

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

$$u(t) = u_0 + K \left(e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right) \quad (1.1)$$

The control error is defined as the difference between the set-point and the current measured value of the controlled variable, i.e. $e(t) = y^{\text{sp}}(t) - y(t)$. 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

$$u(k) = u(k-1) + r_0 e(k) + r_1 e(k-1) + r_2 e(k-2) \quad (1.2)$$

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 , $k = 1, 2, 3, \dots$, the vector of the future increments of the manipulated variable

$$\Delta \mathbf{u}(k) = \begin{bmatrix} \Delta u(k|k) \\ \vdots \\ \Delta u(k + N_u - 1|k) \end{bmatrix} \quad (1.3)$$

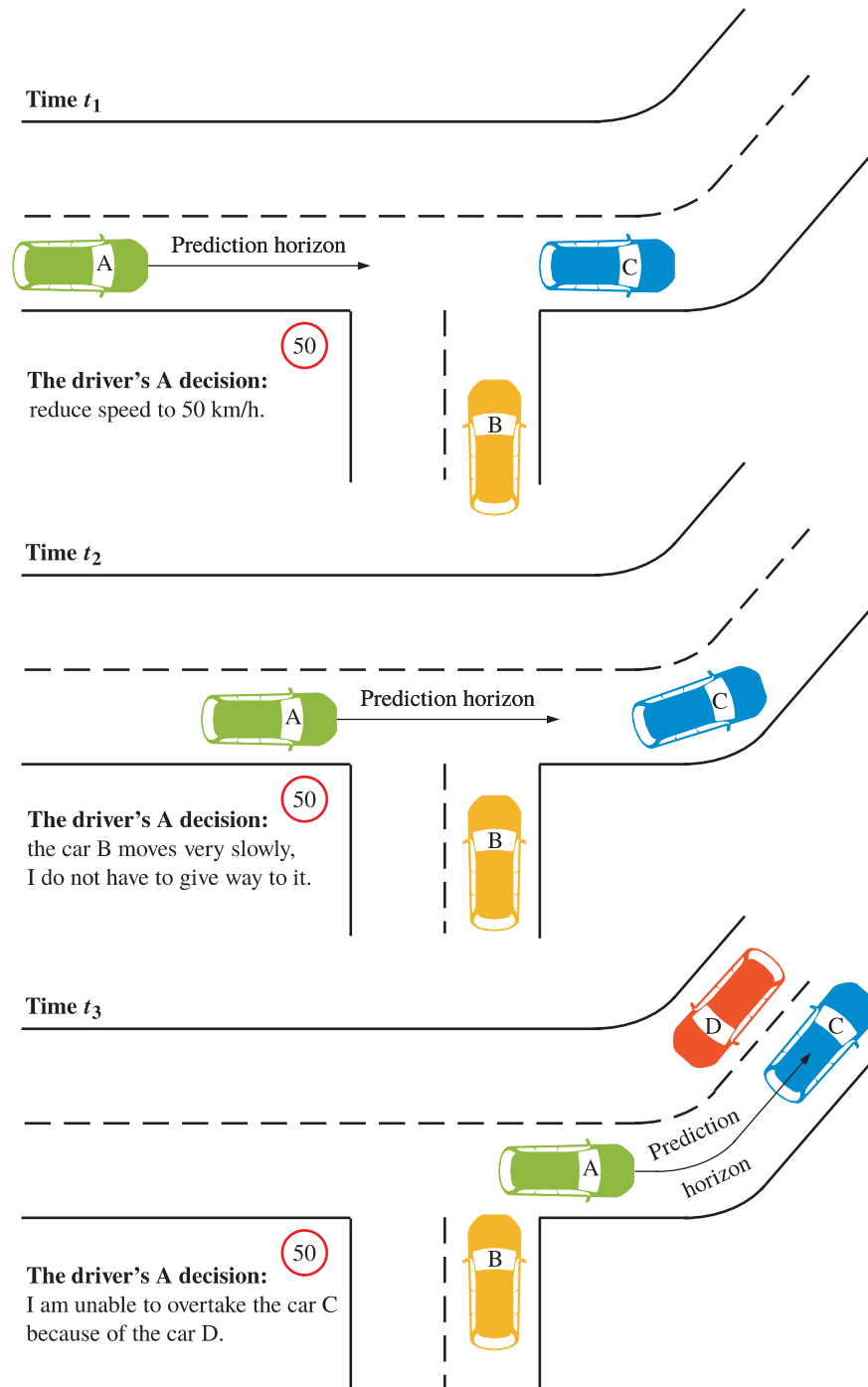


Fig. 1.1 Situations on the road and the driver's A decisions for three example time instants t_1 , t_2 , t_3

