq \hat{y}(k+p|k) \leq y^{\max},\ p = 1, \ldots, N \tag{1.37}$$

are relaxed when they cannot be satisfied. It means that the predicted values of the controlled variables may temporarily violate the hard constraints. As a result, the feasible set is not empty. Using the soft constraints, the rudimentary MPC optimisation problem (1.20) becomes

$$\begin{aligned}
&\min_{\substack{\triangle \boldsymbol{u}(k) \\ \varepsilon^{\min}(k),\ \varepsilon^{\max}(k)}} \Bigg\{ J(k) = \sum_{p=1}^{N} \left\| y^{\mathrm{sp}}(k+p|k) - \hat{y}(k+p|k) \right\|^2_{\boldsymbol{M}_p} \\
&\qquad + \sum_{p=0}^{N_{\mathrm{u}}-1} \left\| \triangle u(k+p|k) \right\|^2_{\boldsymbol{\Lambda}_p} \\
&\qquad + \rho^{\min} \left\| \varepsilon^{\min}(k) \right\|^2 + \rho^{\max} \left\| \varepsilon^{\max}(k) \right\|^2 \Bigg\} \\
&\text{subject to} \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (1.38) \\
&u^{\min} \leq u(k+p|k) \leq u^{\max},\ p = 0, \ldots, N_{\mathrm{u}}-1 \\
&\triangle u^{\min} \leq \triangle u(k+p|k) \leq \triangle u^{\max},\ p = 0, \ldots, N_{\mathrm{u}}-1 \\
&y^{\min} - \varepsilon^{\min}(k) \leq \hat{y}(k+p|k) \leq y^{\max} + \varepsilon^{\max}(k),\ p = 1, \ldots, N \\
&\varepsilon^{\min}(k) \geq \mathbf{0}_{n_{\mathrm{y}} \times 1},\ \varepsilon^{\max}(k) \geq \mathbf{0}_{n_{\mathrm{y}} \times 1}
\end{aligned}$$

When the original hard constraints (1.37) cannot be satisfied, they are temporarily violated. It is done by relaxing the minimal and maximal predicted values of the controlled variables by $\varepsilon^{\min}(k)$ and $\varepsilon^{\max}(k)$, respectively. The MPC algorithm calculates not only the future control increments $\triangle \boldsymbol{u}(k)$ but also the vectors $\varepsilon^{\min}(k)$ and $\varepsilon^{\max}(k)$ of length n_{y}. Because it is natural that the original hard output constraints should be relaxed only when necessary, the degree of violations of the hard constraints is minimised in the cost-function by additional penalty terms; $\rho^{\mathrm{min}}, \rho^{\mathrm{max}} > 0$ are penalty coefficients. Additionally, the last two constraints require that the degree of constraints' violation is non-negative. The number of decision variables of the optimisation problem (1.38) is $n_{\mathrm{u}}N_{\mathrm{u}} + 2n_{\mathrm{y}}$, the number of constraints is $4n_{\mathrm{u}}N_{\mathrm{u}} + 2n_{\mathrm{y}}N + 2n_{\mathrm{y}}$.

Using the vector-matrix notation, the rudimentary MPC optimisation problem with soft output constraints (1.38) may be easily transformed to the following task in a compact vector-matrix notation, similar to the task (1.35)

$$\begin{aligned}
&\min_{\substack{\triangle \boldsymbol{u}(k)\\ \varepsilon^{\mathrm{min}}(k),\ \varepsilon^{\mathrm{max}}(k)}} \Big\{ J(k) = \left\| \boldsymbol{y}^{\mathrm{sp}}(k) - \hat{\boldsymbol{y}}(k) \right\|_{\boldsymbol{M}}^2 + \left\| \triangle \boldsymbol{u}(k) \right\|_{\boldsymbol{\Lambda}}^2 \\
&\qquad\qquad + \rho^{\mathrm{min}} \left\| \varepsilon^{\mathrm{min}}(k) \right\|^2 + \rho^{\mathrm{max}} \left\| \varepsilon^{\mathrm{max}}(k) \right\|^2 \Big\} \\
&\text{subject to} \\
&\boldsymbol{u}^{\mathrm{min}} \leq \boldsymbol{J} \triangle \boldsymbol{u}(k) + \boldsymbol{u}(k-1) \leq \boldsymbol{u}^{\mathrm{max}} \\
&\triangle \boldsymbol{u}^{\mathrm{min}} \leq \triangle \boldsymbol{u}(k) \leq \triangle \boldsymbol{u}^{\mathrm{max}} \\
&\boldsymbol{y}^{\mathrm{min}} - \underline{\boldsymbol{\varepsilon}}^{\mathrm{min}}(k) \leq \hat{\boldsymbol{y}}(k) \leq \boldsymbol{y}^{\mathrm{max}} + \underline{\boldsymbol{\varepsilon}}^{\mathrm{max}}(k) \\
&\varepsilon^{\mathrm{min}}(k) \geq \boldsymbol{0}_{n_{\mathrm{y}} \times 1},\ \varepsilon^{\mathrm{max}}(k) \geq \boldsymbol{0}_{n_{\mathrm{y}} \times 1}
\end{aligned} \tag{1.39}$$

where the vectors of length $n_{\mathrm{y}}N$ are

$$\underline{\boldsymbol{\varepsilon}}^{\mathrm{min}}(k) = \begin{bmatrix} \varepsilon^{\mathrm{min}}(k) \\ \vdots \\ \varepsilon^{\mathrm{min}}(k) \end{bmatrix},\ \underline{\boldsymbol{\varepsilon}}^{\mathrm{max}}(k) = \begin{bmatrix} \varepsilon^{\mathrm{max}}(k) \\ \vdots \\ \varepsilon^{\mathrm{max}}(k) \end{bmatrix} \tag{1.40}$$

In the soft output approach it is possible to allow that the degree of relaxation of the same controlled variable may change over the prediction horizon. In such a case, in the optimisation problem (1.38), the soft constraints are

$$y^{\mathrm{min}} - \varepsilon^{\mathrm{min}}(k+p) \leq \hat{y}(k+p|k) \leq y^{\mathrm{max}} + \varepsilon^{\mathrm{max}}(k+p),\ p = 1, \ldots, N \tag{1.41}$$

The vectors of additional decision variables of the MPC optimisation task are now

$$\boldsymbol{\varepsilon}^{\mathrm{min}}(k) = \begin{bmatrix} \varepsilon^{\mathrm{min}}(k+1|k) \\ \vdots \\ \varepsilon^{\mathrm{min}}(k+N|k) \end{bmatrix},\ \boldsymbol{\varepsilon}^{\mathrm{max}}(k) = \begin{bmatrix} \varepsilon^{\mathrm{max}}(k+1|k) \\ \vdots \\ \varepsilon^{\mathrm{max}}(k+N|k) \end{bmatrix} \tag{1.42}$$

Unfortunately, the number of decision variables increases to $n_{\mathrm{u}}N_{\mathrm{u}}+2n_{\mathrm{y}}N$, the number of constraints is $4n_{\mathrm{u}}N_{\mathrm{u}} + 4n_{\mathrm{y}}N$. In practical applications of MPC, the assumption that the output constraints are relaxed by the same degree for the whole prediction horizon (for the consecutive controlled variables) and only $2n_{\mathrm{y}}$ additional variables are used gives very good results, very close to those possible when as many as $2n_{\mathrm{y}}N$ additional variables are necessary [96].

1.3 Parameterisation of Decision Variables

Laguerre, Kautz and other orthonormal functions may be successfully used for modelling of dynamical systems in linear [137] and nonlinear [138] cases, respectively. Application of orthonormal Laguerre functions to parameterise the calculated future sequence of the manipulated variables may be used in MPC algorithms based on linear state-space models: in continuous-time [186] and discrete-time [187] versions, respectively, as well as in the DMC algorithm, in which a step-response model is used for prediction [178]. A systematic tuning methodology to find parameters of Laguerre functions in parameterised MPC is discussed in [61, 75]. MPC algorithms with Laguerre parameterisation have been developed for different technological processes. Example applications include: buildings [19], wave energy converters [69], magnetically actuated satellites [76], wind turbines [84], hexacopters [104] and power systems [202]. All cited MPC algorithms use linear models for prediction. In this book, the Laguerre functions are used to parameterise the decision vector of all discussed nonlinear MPC algorithms, i.e. to reduce the number of decision variables that are actually optimised on-line.

At first, let us consider the SISO case. Let $l_1(k),\ldots,l_{n_{\mathrm{L}}}(k)$ denote n_{L} Laguerre functions. The transfer function of the Laguerre function of the order n is [185]

$$G_n(z) = \frac{\sqrt{1-(a_{\mathrm{L}})^2}}{z-a_{\mathrm{L}}}\left(\frac{1-a_{\mathrm{L}}z}{z-a_{\mathrm{L}}}\right)^{n-1} \tag{1.43}$$

where a_{L} is a scaling factor, often named a Laguerre pole. For stability, the condition $0 \leq a_{\mathrm{L}} < 1$ must be satisfied. The transfer functions $G_n(z)$ satisfy the following orthonormality conditions

$$\frac{1}{2\pi}\int_{-\pi}^{\pi} G_n(e^{j\omega})G_n(e^{j\omega})^*\mathrm{d}\omega = 1 \tag{1.44}$$

$$\frac{1}{2\pi}\int_{-\pi}^{\pi} G_m(e^{j\omega})G_n(e^{j\omega})^*\mathrm{d}\omega = 0 \text{ for } m \neq n \tag{1.45}$$

where $G_n(e^{j\omega})^*$ denotes complex conjugate of the transfer function $G_n(e^{j\omega})$. The Laguerre functions are defined as inverse $\mathcal{Z}$-transforms of the transfer functions $G_n(z)$

$$l_n(k) = \mathcal{Z}^{-1}(G_n(z)) \tag{1.46}$$

Taking into account the structure of the obtained Laguerre functions, it may be found that [187]

$$L(k+1) = \Omega L(k) \tag{1.47}$$

where the vector of length n_{L} is

$$L(k) = \left[l_1(k)\ldots l_{n_{\mathrm{L}}}(k)\right]^{\mathrm{T}} \tag{1.48}$$

and the matrix of dimensionality $n_{\mathrm{L}} \times n_{\mathrm{L}}$ is

$$\Omega = \begin{bmatrix} a_{\mathrm{L}} & 0 & 0 & \ldots & 0 \\ \beta_{\mathrm{L}} & a_{\mathrm{L}} & 0 & \ldots & 0 \\ -a_{\mathrm{L}}\beta_{\mathrm{L}} & \beta_{\mathrm{L}} & a_{\mathrm{L}} & \ldots & 0 \\ a_{\mathrm{L}}^2\beta_{\mathrm{L}} & -a_{\mathrm{L}}\beta_{\mathrm{L}} & \beta_{\mathrm{L}} & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ (-a_{\mathrm{L}})^{n_{\mathrm{L}}-2}\beta & (-a_{\mathrm{L}})^{n_{\mathrm{L}}-3}\beta & \ldots & \beta_{\mathrm{L}} & a_{\mathrm{L}} \end{bmatrix} \tag{1.49}$$

The initial condition is

$$L(0) = \sqrt{1 - a_{\mathrm{L}}^2} \begin{bmatrix} 1 \\ -a_{\mathrm{L}} \\ a_{\mathrm{L}}^2 \\ -a_{\mathrm{L}}^3 \\ \vdots \\ (-a_{\mathrm{L}})^{n_{\mathrm{L}}-1} \end{bmatrix} \tag{1.50}$$

and $\beta_{\mathrm{L}} = 1 - a_{\mathrm{L}}^2$. The orthonormality conditions (1.44)-(1.45) may also be formulated for the discrete-time description

$$\sum_{k=0}^{\infty} l_i(k) l_j(k) = 0 \text{ for } i \neq j \tag{1.51}$$

$$\sum_{k=0}^{\infty} l_i(k) l_j(k) = 1 \text{ for } i = j \tag{1.52}$$

The idea of parameterisation is to eliminate the necessity of calculating at each sampling instant as many as N_{u} future increments $\triangle u(k|k), \ldots, \triangle u(k + N_{\mathrm{u}} - 1|k)$, i.e. the whole vector $\triangle \boldsymbol{u}(k)$ (Eq. (1.3)). The future control increments are parameterised using the Laguerre functions in the following way [187]

$$\triangle u(k + p|k) = \sum_{i=1}^{n_{\mathrm{L}}} l_i(p) c_i(k) \tag{1.53}$$

Using the vector notation, we have

$$\triangle u(k + p|k) = L^{\mathrm{T}}(p) c(k) \tag{1.54}$$

where the vector of coefficients is

$$c(k) = \left[c_1(k) \ldots c_{n_{\mathrm{L}}}(k)\right]^{\mathrm{T}} \tag{1.55}$$

For the whole vector of future increments of the manipulated variable over the control horizon, we have

$$\triangle \boldsymbol{u}(k) = \boldsymbol{L}c(k) \tag{1.56}$$

where the matrix of dimensionality $N_{\mathrm{u}} \times n_{\mathrm{L}}$ is

$$\boldsymbol{L} = \begin{bmatrix} l_1(0) & l_2(0) & \ldots & l_{n_{\mathrm{L}}}(0) \\ l_1(1) & l_2(1) & \ldots & l_{n_{\mathrm{L}}}(1) \\ \vdots & \vdots & \ddots & \vdots \\ l_1(N_{\mathrm{u}}-1) & l_2(N_{\mathrm{u}}-1) & \ldots & l_{n_{\mathrm{L}}}(N_{\mathrm{u}}-1) \end{bmatrix} \tag{1.57}$$

In parameterised MPC the vector of decision variables is $c(k)$, not $\triangle \boldsymbol{u}(k)$. Since $n_{\mathrm{L}} < N_{\mathrm{u}}$, the number of decision variables used in the MPC optimisation problem solved on-line is reduced. Having calculated the optimal vector $c^{\mathrm{opt}}(k)$ from the MPC optimisation problem, using Eq. (1.56) and taking into account the structure of the matrix $\boldsymbol{L}$ given by Eq. (1.57), the current optimal value of the manipulated variable is calculated from

$$u(k) = \begin{bmatrix} l_1(0) \; l_2(0) \; \ldots \; l_{n_{\mathrm{L}}}(0) \end{bmatrix} c^{\mathrm{opt}}(k) + u(k-1) \tag{1.58}$$

and applied to the process.

Having discussed the SISO case, we will consider parameterisation using Laguerre functions for MIMO processes. In order to obtain a flexible solution, we assume that for the consecutive manipulated variables separate Laguerre poles $a_{\mathrm{L}}^1, \ldots, a_{\mathrm{L}}^{n_{\mathrm{u}}}$ are used. Furthermore, we also assume that for the consecutive variables different numbers of Laguerre functions are possible, i.e. $n_{\mathrm{L}}^1, \ldots, n_{\mathrm{L}}^{n_{\mathrm{u}}}$. Similarly to Eq. (1.53) used in the SISO case, the future control increments are parameterised in the following way

$$\triangle u_1(k+p|k) = \sum_{i=1}^{n_{\mathrm{L}}^1} l_{1,i}(p) c_{1,i}(k) \tag{1.59}$$

$$\vdots$$

$$\triangle u_{n_{\mathrm{u}}}(k+p|k) = \sum_{i=1}^{n_{\mathrm{L}}^{n_{\mathrm{u}}}} l_{n_{\mathrm{u}},i}(p) c_{n_{\mathrm{u}},i}(k) \tag{1.60}$$

In place of Eq. (1.54), we have

$$\triangle u_1(k+p|k) = L_1^{\mathrm{T}}(p) c_1(k) \tag{1.61}$$

$$\vdots$$

$$\triangle u_{n_{\mathrm{u}}}(k+p|k) = L_{n_{\mathrm{u}}}^{\mathrm{T}}(p) c_{n_{\mathrm{u}}}(k) \tag{1.62}$$

where the vectors of coefficients, of length $n_{\mathrm{L}}^1, \ldots, n_{\mathrm{L}}^{n_{\mathrm{u}}}$, respectively, are

$$c_1(k) = \begin{bmatrix} c_{1,1}(k) \\ \vdots \\ c_{1,n_{\mathrm{L}}^1}(k) \end{bmatrix}, \ldots, c_{n_{\mathrm{u}}}(k) = \begin{bmatrix} c_{n_{\mathrm{u}},1}(k) \\ \vdots \\ c_{n_{\mathrm{u}},n_{\mathrm{L}}^{n_{\mathrm{u}}}}(k) \end{bmatrix} \tag{1.63}$$

For all manipulated variables and the whole vector of future increments over the control horizon, for the MIMO process we also obtain Eq. (1.56), the same as in the SISO case, but now the vector $\triangle \boldsymbol{u}(k)$ is of length $n_{\mathrm{u}}N_{\mathrm{u}}$ and the matrix of dimensionality $n_{\mathrm{u}}N_{\mathrm{u}} \times (n_{\mathrm{L}}^1 + \ldots + n_{\mathrm{L}}^{n_{\mathrm{u}}})$ has the general structure

$$\boldsymbol{L} = \begin{bmatrix} \boldsymbol{L}_1 & \boldsymbol{0}_{N_{\mathrm{u}} \times n_{\mathrm{L}}^2} & \boldsymbol{0}_{N_{\mathrm{u}} \times n_{\mathrm{L}}^3} & \cdots & \boldsymbol{0}_{N_{\mathrm{u}} \times n_{\mathrm{L}}^{n_{\mathrm{u}}}} \\ \boldsymbol{0}_{N_{\mathrm{u}} \times n_{\mathrm{L}}^1} & \boldsymbol{L}_2 & \boldsymbol{0}_{N_{\mathrm{u}} \times n_{\mathrm{L}}^3} & \cdots & \boldsymbol{0}_{N_{\mathrm{u}} \times n_{\mathrm{L}}^{n_{\mathrm{u}}}} \\ \boldsymbol{0}_{N_{\mathrm{u}} \times n_{\mathrm{L}}^1} & \boldsymbol{0}_{N_{\mathrm{u}} \times n_{\mathrm{L}}^2} & \boldsymbol{L}_3 & \cdots & \boldsymbol{0}_{N_{\mathrm{u}} \times n_{\mathrm{L}}^{n_{\mathrm{u}}}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0}_{N_{\mathrm{u}} \times n_{\mathrm{L}}^1} & \boldsymbol{0}_{N_{\mathrm{u}} \times n_{\mathrm{L}}^2} & \boldsymbol{0}_{N_{\mathrm{u}} \times n_{\mathrm{L}}^3} & \cdots & \boldsymbol{L}_{n_{\mathrm{u}}} \end{bmatrix} \tag{1.64}$$

where the consecutive submatrices of dimensionality $N_{\mathrm{u}} \times n_{\mathrm{L}}^n$ are

$$\boldsymbol{L}_n = \begin{bmatrix} l_{n,1}(0) & l_{n,2}(0) & \ldots & l_{n,n_{\mathrm{L}}^n}(0) \\ l_{n,1}(1) & l_{n,2}(1) & \ldots & l_{n,n_{\mathrm{L}}^n}(1) \\ \vdots & \vdots & \ddots & \vdots \\ l_{n,1}(N_{\mathrm{u}}-1) & l_{n,2}(N_{\mathrm{u}}-1) & \ldots & l_{n,n_{\mathrm{L}}^n}(N_{\mathrm{u}}-1) \end{bmatrix} \tag{1.65}$$

for $n = 1, \ldots, n_{\mathrm{u}}$. The vector of optimised decision variables is of length $n_{\mathrm{L}}^1 + \ldots + n_{\mathrm{L}}^{n_{\mathrm{u}}}$ and has the structure

$$c(k) = \begin{bmatrix} c_1(k) \\ \vdots \\ c_{n_{\mathrm{u}}}(k) \end{bmatrix} \tag{1.66}$$

where the subvectors are defined by Eqs. (1.63).

Having calculated the optimal vector $c^{\mathrm{opt}}(k)$ from the MPC optimisation problem, using Eq. (1.56) and taking into account the matrices $\boldsymbol{L}$ and $\boldsymbol{L}_n$, given by Eqs. (1.64) and (1.65), respectively, the current optimal values of the manipulated variables are calculated from

$$u_1(k) = \left[l_{1,1}(0) \; l_{1,2}(0) \; \ldots \; l_{1,n_{\mathrm{L}}^1}(0) \right] c_1^{\mathrm{opt}}(k) + u_1(k-1) \tag{1.67}$$

$$\vdots$$

$$u_{n_{\mathrm{u}}}(k) = \left[l_{n_{\mathrm{u}},1}(0) \; l_{n_{\mathrm{u}},2}(0) \; \ldots \; l_{n_{\mathrm{u}},n_{\mathrm{L}}^{n_{\mathrm{u}}}}(0) \right] c_{n_{\mathrm{u}}}^{\mathrm{opt}}(k) + u_{n_{\mathrm{u}}}(k-1) \tag{1.68}$$

and applied to the process.