is calculated on-line. The symbol $\Delta u(k + p|k)$ 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

$$\Delta u(k|k) = u(k|k) - u(k - 1) \quad (1.4)$$

and the following ones are

$$\Delta u(k + p|k) = u(k + p|k) - u(k + p - 1|k) \quad (1.5)$$

for $p = 1, \dots, N_u - 1$. 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

$$\min_{\Delta \mathbf{u}(k)} \{J(k)\} \quad (1.6)$$

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

$$J(k) = \sum_{p=1}^N (y^{\text{sp}}(k + p|k) - \hat{y}(k + p|k))^2 + \lambda \sum_{p=0}^{N_u-1} (\Delta u(k + p|k))^2 \quad (1.7)$$

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 $N \geq N_u$ are taken into account. The set-point value for the sampling instant $k + p$ known at the current sampling instant k is denoted by $y^{\text{sp}}(k + p|k)$, 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 $y^{\text{sp}}(k)$, is used, i.e. $y^{\text{sp}}(k + 1|k) = \dots = y^{\text{sp}}(k + N|k) = y^{\text{sp}}(k)$. 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 $u(k + p|k) = u(k + N_u - 1|k)$ for $p = N_u, \dots, N$, which means that $\Delta u(k + N_u|k) = \dots = \Delta u(k + N|k) = 0$.

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 .

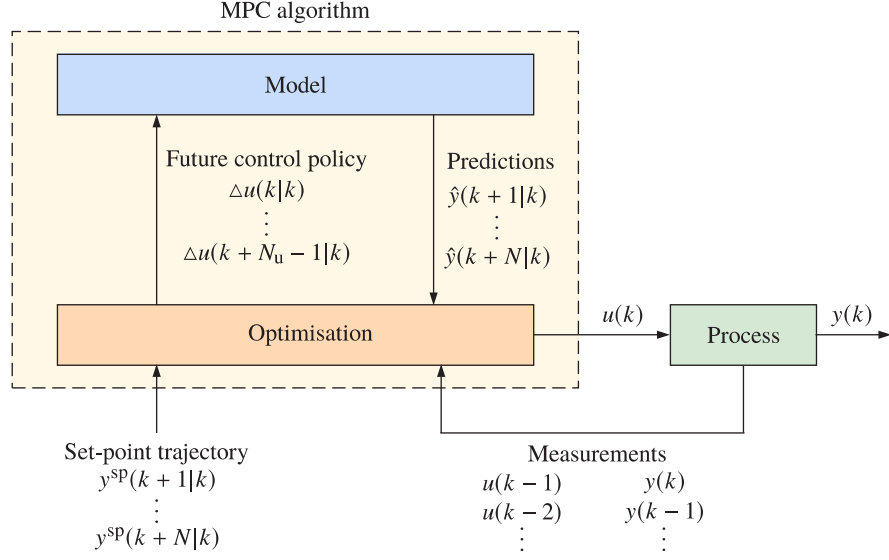


Fig. 1.2 The general structure of the MPC algorithm

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

$$u(k) = \Delta u^{\text{opt}}(k|k) + u(k-1) \quad (1.8)$$

where $\Delta u^{\text{opt}}(k|k)$ is the first element of the vector $\Delta \mathbf{u}^{\text{opt}}(k)$. 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

$$u^{\min} \leq u(k+p|k) \leq u^{\max}, \quad p = 0, \dots, N_u - 1 \quad (1.9)$$

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

$$\Delta u^{\min} \leq \Delta u(k+p|k) \leq \Delta u^{\max}, \quad p = 0, \dots, N_u - 1 \quad (1.10)$$

where $\Delta u^{\min}$ and $\Delta u^{\max}$ are the maximal negative and maximal (positive) changes of the manipulated variable, respectively (usually $\Delta u^{\min} = -\Delta u^{\max}$). 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. $\Delta u(k|k)$. Thirdly, the predicted values of the process output variable may also be limited, which is usually enforced by some technological reasons

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

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

$$\begin{aligned} \min_{\Delta u(k)} & \left\{ J(k) = \sum_{p=1}^N (y^{\text{sp}}(k+p|k) - \hat{y}(k+p|k))^2 + \lambda \sum_{p=0}^{N_u-1} (\Delta u(k+p|k))^2 \right\} \\ \text{subject to} & \\ & u^{\min} \leq u(k+p|k) \leq u^{\max}, \quad p = 0, \dots, N_u - 1 \\ & \Delta u^{\min} \leq \Delta u(k+p|k) \leq \Delta u^{\max}, \quad p = 0, \dots, N_u - 1 \\ & y^{\min} \leq \hat{y}(k+p|k) \leq y^{\max}, \quad p = 1, \dots, N \end{aligned} \quad (1.12)$$

The number of decision variables of the optimisation problem (1.12) is N_u , the number of constraints is $4N_u + 2N$.