1.4 Computational Complexity of MPC Algorithms

In the simplest case, a linear model is used in MPC for prediction and no constraints are taken into account. A few different such MPC methods have been developed, with different structures of linear models. To name the most important MPC approaches based on linear models, we have to mention the following ones:

1. The Predictive Functional Control (PFC) algorithm (also known under the name Model Heuristic Predictive Control (MHPC)) [156, 157] in which the impulse-response process representations are used.
2. The Dynamic Matrix Control (DMC) algorithm [29] in which the step-response models are used.
3. The Generalized Predictive Control (GPC) algorithm [27] in which the discrete-time transfer functions are used.
4. The MPC algorithm with state-space models (MPCS) [112, 177] in which the classical linear state-space models are used.

The use of a linear model implies that the predicted trajectory of the manipulated variables (Eq. (1.22)) is a linear function of the decision variable vector (1.3). Remembering that the typical minimised MPC cost-function is of the quadratic type (Eq. (1.13)), we obtain an unconstrained quadratic optimisation problem. It may be solved analytically, without on-line optimisation. The future increments of the manipulated variables are linear functions of the following: the model parameters, some values of the manipulated variables computed at the previous sampling instants and the values of the process controlled variables measured at the previous sampling instants. Hence, such unconstrained MPC methods are named *unconstrained linear explicit MPC algorithms*.

If a linear model is used for prediction, but the constraints must be taken into account, at each sampling instant, it is necessary to solve on-line a quadratic optimisation task (a quadratic minimised cost-function and linear constraints). Such methods are named *constrained linear MPC algorithms* or, better, *constrained MPC algorithms based on linear models* since in the constrained case, the explicit linear solution does not exist, the optimal solution is obtained as a result of on-line optimisation. Depending on the model used, we obtain constrained MHPC, DMC, GPC and MPCS algorithms. For linear models, provided that $\mu_{p,m} \geq 0$ and $\lambda_{p,n} > 0$, the optimisation task has only one solution, which is the global one. Different approaches may be used to find the solution of the quadratic optimisation MPC problem [171]: the active-set methods, the interior-point ones and the first-order ones. It is necessary to point out that many very computationally efficient quadratic optimisation solvers are available, e.g. qpOASES [52], CVXGEN [126] and OSQP [171]. To speed up calculations, advanced quadratic optimisation algorithms may be specially tailored for MPC, i.e. the special form of the MPC optimisation task may be exploited. They may be used not only for industrial control applications [13] but also in embedded systems [16, 78, 158], for which sampling times are very short, of the order of hundreds, teens or even single milliseconds.

As described in Section 1.3, some basis functions, e.g. Laguerre orthonormal functions, may be used to reduce the number of decision variables of the MPC optimisation problem. The sequence of future manipulated variables is parameterised using a set of basis functions. The optimisation routine does not directly calculate the future manipulated variables or the corresponding increments but the coefficients of the basis functions. In the literature, a few variants of MPC algorithms which use that concept are described [178, 186, 187]. The parameterisation approach may be used in unconstrained linear explicit MPC algorithms and constrained MPC algorithms based on linear models. A similar approach is used in the PFC algorithm, which also uses linear models for prediction [156]. Finally, parameterisation may be used in the nonlinear MPC algorithms [98] which are discussed in the following chapters of this book.

Although the classical quadratic optimisation MPC problem is quite simple, in some applications, it would be best to eliminate the necessity of on-line optimisation at all. It can be proven that for a linear model and the typical quadratic cost-function, the optimal solution of the constrained quadratic optimisation MPC problem is a function of the state [15, 179]. That observation leads to *constrained linear explicit MPC algorithms*. The whole state domain is divided into a number of sets. For each set, the explicit control law is derived off-line. During on-line control, it is only necessary to determine to which set the current state of the process belongs and to use the corresponding precalculated control law; no on-line optimisation is necessary. Although the idea seems to be generally simple and intuitive, it may turn out that many (dozens or even hundreds) of sets and local control laws are required for typical processes.

When a general nonlinear model is used for prediction, the predicted trajectory (1.22) is a nonlinear function of the decision variable vector (1.3). Thus, the minimised cost-function (Eq. (1.13)) is not quadratic but nonlinear. The constraints imposed on the magnitude and on the rate of change of the manipulated variables are linear, but the constraints put on the predicted values of the controlled variables are nonlinear. The general class of the discussed approach is known as fully-fledged *constrained nonlinear MPC algorithms* or *constrained MPC algorithms with nonlinear optimisation*. A constrained nonlinear MPC optimisation problem must be solved on-line at each sampling instant. There are two difficulties of that approach. Firstly, nonlinear optimisation algorithms must be used. They are much more complicated than the classical quadratic optimisation ones. Solution of a constrained nonlinear optimisation task may need a lot of time. It is particularly important in the case of fast dynamical systems, for which very short sampling times are required. Secondly, it is possible that not only one global but several local minima exist. When a suboptimal solution is used for control, the resulting control quality may be lower than expected.

Typically, the Newton-like nonlinear optimisation algorithms are used. The Sequential Quadratic Programming (SQP) [151] and Interior Point (IP) [20] methods are the most frequently used ones in nonlinear MPC. Efficient implementation methods for SQP and IP algorithms have been developed which exploit the particular structure of the MPC optimisation task [53, 153]. Specialised nonlinear optimisation methods, developed with the aim of being used to solve MPC optimisation problems, make it possible to carry out parallel calculations [31, 199]. When the model used for prediction is comprised of a set of differential-algebraic equations, specialised optimisation methods must be used [33]. An excellent review of possible approaches to nonlinear optimisation in MPC is given in [34]. Very infrequently, for nonlinear optimisation other algorithms may be used, e.g. the golden section method [114, 193] or the branch-and-bound approach [195].

When the process dynamics is slow, which makes it possible to use relatively long sampling periods, we may use heuristic global optimisation algorithms. For example, applications of genetic algorithms to solve the constrained nonlinear MPC optimisation task may be found in [103, 149]. Specialised genetic operators (mutation and crossover) are used, tailored for the nature of MPC. An alternative is to use the particle swarm optimisation algorithm [25, 191]. Another option is to use simulated annealing for nonlinear optimisation [1]. It must be stressed that application of heuristic optimisation methods is limited.

There are, however, some deterministic global optimisation methods [164] that may be used in MPC [47]. The cited method is based on a convex relaxation of the MPC cost-function. It is reported to significantly reduce dimensionality of the MPC optimisation task, which lower the overall computational burden. To further reduce computational complexity, a neural multi-model is used rather than one dynamical model applied recurrently.

In practice, *fuzzy MPC* is a very important alternative. To control a nonlinear process, a set of simple local MPC controllers is used. The local controllers are switched on-line, taking into account the current operating point of the process and/or the set-point. Both the unconstrained linear explicit MPC methods and the constrained MPC algorithms based on linear models may be used as local controllers. It is important that the local controllers are developed off-line. During on-line control, it is only necessary to combine the values of the manipulated variables computed by the local controllers in a fuzzy way. Fuzzy DMC algorithms [30, 119, 125] and fuzzy GPC methods are given as examples of the described approach [177]. Advanced methods utilised for prediction generation in the fuzzy DMC algorithm are discussed in [123, 124]. A similar idea is to use multi-linear models for prediction in MPC [200]. A specialised procedure is used to determine the multi-linear process representation from nonlinear Hammerstein or Wiener models.

There are numerous attempts to simplify the general nonlinear MPC optimisation task that must be solved at each sampling instant on-line. The following methods are reported in the literature:

1. The first n_{u} elements of the future control policy are computed from a nonlinear optimisation task, whereas the remaining ones are found from an explicit control law [201]. As a result, the optimisation problem is still nonlinear, but the number of decision variables is equal to n_{u}, not to $n_{\mathrm{u}}N_{\mathrm{u}}$ as in the rudimentary approach.
2. The technique named move blocking may be used [21]. The degree of freedom is reduced by fixing the manipulated variables or their derivatives to be constant over several time-steps. Some of such methods guarantee stability and satisfaction of constraints.

3. Compression of the constraint set is possible [102]. It simplifies the MPC optimisation task. Such an approach may be used together with the move blocking technique.
4. The domain of the calculated manipulated variable may be discretised [115] (in the cited approach, the control horizon is equal to 1). A simple procedure determines its best value and on-line optimisation is not necessary. A more advanced graph search method for finding the control policy is used in [155].
5. In the case of the cascade models, the inverse of the static part of the model may be used to make an attempt to cancel the effect of nonlinearity. It makes it possible to formulate the classical quadratic optimisation MPC problem. For the Hammerstein structure, such an approach is discussed in [54], for the Wiener structure in [4, 23, 70, 133, 134, 168]. The same method may also be used for cascade models with 3 blocks, e.g. the Hammerstein-Wiener ones as described in [35, 63, 147]. As pointed out in Section 3.1, the discussed approach has important structural disadvantages and limitations. Moreover, as demonstrated in simulations discussed in this book, it is very sensitive to model errors and disturbances.
6. In the fast MPC algorithm [190] the MPC optimisation task is not solved precisely but in an approximate way. Although it may have a negative effect on the resulting control quality, the time of calculations necessary at each sampling instant is likely to be significantly reduced. As proved in [165], for stability, it is sufficient to use a feasible control strategy, i.e. the one that satisfies all the existing constraints, not the optimal one.
7. The numerical optimisation procedure used in the MPC algorithm may be replaced by a specially designed neural network which acts as a neural optimiser. There are a few neural structures which solve the quadratic optimisation problem [109, 188]. The network described in [109] is used for optimisation in an MPC algorithm based on a linear model [141] and in an MPC algorithm with on-line model linearisation [140].
8. The MPC algorithm may be replaced by a specially designed neural network which acts as a neural approximator that attempts to mimic the whole MPC algorithm [2, 142]. At first, the classical nonlinear MPC algorithm is developed and run on-line (or off-line in simulations) for different operating conditions and set-points. A data set is collected and next used to train a neural approximator. For a given operating point of the process, determined by measurements of the process input and output variables, as well as the set-point, the approximator finds the current values of the manipulated variables. An approximator may also be used to find the initial solution of the MPC optimisation problem [180]. Finding the initial solution is likely to significantly shorten the calculation time in embedded, microprocessor-based systems [77].
9. The prediction and control horizons may be equal to 1 and the current value of the manipulated variable may be computed by a simple binary search algorithm [160].
10. The Experience-driven Predictive Control (EPC) algorithm constructs a database of feedback controllers that are parameterised by the system dynamics [32].

When, for given conditions, the control law does not exist, it is calculated by a conventional MPC algorithm based on a linear model. In order to obtain a quadratic optimisation task, for prediction Locally-Weighted Projection Regression (LWPR) models are used, which allow for easy on-line model adaptation.

11. The nonlinear optimisation MPC problem is relaxed into a Mixed Integer Linear Programming (MILP) one. Next, the solution of the MILP problem is taken as a starting point of the nonlinear one [189].
12. *Constrained explicit nonlinear MPC algorithms* are possible [57, 71]. Unfortunately, a huge number of local control laws may be necessary.
13. A specialised model may be used in which the output values for the consecutive sampling instants within the prediction horizon are linear functions of the calculated future manipulated variables, but they are nonlinear functions of the past (the quasi-linear model) [106]. Such an approach results in a quadratic optimisation MPC task. Neural networks are used for modelling.
14. When Linear Parameter Varying (LPV) models are used for prediction, the general nonlinear optimisation problem is replaced by a convex Linear Matrix Inequalities (LMIs) optimisation task [203, 205, 204]. Neural networks may calculate coefficients of the LPV models.
15. Model convexity may be achieved when Input Convex Neural Networks (ICNNs) are used [8]. ICNNs are obtained by explicitly constraining the model outputs to be convex functions of the inputs during model development. As a result, convex MPC optimisation problems are obtained: unconstrained [26] or constrained ones [196].
16. A class of linear predictors may be used to describe a nonlinear system [127]. The key step in obtaining such accurate predictions is to lift (or embed) the nonlinear dynamics into a higher dimensional space in which its evolution of this lifted state is (approximately) linear. The idea corresponds to the Koopman operator [79, 80]. When such a model is used in MPC, we obtain a quadratic MPC optimisation task [82, 127]. An alternative method, named polyflows, is discussed in [72].

Finally, on-line linearisation must be discussed as the method which makes it possible to significantly reduce computational burden of nonlinear MPC. Details of numerous such MPC methods are presented in Chapters 3 and 7 for input-output and state-space Wiener process descriptions, respectively. Let us now only give a short literature review. In general, two categories of computationally efficient MPC algorithms may be distinguished: with on-line model linearisation and with on-line trajectory linearisation. In both cases, we obtain computationally simple quadratic optimisation problems, the necessity of on-line nonlinear optimisation is eliminated.

In the simplest approach, a linear approximation of the nonlinear model is computed on-line for the current operating point of the process. Typically, model linearisation is performed at each sampling instant but, for some “less nonlinear” processes or when changes of the set-point are slow and infrequent, model linearisation may be repeated less frequently. Next, the obtained linearised model is used to calculate the predicted trajectory of the controlled variables. Thanks to linearisation, the predicted trajectory is a linear function of the vector of decision variables (1.3), which is a characteristic feature of the classical MPC algorithms based on linear models.

Hence, a quadratic optimisation problem is formulated when the constraints must be taken into account or even the explicit unconstrained solution is possible.

The *MPC algorithms with on-line model linearisation* may be divided into two categories [91, 177]. In the first one, the time-varying linear approximation of the rudimentary nonlinear model is used to calculate future predictions and the influence of the past, i.e. the free trajectory. In the second approach to MPC with successive linearisation, the linearised model is only used to calculate the future predictions, whereas the nonlinear model is used to find the nonlinear free trajectory. The first approach is used to control a spark-ignition engine in [28] and an aircraft gas turbine engine in [130]. Applications to a polymerisation reactor and a distillation column are presented in [85]. When necessary, the nonlinear model may be retrained on-line as shown in [3], applications of the algorithm to a fluidised bed furnace reactor and the autopilot of the F-16 aircraft are described. An application to a boiler-turbine unit in a power plant described by a state-space process model is detailed in [96], two variants of soft constraints are considered. Although the algorithm may be implemented for practically any differentiable model, a straightforward calculation is possible for Wiener structures since the linearised model is found in a simplified way, as a multiplication of the linear dynamic part and the time-varying gain of the nonlinear static block [5]. A similar calculation method is possible for the Hammerstein model. The second approach, i.e. with the nonlinear free trajectory, is used to control a solar power plant in [9, 17], a spark-ignition engine [162], a yeast fermentation reactor [91], a polymerisation reactor and a distillation column [85]. Also in the second approach simple calculations are possible when Hammerstein [91, 121] or Wiener [87, 91, 120, 122] models are used.

In more advanced *MPC algorithms with on-line trajectory linearisation*, not the model itself is linearised, but a linear approximation of the predicted trajectory of the controlled variables over the whole prediction horizon is directly calculated. Unlike the simple MPC algorithms with model linearisation, linearisation is not performed for the current operating point of the process, defined by past measurements of the process input and output signals, but carried out along some future trajectory of the manipulated variables defined for the whole control horizon. Similarly to the simple algorithm with on-line model linearisation, a quadratic optimisation problem is next formulated. The explicit unconstrained solution is also possible. In practice, the classical MPC algorithm with model linearisation may be used when the process is close to the desired set-point. If it is not true, the calculated solution defines the future trajectory of the manipulated variables along which a linear approximation of the predicted trajectory of the controlled variables is calculated. Such a hybrid MPC structure is presented in [88, 91], an application to a high-pressure distillation column is discussed. An application of the algorithm to a solid oxide fuel cell is presented in [97], the method of coping with infeasibility caused by linearisation of nonlinear technological constraints (fuel utilisation) are discussed. The MPC algorithm with trajectory linearisation is also of course possible when the process is described by cascade models, including: Hammerstein [91] (for a polymerisation reactor benchmark), Wiener [94] (for a neutralisation reactor) and [100] (for a proton exchange membrane fuel cell), Hammerstein-Wiener [93] as well as Wiener-

Hammerstein [95] (for a heat exchanger) structures. Although all cited works are concerned with the input-output process representation, the MPC algorithm with trajectory linearisation is, of course, possible for the state-space representation [101] (implementation details for the Wiener model are given).

Finally, let us mention computationally efficient *MPC algorithms with on-line linearisation and approximation*. The approximator is used in order to eliminate some calculations that must be repeated at each sampling instant. They are necessary in the classical MPC algorithms with on-line linearisation. Successive model linearisation and prediction calculation may be simplified using an approximator which directly estimates, at each sampling instant, the time-varying matrix of step response coefficients of the linearised model [91]. An application of that approach to a simulated distillation column is detailed in [90]. The same approximation method may be used in the nonlinear DMC algorithm [86, 91]. A significant reduction of computational complexity in comparison with the classical MPC algorithms with on-line linearisation may be obtained when explicit unconstrained versions of the discussed algorithms are considered. It may be proved [91, 92] that in such a case, the optimal vector of the decision variable vector (1.3) is a linear function of the set-point, model parameters and some past measurements. The time-varying vector of coefficients of the control law is determined on-line by a neural approximator for the current operating point. As a result, on-line model linearisation and some other calculations are not necessary, which significantly speeds up calculations. A simulation study concerned with a high-pressure distillation process is presented in [91, 92]. In all mentioned cases, neural networks are used as approximators, although other structures are also possible.

1.5 Example Applications of MPC Algorithms

MPC is regarded as the only one among the advanced control techniques, defined as more advanced than the classical PID controller, which is successfully used in numerous industrial applications [152]. Let us cite a number of typical applications. Traditionally, MPC algorithms may be successfully used for controlling the following industrial processes:

- chemical reactors [64, 166, 175, 198],
- distillation columns [11, 65, 111, 74, 116, 148, 184],
- combustion in pulverized-coal-fired boilers (in power plants) [62],
- greenhouses [60],
- hydraulic systems [12],
- solar power stations [9, 55],
- waste water treatment plants [131],
- electromagnetic mills [136],
- cement kilns [170].

Typically, the sampling period of industrial MPC algorithms used in process control is quite long, of the order of seconds, a dozens of seconds or even minutes. Programmable Logic Controllers (PLCs) are used for implementation of MPC algorithms in industrial process control. In addition to that, thanks to availability of fast microcontrollers, it is possible to develop MPC algorithms for fast dynamical systems (in embedded systems). In contrast to the mentioned industrial applications, they require short sampling times, shorter than one second, typically of millisecond order. Example applications of fast MPC include:

- fuel cells [59],
- active vibration attenuation [176],
- combustion engines [28, 73, 154],
- robots [183, 22, 139],
- servomotors [24],
- quadrotors [7],
- stratospheric airships [108],
- power converters [194],
- electrical inverters [110],
- induction machines [51].

Many research works are concerned with automotive applications. A few examples are: autonomous driving [105, 173], autonomous racing [6], traction control [68], vehicle roll-over [67].

There are some applications of MPC in medicine, e.g. muscle relaxant anaesthesia [114] and artificial pancreas [66].

In addition to industrial and embedded applications of MPC, it is interesting to mention a few original and less frequent applications in which MPC algorithms also turn out to be very efficient:

- drinking water transport networks [143],
- supermarket refrigeration systems [161],
- traffic on highways [14],
- high energy physics accelerators [18],
- inventory management in hospitals [113].

Important applications of MPC are concerned with building control. Typically, only temperature control (stabilisation despite changes of the outside temperature, which is a disturbance) is considered [56, 172]. In more advanced solutions, thermal comfort is controlled [197], i.e. temperature, humidity and other factors. MPC may cooperate with on-line energy optimisation which determines optimal set-points for MPC [10].

It is important to emphasise that all cited works in Chapter 1.5 discuss real applications only. In addition to that, hundreds or even thousands of works annually discuss simulation results.