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 $u = [u_1 \dots u_{n_u}]^T$ and the vector of controlled variables is $y = [y_1 \dots y_{n_y}]^T$. The vector of decision variables of the MPC algorithm (1.3)

is hence of length $n_u N_u$. The minimised MPC cost-function for the MIMO case is

$$J(k) = \sum_{p=1}^N \sum_{m=1}^{n_y} \mu_{p,m} (y_m^{\text{sp}}(k+p|k) - \hat{y}_m(k+p|k))^2 + \sum_{p=0}^{N_u-1} \sum_{n=1}^{n_u} \lambda_{p,n} (\Delta u_n(k+p|k))^2 \quad (1.13)$$

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

$$J(k) = \sum_{p=1}^N \|y^{\text{sp}}(k+p|k) - \hat{y}(k+p|k)\|_{\mathbf{M}_p}^2 + \sum_{p=0}^{N_u-1} \|\Delta u(k+p|k)\|_{\mathbf{\Lambda}_p}^2 \quad (1.14)$$

Now, the set-point vector for the sampling instant $k+p$ known at the current sampling instant k is denoted by $y^{\text{sp}}(k+p|k)$, 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 $\mathbf{M}_p = \text{diag}(\mu_{p,1}, \dots, \mu_{p,n_y}) \geq 0$ is of dimensionality $n_y \times n_y$, the matrix $\mathbf{\Lambda}_p = \text{diag}(\lambda_{p,1}, \dots, \lambda_{p,n_u}) > 0$ is of dimensionality $n_u \times n_u$.

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

$$u_n^{\min} \leq u_n(k+p|k) \leq u_n^{\max}, \quad p = 0, \dots, N_u - 1, \quad n = 1, \dots, n_u \quad (1.15)$$

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

$$\Delta u_n^{\min} \leq \Delta u_n(k+p|k) \leq \Delta u_n^{\max}, \quad p = 0, \dots, N_u - 1, \quad n = 1, \dots, n_u \quad (1.16)$$

where $\Delta u_n^{\min}$ and $\Delta u_n^{\max}$ 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

$$y_m^{\min} \leq \hat{y}_m(k+p|k) \leq y_m^{\max}, \quad p = 1, \dots, N, \quad m = 1, \dots, n_y \quad (1.17)$$

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

$$u^{\min} = \begin{bmatrix} u_1^{\min} \\ \vdots \\ u_{n_u}^{\min} \end{bmatrix}, \quad u^{\max} = \begin{bmatrix} u_1^{\max} \\ \vdots \\ u_{n_u}^{\max} \end{bmatrix}, \quad \Delta u^{\min} = \begin{bmatrix} \Delta u_1^{\min} \\ \vdots \\ \Delta u_{n_u}^{\min} \end{bmatrix}, \quad \Delta u^{\max} = \begin{bmatrix} \Delta u_1^{\max} \\ \vdots \\ \Delta u_{n_u}^{\max} \end{bmatrix} \quad (1.18)$$

and the following vectors of length n_y

$$y^{\min} = \begin{bmatrix} y_1^{\min} \\ \vdots \\ y_{n_y}^{\min} \end{bmatrix}, \quad y^{\max} = \begin{bmatrix} y_1^{\max} \\ \vdots \\ y_{n_y}^{\max} \end{bmatrix} \quad (1.19)$$

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

$$\min_{\Delta u(k)} \left\{ J(k) = \sum_{p=1}^N \|y^{\text{sp}}(k+p|k) - \hat{y}(k+p|k)\|_{M_p}^2 + \sum_{p=0}^{N_u-1} \|\Delta u(k+p|k)\|_{\Lambda_p}^2 \right\} \quad (1.20)$$

subject to

$$\begin{aligned} u^{\min} &\leq u(k+p|k) \leq u^{\max}, \quad p = 0, \dots, N_u - 1 \\ \Delta u^{\min} &\leq \Delta u(k+p|k) \leq \Delta u^{\max}, \quad p = 0, \dots, N_u - 1 \\ y^{\min} &\leq \hat{y}(k+p|k) \leq y^{\max}, \quad p = 1, \dots, N \end{aligned}$$

where the norm is defined as $\|x\|_A^2 = x^T A x$ (the matrix 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 $n_u N_u$, the number of constraints is $4n_u N_u + 2n_y N$.

Although at each sampling instant as many as $n_u N_u$ 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. $u(k)$, $\Delta u^{\text{opt}}(k|k)$ and $u(k-1)$, are of length n_u .