References

1. Aggelogiannaki, E., Sarimveis, H.: A simulated annealing algorithm for prioritized multiobjective optimization–implementation in an adaptive model predictive control configuration. IEEE Transactions on Systems, Man and Cybernetics–Part B: Cybernetics **37**, 902–915 (2007)
2. Åkesson, B.M., Toivonen, H.T., Waller, J.B., Nyström, R.H.: Neural network approximation of a nonlinear model predictive controller applied to a pH neutralization process. Computers & Chemical Engineering **29**, 323–335 (2005)
3. Akpan, V.A., Hassapis, G.D.: Nonlinear model identification and adaptive model predictive control using neural networks. ISA Transactions **50**, 177–194 (2011)
4. Al-Duwaish, H., Karim, M., Chandrasekar, V.: Use of multilayer feedforward neural networks in identification and control of Wiener model. IEE Proceedings: Control Theory and Applications **143**, 255–258 (1996)
5. Al Seyab, R.K., Cao, Y.: Nonlinear model predictive control for the ALSTOM gasifier. Journal of Process Control **16**, 795–808 (2006)
6. Alcalá, E., Puig, V., Quevedo, J., Rosolia, U.: Autonomous racing using linear parameter varying-model predictive control (LPV-MPC). Control Engineering Practice **95**, 104270 (2020)
7. Alexis, K., Nikolakopoulos, G., Tzes, A.: Switching model predictive attitude control for a quadrotor helicopter subject to atmospheric disturbances. ISA Transactions **19**, 1195–1207 (2011)
8. Amos, B., Xu, L., Kolter, J.Z.: Input convex neural networks. In: Proceedings of the 34th International Conference on Machine Learning, pp. 146–155. Sydney, NSW, Australia (2017)
9. Arahal, M.R., M., B., F., C.E.: Neural identification applied to predictive control of a solar plant. Control Engineering Practice **6**, 333–344 (1998)
10. Ascione, F., Bianco, N., De Stasio, C., Mauro, G.M., Vanoli, G.P.: Simulation-based model predictive control by the multi-objective optimization of building energy performance and thermal comfort. Energy and Buildings **111**, 131–144 (2016)
11. Assandri, A.D., de Prada, C., Rueda, A., Martínez, J.S.: Nonlinear parametric predictive temperature control of a distillation column. Control Engineering Practice **21**, 1795–1806 (2013)
12. Bakhshande, F., Spiller, M., King, Y.L., Söffker, D.: Computationally efficient model predictive control for real time implementation experimentally applied on a hydraulic differential cylinder. IFAC-PapersOnLine **53**, 8979–8984 (2020)
13. Bartletta, R.A., Biegler, L.T., Backstromb, J., Gopal, V.: Quadratic programming algorithms for large-scale model predictive control. Journal of Process Control **12**, 775–795 (2002)
14. Bellemans, T., De Schutter, B., De Moor, B.: Model predictive control for ramp metering of motorway traffic: a case study. Control Engineering Practice **14**, 757–767 (2006)
15. Bemporad, A., Morari, M., Dua, V., Pistikopoulos, E.: The explicit linear quadratic regulator for constrained systems. Automatica **38**, 3–20 (2002)
16. Bemporad, A., Patrinos, P.: Simple and certifiable quadratic programming algorithms for embedded linear model predictive control. IFAC Proceedings Volumes **45**, 14–20 (2012)
17. Berenguel, M., Arahal, M.R., Camacho, E.F.: Modelling the free response of a solar plant for predictive control. Control Engineering Practice **6**, 1257–1266 (1998)
18. Blanco, E., de Prada, C., Cristea, S., Casas, J.: Nonlinear predictive control in the LHC accelerator. Control Engineering Practice **17**, 1136–1147 (2009)
19. Bosschaerts, W., Van Renterghem, T., Hasan, O.A., Limam, K.: Development of a model based predictive control system for heating buildings. Energy Procedia **122**, 519–528 (2017)
20. Byrd, R.H., Hribar, M.E., Nocedal, J.: An interior point algorithm for large-scale nonlinear programming. SIAM Journal on Optimization **9**, 877–900 (1999)
21. Cagienard, R., Grieder, P., Kerrigan, E.C., Morari, M.: Move blocking strategies in receding horizon control. In: Proceedings of the 43rd IEEE Conference on Decision and Control (CDC 2004), pp. 2023–2028. Nassau, Bahamas (2004)

22. Castañeda, L.Á., Guzman-Vargas L. Chairez, I., Luviano-Juárez, A.: Output based bilateral adaptive control of partially known robotic systems. Control Engineering Practice **98**, 104362 (2020)
23. Cervantes, A.L., Agamennoni, O.E., Figueroa, J.L.: A nonlinear model predictive control system based on Wiener piecewise linear models. Journal of Process Control **13**, 655–666 (2003)
24. Chaber, P., Ławryńczuk, M.: Fast analytical model predictive controllers and their implementation for STM32 ARM microcontroller. IEEE Transactions on Industrial Informatics **15**, 4580–4590 (2019)
25. Chen, L., Du, S., He, Y., Liang, M., Xu, D.: Robust model predictive control for greenhouse temperature based on particle swarm optimization. Information Processing in Agriculture **5**, 329–338 (2018)
26. Chen, Y., Shi, Y., Zhang, B.: Optimal control via neural networks: a convex approach. In: Proceedings of the International Conference on Learning Representations. New Orleans, USA (2019)
27. Clarke, D.W., Mohtadi, C., Tuffs, P.S.: Generalized predictive control–part i. the basic algorithm. Automatica **23**, 137–148 (1987)
28. Colin, G., Chamaillard, Y., Bloch, G., Corde, G.: Neural control of fast nonlinear systems–application to a turbocharged SI engine with VCT. IEEE Transactions on Neural Networks **18**, 1101–1114 (2007)
29. Cutler, C.R., Ramaker, B.L.: Dynamic matrix control–a computer control algorithm. In: Proceedings of the AIChE National Meeting. Houston, Texas, USA (1979)
30. D., D., D., C.: A practical multiple model adaptive strategy for single-loop MPC. Control Engineering Practice **11**, 141–159 (2003)
31. Deng, H., Ohtsuka, T.: A parallel newton-type method for nonlinear model predictive control. Automatica **109**, 108560 (2019)
32. Desaraju, V.R., Nathan, M.: Leveraging experience for computationally efficient adaptive nonlinear model predictive control. In: Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA 2017), pp. 5314–5320. Singapore (2017)
33. Diehl, M., Bock, H.G., Schlöder, J.P., Findeisen, R., Nagy, Z., Allgöwer, F.: Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations. Journal of Process Control **12**, 577–585 (2002)
34. Diehl, M., Ferreau, H.J., Haverbeke, N.: Efficient numerical methods for nonlinear mpc and moving horizon estimation. In: L. Magni, D.M. Raimondo, F. Allgöwer (eds.) Nonlinear model predictive control, *Lecture Notes in Control and Information Sciences*, vol. 384, pp. 391–417. Springer, Berlin, Heidelberg (2009)
35. Ding, B., Ping, X.: Dynamic output feedback model predictive control for nonlinear systems represented by Hammerstein-Wiener model. Journal of Process Control **22**, 1773–1784 (2012)
36. Domański, P.D.: Control Performance Assessment: Theoretical Analyses and Industrial Practice, *Studies in Systems, Decision and Control*, vol. 245. Springer, Cham (2020)
37. Domański, P.D.: Performance assessment of predictive control—A survey. Algorithms **13**, 97 (2020)
38. Domański, P.D., Ławryńczuk, M.: Assessment of predictive control performance using fractal measures. Nonlinear Dynamics **89**, 773–790 (2017)
39. Domański, P.D., Ławryńczuk, M.: Assessment of the GPC control quality using non-gaussian statistical measures. International Journal of Applied Mathematics and Computer Science **27**, 291–307 (2017)
40. Domański, P.D., Ławryńczuk, M.: Control quality assessment for processes with asymmetric properties and its application to pH reactor. IEEE Access **8**, 94535–94546 (2020)
41. Domański, P.D., Ławryńczuk, M.: Multi-criteria control performance assessment method for a multivariate MPC. In: Proceedings of the American Control Conference (ACC 2020), pp. 1968–1973. Denver, Colorado, USA (2020)

42. Domański, P.D., Ławryńczuk, M.: Quality assessment of nonlinear model predictive control using fractal and entropy measures. In: W. Lacarbonara, B. Balachandran, J. Ma, J. Tenreiro Machado, G. Stepan (eds.) Nonlinear Dynamics and Control, pp. 147–156. Springer, Cham (2020)
43. Domek, S.: Switched state model predictive control of fractional-order nonlinear discrete-time systems. Asian Journal of Control **15**, 658–668 (2013)
44. Domek, S.: Fractional-order model predictive control with small set of coincidence points. In: K. Latawiec, M. Łukaniszyn, R. Stanisławski (eds.) Advances in Modelling and Control of Non-integer-Order Systems, *Lecture Notes in Electrical Engineering*, vol. 320, pp. 135–144. Springer, Cham (2015)
45. Domek, S.: Model-plant mismatch in fractional order model predictive control. In: S. Domek, P. Dworak (eds.) Theoretical Developments and Applications of Non-Integer Order Systems, *Lecture Notes in Electrical Engineering*, vol. 357, pp. 281–291. Springer, Cham (2016)
46. Domek, S.: Switched fractional state-space predictive control methods for non-linear fractional systems. In: A.B. Malinowska, D. Mozyrska, Ł. Sajewski (eds.) Advances in Non-Integer Order Calculus and Its Applications, *Lecture Notes in Electrical Engineering*, vol. 3559, pp. 113–127. Springer, Cham (2020)
47. Doncevic, D.T., Schweidtmann, A.M., Vaupel, Y., Schäfer, P., Caspari, A., Mitsos, A.: Deterministic global nonlinear model predictive control with recurrent neural networks embedded. IFAC-PapersOnLine **53**, 5273–5278 (2020)
48. Ellis, M., Christofides, P.D.: On finite-time and infinite-time cost improvement of economic model predictive control for nonlinear systems. Automatica **50**, 2561–2569 (2014)
49. Ellis, M., Durand, H., Christofides, P.D.: A tutorial review of economic model predictive control methods. Journal of Process Control **24**, 1156–1178 (2014)
50. Engell, S.: Feedback control for optimal process operation. Journal of Process Control **17**, 203–219 (2007)
51. Englert, T., Graichen, K.: Nonlinear model predictive torque control and setpoint computation of induction machines for high performance applications. Control Engineering Practice **99**, 104415 (2016)
52. Ferreau, H.J., Kirches, C., Potschka, A., Bock, H.G., Diehl, M.: qpOASES: a parametric active-set algorithm for quadratic programming. Mathematical Programming Computation **6**, 327–363 (2014)
53. Frasch, J.V., Sager, S., Diehl, M.: A parallel quadratic programming method for dynamic optimization problems. Mathematical Programming Computatione **7**, 289–329 (2015)
54. Fruzzetti, K.P., Palazoğlu, A., A., M.K.: Nonlinear model predictive control using Hammerstein models. Journal of Process Control **7**, 31–41 (1997)
55. Gallego, A.J., Merello, G.M., Berenguel, M., Camacho, E.F.: Gain-scheduling model predictive control of a Fresnel collector field. Control Engineering Practice **82**, 1–13 (2019)
56. Gorni, D., del Mar Castilla, M., Visioli, A.: An efficient modelling for temperature control of residential buildings. Building and Environment **103**, 86–98 (2016)
57. Grancharova, A., Johansen, T.A.: Explicit Nonlinear Model Predictive Control, *Lecture Notes in Control and Information Sciences*, vol. 429. Springer, Berlin (2012)
58. Griffith, D.W., Biegler, L.T., Patwardhan, S.C.: Robustly stable adaptive horizon nonlinear model predictive control. Journal of Process Control **70**, 109–122 (2018)
59. Gruber, J.K., Doll, M., Bordons, C.: Design and experimental validation of a constrained mpc for the air feed of a fuel cell. Control Engineering Practice **17**, 874–885 (2009)
60. Gruber, J.K., Guzmán, J.L., Rodríguez, F., Bordons, C., Berenguel, M., Sánchez, J.A.: Nonlinear mpc based on a Volterra series model for greenhouse temperature control using natural ventilation. Control Engineering Practice **19**, 354–366 (2011)
61. Gutiérrez-Urquídez, R.C., Valencia-Palomo, G., Rodríguez-Elias, O.M., Trujillo, L.: Systematic selection of tuning parameters for efficient predictive controllers using a multiobjective evolutionary algorithm. Applied Soft Computing **31**, 326–338 (2015)
62. Havlena, V., Findejs, J.: Application of model predictive control to advanced combustion control. Control Engineering Practice **13**, 671–680 (2005)

63. Hong, M., Cheng, S.: Hammerstein-Wiener model predictive control of continuous stirred tank reactor. In: W. Hu (ed.) Electronics and Signal Processing, *Lecture Notes in Electric Engineering*, vol. 97, pp. 235–242. Springer, Berlin, Heidelberg (2011)
64. Hosen, M.A., Hussain, M.A., Mjalli, F.S.: Control of polystyrene batch reactors using neural network based model predictive control (NNMPC): An experimental investigation. Control Engineering Practice **19**, 454–467 (2011)
65. Huyck, B., De Brabanter, J., De Moor, B., Van Impe, J.F., Logist, F.: Online model predictive control of industrial processes using low level control hardware: A pilot-scale distillation column case study. Control Engineering Practice **28**, 34–48 (2014)
66. Incremona, G.P., Messori, M., Toffanin, C., Cobelli, C., Magni, L.: Model predictive control with integral action for artificial pancreas. Control Engineering Practice **77**, 86–94 (2019)
67. Jalali, M., Hashemi, E., Khajepour, A., Chen, S.K., Litkouhi, B.: Model predictive control of vehicle roll-over with experimental verification. Control Engineering Practice **77**, 256–266 (2018)
68. Jalali, M., Khajepour, A., Chen, S.K., Litkouhi, B.: Integrated stability and traction control for electric vehicles using model predictive control. Control Engineering Practice **54**, 256–266 (2016)
69. Jama, M., Wahyudie, A., Noura, H.: Robust predictive control for heaving wave energy converters. Control Engineering Practice **77**, 138–149 (2018)
70. Jia, L., Li, Y., Li, F.: Correlation analysis algorithm-based multiple-input single-output Wiener model with output noise. Complexity p. 9650254 (2019)
71. Johansen, T.A.: Approximate explicit receding horizon control of constrained nonlinear systems. Automatica **40**, 293–300 (2004)
72. Jungers, R.M., Tabuada, P.: Non-local linearization of nonlinear differential equations via *polyflows*. In: Proceedings of the American Control Conference (ACC 2019), pp. 1906–1911. Philadelphia, Pensylwania, USA (2019)
73. Kaleli, A.: Development of the predictive based control of an autonomous engine cooling system for variable engine operating conditions in SI engines: design, modeling and real-time application. Control Engineering Practice **100**, 104424 (2020)
74. Kawathekar, R., Riggs, J.B.: Nonlinear model predictive control of a reactive distillation column. Control Engineering Practice **15**, 231–239 (2007)
75. Khan, B., Rossiter, J.A.: Alternative parameterisation within predictive control: a systematic selection. International Journal of Control **86**, 1397–1409 (2013)
76. Kim, J., Jung, Y., Bang, H.: Linear time-varying model predictive control of magnetically actuated satellites in elliptic orbits. Acta Astronautica **151**, 791–804 (2018)
77. Klaučo, M., Kalúz, M., Kvasnica, M.: Machine learning-based warm starting of active set methods in embedded model predictive control. Engineering Applications of Artificial Intelligence **77**, 1–8 (2019)
78. Kögel, M., Findeisen, R.: A fast gradient method for embedded linear predictive control. IFAC Proceedings Volumes **44**, 1362–1367 (2011)
79. Koopman, B.: Hamiltonian systems and transformation in Hilbert space. Proceedings of the National Academy of Sciences of the United States of America **17**, 315–318 (1931)
80. Koopman, B., von Neuman, J.: Dynamical systems of continuous spectra. Proceedings of the National Academy of Sciences of the United States of America **18**, 255–263 (1932)
81. Korbicz, J., Kościelny, J.M., Kowalczuk, Z.: Fault diagnosis: models, artificial intelligence, applications. Springer, Heidelberg (2004)
82. Korda, M., Mezić, I.: Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control. Automatica **93**, 149–160 (2018)
83. Kościelny, J.M.: Fault Diagnosis of Automated Industrial Processes. Academic Publishing House EXIT, Warsaw (2001). In Polish
84. Lasheen, A., Saad, M.S., Emara, H.M., Elshafei, A.L.: Continuous-time tube-based explicit model predictive control for collective pitching of wind turbine. Energy **118**, 1222–1233 (2017)

85. Ławryńczuk, M.: A family of model predictive control algorithms with artificial neural networks. International Journal of Applied Mathematics and Computer Science **17**, 217–232 (2007)
86. Ławryńczuk, M.: Neural Dynamic Matrix Control algorithm with disturbance compensation. In: N. García Pedrajas, F. Herrera, C. Fyfe, J.M. Benítez, A. M. (eds.) Proceedings of the 23th International Conference on Industrial, Engineering & Other Applications of Applied Intelligent Systems (IEA-AIE 2010), Cordoba, Spain, *Lecture Notes in Artificial Intelligence*, vol. 6098, pp. 52–61. Springer, Berlin (2010)
87. Ławryńczuk, M.: Nonlinear predictive control based on multivariable neural Wiener models. In: A. Dobnikar, U. Lotrič, B. Šter (eds.) Proceedings of the 10th International Conference on Adaptive and Natural Computing Algorithms (ICANNGA 2011), *Lecture Notes in Computer Science*, vol. 6593, pp. 31–40. Springer, Berlin (2011)
88. Ławryńczuk, M.: On improving accuracy of computationally efficient nonlinear predictive control based on neural models. Computers Engineering Science **66**, 5253–5267 (2011)
89. Ławryńczuk, M.: On-line set-point optimisation and predictive control using neural Hammerstein models. Chemical Engineering Journal **166**, 269–287 (2011)
90. Ławryńczuk, M.: Predictive control of a distillation column using a control-oriented neural model. In: A. Dobnikar, U. Lotrič, B. Šter (eds.) Proceedings of the 10th International Conference on Adaptive and Natural Computing Algorithms (ICANNGA 2011), *Lecture Notes in Computer Science*, vol. 6593, pp. 230–239. Springer, Berlin (2011)
91. Ławryńczuk, M.: Computationally Efficient Model Predictive Control Algorithms: a Neural Network Approach, *Studies in Systems, Decision and Control*, vol. 3. Springer, Cham (2014)
92. Ławryńczuk, M.: Explicit nonlinear predictive control algorithms with neural approximation. Neurocomputing **129**, 570–584 (2014)
93. Ławryńczuk, M.: Nonlinear predictive control for Hammerstein-Wiener systems. ISA Transactions **55**, 49–62 (2015)
94. Ławryńczuk, M.: Modelling and predictive control of a neutralisation reactor using sparse support vector machine Wiener models. Neurocomputing **205**, 311–328 (2016)
95. Ławryńczuk, M.: Nonlinear predictive control of dynamic systems represented by Wiener-Hammerstein models. Nonlinear Dynamics **86**, 1193–1214 (2016)
96. Ławryńczuk, M.: Nonlinear predictive control of a boiler-turbine unit: A state-space approach with successive on-line model linearisation and quadratic optimisation. ISA Transactions **67**, 476–495 (2017)
97. Ławryńczuk, M.: Constrained computationally efficient nonlinear predictive control of Solid Oxide Fuel Cell: Tuning, feasibility and performance. ISA Transactions **99**, 270–289 (2020)
98. Ławryńczuk, M.: Nonlinear model predictive control for processes with complex dynamics: a parameterisation approach using Laguerre functions. International Journal of Applied Mathematics and Computer Science **30**, 35–46 (2020)
99. Ławryńczuk, M., Ocłoń, P.: Model Predictive Control and energy optimisation in residential building with electric underfloor heating system. Energy **182**, 1028–1044 (2019)
100. Ławryńczuk, M., Söffker, D.: Wiener structures for modeling and nonlinear predictive control of proton exchange membrane fuel cell. Nonlinear Dynamics **95**, 1639–1660 (2019)
101. Ławryńczuk, M., Tatjewski, P.: Offset-free state-space nonlinear predictive control for Wiener systems. Information Sciences **511**, 127–151 (2020)
102. Li, S.E., Jia, Z., Li, K., Cheng, B.: Fast online computation of a model predictive controller and its application to fuel economy-oriented adaptive cruise control. IEEE Transactions on Industrial Informatics **16**, 1199–1209 (2015)
103. Li, Y., Shen, J., Lu, J.: Constrained model predictive control of a solid oxide fuel cell based on genetic optimization. Journal of Power Sources **196**, 5873–5880 (2011)
104. Ligthart, J.A.J., Poksawat, P., Wang, L., Nijmeijer, H.: Experimentally validated model predictive controller for a hexacopter. IFAC-PapersOnLine **50**, 4076–4081 (2017)
105. Lima, P.F., Pereira, G.C., Mårtensson, J., Wahlberg, B.: Experimental validation of model predictive control stability for autonomous driving. Control Engineering Practice **81**, 244–255 (2018)

106. Liu, G.P., Kadirkamanathan, V., Billings, S.A.: Predictive control for non-linear systems using neural networks. International Journal of Control **71**, 1119–1132 (1998)
107. Liu, S., Liu, J.: Economic model predictive control with extended horizon. Automatica **73**, 180–192 (2016)
108. Liu, S., Sang, Y., Jin, H.: Robust model predictive control for stratospheric airships using LPV design. Control Engineering Practice **81**, 231–243 (2018)
109. Liu, S., Wang, J.: A simplified dual neural network for quadratic programming with its KWTA application. IEEE Transactions on Neural Networks **17**, 1500–1510 (2006)
110. Liu, Y., Ge, B., Abu-Rub, H., Sun, H., Peng, F.Z., Xue, Y.: Model predictive direct power control for active power decoupled single-phase quasi-Z-source inverter. IEEE Transactions on Industrial Informatics **12**, 1550–1559 (2016)
111. Lopez-Negrete, R., D'Amato, F.J., Biegler, L.T., Kumar, A.: Fast nonlinear model predictive control: Formulation and industrial process applications. Computers & Chemical Engineering **51**, 55–64 (2013)
112. Maciejowski, J.: Predictive control with constraints. Prentice Hall, Harlow (2002)
113. Maestre, J.M., Fernández, M.I., Jurado, I.: An application of economic model predictive control to inventory management in hospitals. Control Engineering Practice **71**, 120–128 (2018)
114. Mahfouf, M., Linkens, D.A.: Non-linear generalized predictive control (NLGPC) applied to muscle relaxant anaesthesia. International Journal of Control **71**, 239–257 (1998)
115. Makarow, A., Keller, M., Rösmann, C., Bertram, T.: Model predictive trajectory set control with adaptive input domain discretization. In: Proceedings of the American Control Conference (ACC 2018), pp. 3159–3164. Milwaukee, USA (2018)
116. Martin, P.A., Odloak, D., Kassab, F.: Robust model predictive control of a pilot plant distillation column. Control Engineering Practice **21**, 231–241 (2013)
117. Martins, M.A.F., Odloak, D.: A robustly stabilizing model predictive control strategy of stable and unstable processes. Automatica **67**, 132–143 (2016)
118. Marusak, P.M.: Oeasily reconfigurable analytical fuzzy predictive controllers: Actuator faults handling. In: L. Kang, Z. Cai, X. Yan, Y. Liu (eds.) Advances in Computation and Intelligence, *Lecture Notes in Computer Science*, vol. 5370, pp. 396–405. Springer, Berlin, Heidelberg (2008)
119. Marusak, P.M.: Advantages of an easy to design fuzzy predictive algorithm in control systems of nonlinear chemical reactors. Applied Soft Computing **9**, 1111–1125 (2009)
120. Marusak, P.M.: Application of fuzzy Wiener models in efficient MPC algorithms. In: M. Szczuka, M. Kryszkiewicz, S. Ramanna, R. Jensen, Q. Hu (eds.) Rough Sets and Current Trends in Computing, *Lecture Notes in Artificial Intelligence*, vol. 6086, pp. 669–677. Springer, Berlin, Heidelberg (2010)
121. Marusak, P.M.: On prediction generation in efficient MPC algorithms based on fuzzy Hammerstein models. In: L. Rutkowski, R. Scherer, R. Tadeusiewicz, L.A. Zadeh, J.M. Zurada (eds.) Artificial Intelligence and Soft Computing, *Lecture Notes in Computer Science*, vol. 6113, pp. 136–143. Springer, Berlin, Heidelberg (2010)
122. Marusak, P.M.: Efficient MPC algorithms based on fuzzy Wiener models and advanced methods of prediction generation. In: L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L.A. Zadeh, J.M. Zurada (eds.) Artificial Intelligence and Soft Computing, *Lecture Notes in Computer Science*, vol. 7267, pp. 292–300. Springer, Berlin, Heidelberg (2012)
123. Marusak, P.M.: Numerically efficient fuzzy MPC algorithm with advanced generation of prediction—application to a chemical reactor. Algorithms **13**, 143 (2020)
124. Marusak, P.M.: Advanced construction of the dynamic matrix in numerically efficient fuzzy MPC algorithms. Algorithms **14**, 25 (2021)
125. Marusak, P.M.: A numerically efficient fuzzy MPC algorithms with fast generation of the control signal. International Journal of Applied Mathematics and Computer Science **31**, 59–71 (2021)
126. Mattingley, J., Boyd, S.: CVXGEN: a code generator for embedded convex optimization. Optimization and Engineering **13**, 1–27 (2012)

127. Mauroy, A., Mezić, I., Susuki, Y. (eds.): The Koopman Operator in Systems and Control: Concepts, Methodologies, and Applications, *Lecture Notes in Control and Information Sciences*, vol. 484. Springer, Cham (2020)
128. Mayne, D.Q.: Model predictive control: Recent developments and future promise. Automatica **50**, 2967–2986 (2014)
129. Mayne, D.Q., Rawlings, J.B., Rao, C.V., Scokaert, P.O.M.: Constrained model predictive control: Stability and optimality. Automatica **36**, 789–814 (2000)
130. Mu, J., Rees, D., Liu, G.P.: Advanced controller design for aircraft gas turbine engines. Control Engineering Practice **13**, 1001–1015 (2005)
131. Mulas, M., Tronci, S., Corona, F., Haimi, H., Lindell, P., Heinonen, M., Vahala, R., Baratti, R.: Predictive control of an activated sludge process: An application to the Viikinmäki wastewater treatment plant. Control Engineering Practice **35**, 89–100 (2015)
132. Müller, M.A., Grüne, L.: Economic model predictive control without terminal constraints for optimal periodic behavior. Automatica **70**, 128–139 (2016)
133. Norquay, S.J., Palazoğlu, A., Romagnoli, J.A.: Model predictive control based on Wiener models. Chemical Engineering Science **53**, 75–84 (2016)
134. Norquay, S.J., Palazoğlu, A., Romagnoli, J.: Application of wiener model predictive control (WMPC) to an industrial C2 splitter. Journal of Process Control **9**, 461–473 (1999)
135. Ntouskas, S., Sarimveis, H., Sopasakis, P.: Model predictive control for offset-free reference tracking of fractional order systems. Control Engineering Practice **71**, 26–33 (2018)
136. Ogonowski, S., Bismor, D., Ogonowski, Z.: Control of complex dynamic nonlinear loading process for electromagnetic mill. Archives of Control Sciences **30**, 471–500 (2020)
137. Oliveira, G.H.C., da Rosa, A., Campello, R.J.G.B., Machado, J.B., Amaral, W.C.: An introduction to models based on Laguerre, Kautz and other related orthonormal functions – part I: linear and uncertain models. International Journal of Modelling, Identification and Control **14**, 121–132 (2011)
138. Oliveira, G.H.C., da Rosa, A., Campello, R.J.G.B., Machado, J.B., Amaral, W.C.: An introduction to models based on Laguerre, Kautz and other related orthonormal functions – part II: Non-linear models. International Journal of Modelling, Identification and Control **16**, 1–14 (2012)
139. Ortega, J.G., Camacho, E.F.: Mobile robot navigation in a partially structured static environment, using neural predictive control. Control Engineering Practice **4**, 1669–1679 (1996)
140. Pan, Y., Wang, J.: Nonlinear model predictive control using a recurrent neural network. In: Proceedings of the International Joint Conference on Neural Networks (IJCNN 2008), pp. 2296–2301. Hong Kong (2008)
141. Pan, Y., Wang, J.: Two neural network approaches to model predictive control. In: Proceedings of the American Control Conference (ACC 2008), pp. 1685–1690. Washington, USA (2008)
142. Parisini, T., Zoppoli, R.: A receding-horizon regulator for nonlinear systems and a neural approximation. Automatica **31**, 1443–1451 (1995)
143. Pascual, J., Romera, J., Puig, V., Cembrano, G., Creus, R., Minoves, M.: Operational predictive optimal control of Barcelona water transport network. Control Engineering Practice **21**, 1020–1034 (2013)
144. Patan, K.: Two stage neural network modelling for robust model predictive control. ISA Transactions **72**, 56–65 (2018)
145. Patan, K.: Robust and Fault-Tolerant Control: Neural-Network-Based Solutions, *Studies in Systems, Decision and Control*, vol. 197. Springer, Cham (2019)
146. Patan, K., Korbicz, J.: Nonlinear model predictive control of a boiler unit: a fault tolerant control study. International Journal of Applied Mathematics and Computer Science **22**, 225–237 (2012)
147. Patikirikorala, T., Wang, L., Colman, A., Han, J.: Hammerstein-Wiener nonlinear model based predictive control for relative QoS performance and resource management of software systems. Control Engineering Practice **20**, 49–61 (2012)
148. Porfírio, C., Odloak, D.: Optimizing model predictive control of an industrial distillation column. Control Engineering Practice **19**, 1137–1146 (2011)

149. Potočnik, P., Grabec, I.: Nonlinear model predictive control of a cutting process. Neurocomputing **43**, 107–126 (2002)
150. Pour, F.K., Puig, V., Ocampo-Martinez, C.: Multi-layer health-aware economic predictive control of a pasteurization pilot plant. International Journal of Applied Mathematics and Computer Science **28**, 97–110 (2018)
151. Powell, M.J.D.: A fast algorithm for nonlinearly constrained optimization calculations. In: G.A. Watson (ed.) Numerical Analysis, *Lecture Notes in Mathematics*, vol. 630, pp. 144–157. Springer, Dundee (1978)
152. Qin, S.J., Badgwell, T.A.: A survey of industrial model predictive control technology. Control Engineering Practice **11**, 733–764 (2003)
153. Rao, C.V., Wright, S.J., Rawlings, J.B.: Application of interior-point methods to model predictive control. Journal of Optimization Theory and Applications **99**, 723–757 (1998)
154. Raut, A., Irdmousa, B.K., Shahbakhti, M.: Dynamic modeling and model predictive control of an rcci engine. Control Engineering Practice **81**, 129–144 (2018)
155. Reese, B.M., Collins, E.G.: A graph search and neural network approach to adaptive nonlinear model predictive control. Engineering Applications of Artificial Intelligence **55**, 250–268 (2016)
156. Richalet, J., O'Donovan, D.: Predictive Functional Control: Principles and Industrial Applications. Springer, London (2009)
157. Richalet, J.A., Rault, A., Testud, J.L., Papon, J.: Model predictive heuristic control: application to an industrial processes. In: Proceedings of the AIChE National Meeting, vol. 14, pp. 413–428 (1979)
158. Richter, S., Morari, M., Jones, C.N.: Towards computational complexity certification for constrained MPC based on Lagrange relaxation and the fast gradient method. In: Proceedings of the 2011 IEEE 50th Annual Conference on Decision and Control (CDC) and European Control Conference (ECC), pp. 5223–5229. Orlando, Florida, USA (2011)
159. Rodrigues, M.A., Odloak, D.: An infinite horizon model predictive control for stable and integrating processes. Computers & Chemical Engineering **27**, 1113–1128 (2003)
160. Saeed, J., Hasan, A.: Unit prediction horizon binary search-based model predictive control of full-bridge DC-DC converter. IEEE Transactions on Control Systems Technology **26**, 463–474 (2018)
161. Sarabia, D., Capraro, F., Larsen, L.F.S., de Prada, C.: Hybrid NMPC of supermarket display cases. Control Engineering Practice **17**, 428–441 (2009)
162. Saraswati, S., Chand, S.: Online linearization-based neural predictive control of air-fuel ratio in SI engines with PID feedback correction scheme. Neural Computing and Applications **19**, 919–933 (2010)
163. Scattolini, R.: Architectures for distributed and hierarchical model predictive control – a review. Journal of Process Control **19**, 723–731 (2009)
164. Schweidtmann, A.M., Mitsos, A.: Deterministic global optimization with artificial neural networks embedded. Journal of Optimization Theory and Applications **180**, 925–948 (2019)
165. Scokaert, P.O.M., Mayne, D.Q., Rawlings, J.B.: Suboptimal model predictive control (feasibility implies stability). IEEE Transactions on Automatic Control **44**, 648–654 (1999)
166. Seki, H., Ogawa, M., Ooyama, S., Akamatsu, K., Ohshima, M., Yang, W.: Industrial application of a nonlinear model predictive control to polymerization reactors. Control Engineering Practice **9**, 819–828 (2001)
167. Seybold, L., Witczak, M., Majdziek, P., Stetter, R.: Towards robust predictive fault-tolerant control for a battery assembly unit. International Journal of Applied Mathematics and Computer Science **25**, 849–862 (2015)
168. Shafiee, G., M., A.M., Jahed-Motlagh, M.R., Jalali, A.A.: Nonlinear predictive control of a polymerization reactor based on piecewise linear Wiener model. Chemical Engineering Journal **143**, 282–292 (2008)
169. Sopasakis, P., Sarimveis, H.: Stabilising model predictive control for discrete-time fractional-order systems. Automatica **75**, 24–31 (2017)
170. Stadler, K.S., Poland, J., Gallestey, E.: Model predictive control of a rotary cement kiln. Control Engineering Practice **19**, 1–9 (2011)

171. Stellato, B., Banjac, G., Goulart, P., Bemporad, A., Boyd, S.: OSQP: an operator splitting solver for quadratic programs. Mathematical Programming Computation (2020). In press
172. Sturzenegger, D., Gyalistras, D., Morari, M., Smith, R.S.: Model predictive climate control of a Swiss office building: implementation, results, and cost–benefit analysis. IEEE Transactions on Control system technology **24**, 1–12 (2016)
173. Suh, J., Yi, K., Jung, J., Lee, K., Chong, H., Ko, B.: Design and evaluation of a model predictive vehicle control algorithm for automated driving using a vehicle traffic simulator. Control Engineering Practice **51**, 256–266 (2016)
174. Sun, J., Kolmanovsky, I.V., Ghaemi, R., Chen, S.: A stable block model predictive control with variable implementation horizon. Automatica **43**, 1945–1953 (2007)
175. Tahir, F., Mercer, E., Lowdon, I., Lovett, D.: Advanced process control and monitoring of a continuous flow micro-reactor. Control Engineering Practice **77**, 225–234 (2018)
176. Takács, G., Batista, G., Gulan, M., Rohal'-Ilkiv, B.: Embedded explicit model predictive vibration control. Mechatronics **36**, 54–62 (2016)
177. Tatjewski, P.: Advanced control of industrial processes, structures and algorithms. Springer, London (2007)
178. Tatjewski, P.: DMC algorithm with Laguerre functions. In: A. Bartoszewicz, J. Kabziński, J. Kacprzyk (eds.) Advanced, Contemporary Control, *Advances in Intelligent Systems and Computing*, vol. 1196, pp. 1006–1017. Springer, Cham (2020)
179. Tøndel, P., Johansen, T.A., Bemporad, A.: An algorithm for multi-parametric quadratic programming and explicit mpc solutions. Automatica **39**, 489–497 (2003)
180. Vaupel, Y., Hamacher, N.C., Caspari, A., Mhamdi, A., Kevrekidis, I.G., Mitsos, A.: Accelerating nonlinear model predictive control through machine learning. Journal of Process Control **92**, 261–270 (2020)
181. Vega, P., Revollar, S., Francisco, M., Martın, J.M.: Integration of set point optimization techniques into nonlinear mpc for improving the operation of WWTPs. Computers & Chemical Engineering **68**, 78–95 (2014)
182. Vermillion, C., Menezes, A., Kolmanovsky, I.: Stable hierarchical model predictive control using an inner loop reference model and λ-contractive terminal constraint sets. Automatica **50**, 92–99 (2014)
183. Vivas, A., Poignet, P.: Predictive functional control of a parallel robot. Control Engineering Practice **13**, 863–874 (2005)
184. Volk, U., Kniese, D.W., Hahn, R., Haber, R., Schmitz, U.: Optimized multivariable predictive control of an industrial distillation column considering hard and soft constraints. Control Engineering Practice **13**, 913–927 (2005)
185. Wahlberg, B.: System identification using Laguerre models. IEEE Transactions on Automatic Control **36**, 551–562 (1991)
186. Wang, L.: Continuous time model predictive control design using orthonormal functions. International Journal of Control **74**, 1588–1600 (2001)
187. Wang, L.: Discrete model predictive controller design using Laguerre functions. Journal of Process Control **14**, 131–142 (2004)
188. Wang, L.X., Wan, F.: Structured neural networks for constrained model predictive control. Automatica **37**, 1235–1243 (2001)
189. Wang, X., Mahalec, V., F., Q.: Globally optimal nonlinear model predictive control based on multi-parametric disaggregation. Journal of Process Control **52**, 1–13 (2017)
190. Wang, Y., Boyd, S.: Fast model predictive control using online optimization. IEEE Transactions on Control Systems Technology **18**, 267–278 (2010)
191. Wang, Y., Luo, L., Zhang, F., Wang, S.: GPU-based model predictive control for continuous casting spray cooling control system using particle swarm optimization. Control Engineering Practice **84**, 349–364 (2019)
192. Witczak, M.: Fault Diagnosis and Fault-Tolerant Control Strategies for Non-Linear Systems: Analytical and Soft Computing Approaches, *Lecture Notes in Electrical Engineering*, vol. 266. Springer, Cham (2014)
193. Wu, X., Zhu, X., Cao, G., Tu, H.: Predictive control of sofc based on a GA-RBF neural network model. Journal of Power Sources **179**, 232–239 (2008)

194. Xia, C., Liu, T., Shi, T., Song, Z.: A simplified finite-control-set model-predictive control for power converters. IEEE Transactions on Industrial Informatics **10**, 991–1002 (2014)
195. Yang, J., Li, X., Mou, H., Jian, L.: Predictive control of solid oxide fuel cell based on an improved takagi-sugeno fuzzy model. Journal of Power Sources **193**, 699–705 (2009)
196. Yang, S., Bequette, B.W.: Optimization-based control using input convex neural networks. Computers & Chemical Engineering **144**, 107143 (2020)
197. Yang, S., Wan, M.P., Ng, B.F., Zhang, T., Babu, S., Zhang, Z., Chen, W., Dubey, S.: A state-space thermal model incorporating humidity and thermal comfort for model predictive control in buildings. Energy and Buildings **170**, 25–39 (2018)
198. Yu, D.L., Gomm, J.B.: Implementation of neural network predictive control to a multivariable chemical reactor. Control Engineering Practice **11**, 1315–1323 (2003)
199. Yu, Z., Biegler, L.T.: Advanced-step multistage nonlinear model predictive control: robustness and stability. Journal of Process Control **85**, 15–29 (2020)
200. Zhang, J., Chin, K.S., Ławryńczuk, M.: Multilinear model decomposition and predictive dontrol of MIMO two-block cascade systems. Industrial & Engineering Chemistry Research **56**, 14101–14114 (2017)
201. Zheng, A.: A computationally efficient nonlinear MPC algorithm. In: Proceedings of the American Control Conference (ACC 1997), pp. 1623–1627. Albuquerque, New Mexico, USA (1997)
202. Zheng, Y., Zhou, J., Xu, Y., Zhang, Y., Qian, Z.: A distributed model predictive control based load frequency control scheme for multi-area interconnected power system using discrete-time Laguerre functions. ISA Transactions **68**, 127–140 (2017)
203. Zhou, F., Peng, H., Zeng, X., Tian, X., Peng, X.: RBF-ARX model-based robust MPC for nonlinear systems with unknown and bounded disturbance. Journal of the Franklin Institute **354**, 8072–8093 (2017)
204. Zhou, F., Peng, H., Zhang, G., Zeng, X.: A robust controller design method based on parameter variation rate of RBF-ARX model. IEEE Access **7**, 160284–160294 (2019)
205. Zhou, F., Peng, H., Zhang, G., Zeng, X., Peng, X.: Robust predictive control algorithm based on parameter variation rate information of functional-coefficient ARX model. IEEE Access **7**, 27231–27243 (2019)