Chapter 7
MPC Algorithms Using State-Space Wiener Models

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 details MPC algorithms for processes described by state-space Wiener models. At first, the simple MPC method based on the inverse static model is recalled and the rudimentary MPC-NO algorithm is described. Next, the computationally efficient MPC methods with on-line model linearisation are characterised: the MPC-SSL and the MPC-NPSL ones as well as two MPC schemes with on-line trajectory linearisation: the MPC-NPLT and MPC-NPLPT schemes. All MPC algorithms are presented without and with parameterisation using Laguerre functions. The classical and an original, very efficient prediction method, which lead to offset-free control, are presented. Finally, state estimation methods for MPC are shortly mentioned.

7.1 MPC-inv Algorithm in State-Space

In the simplest approach, we may use the MPC algorithm in which the inverse model of the static block is used. The general control system structure is presented in Fig. 3.1, the same as in the input-output process representation. In both input-output and state-space formulations, limitations of that approach are the same, as discussed in Chapter 3.1, i.e. the inverse model must exist and it is best when the process is described by the SISO Wiener model or the MIMO Wiener models I or III since, in such cases, all inverse models are of SISO type. For more complex Wiener models, the inverse models may be very complicated which makes implementation difficult or even impossible. Example simulated processes for which the MPC-inv algorithm based on the state-space Wiener model is discussed in the literature are: a continuous stirred tank reactor [3], polymerisation reactors [3, 6, 17], a neutralisation reactor [4].

7.2 MPC-NO Algorithm in State-Space

Prediction in State-Space for Offset-Free Control

A conventional method for providing offset-free MPC with state estimation is to augment the state equations taking into account in the model additional states of disturbances. For linear state-space systems, such a method was discussed in [5, 12, 13, 15, 16]. Next, this approach was extended in [14] to deal with nonlinear processes of the following general form (the subscript "p" refers to the process)

$$x(k+1) = f_{\mathrm{p}}(x(k), u(k), d_{\mathrm{p}}(k)) \tag{7.1}$$

$$y(k) = g_{\mathrm{p}}(x(k), d_{\mathrm{p}}(k)) \tag{7.2}$$

where $d_{\mathrm{p}}(k)$ represents all and generally unknown true disturbances affecting the controlled process. For prediction in MPC, the following augmented model is used

$$x(k+1) = f_{\mathrm{aug}}(x(k), u(k), d(k)) \tag{7.3}$$

$$d(k+1) = d(k) \tag{7.4}$$

$$y(k) = g_{\mathrm{aug}}(x(k), d(k)) \tag{7.5}$$

where d is the vector of disturbances used in the model, of length n_{d}, i.e. $d = \left[d_1 \ldots d_{n_{\mathrm{d}}}\right]^{\mathrm{T}}$. It is required that the number of disturbances located in the model does not exceed the number of measured outputs, i.e. $n_{\mathrm{d}} \leq n_{\mathrm{y}}$, which is an obvious disadvantage of the augmented state method. For example, when $n_{\mathrm{y}} = 1$, as many as $n_{\mathrm{x}} + 1$ possibilities exists: the disturbance d may be located in the consecutive n_{x} state equations or in the output equation. The actual location of the disturbance(s) is an issue, usually many possibilities must be verified and the best one chosen.

In this work an original prediction calculation method is used to determine the predicted values of state and output variables by means of the state-space Wiener model of the process. The prediction method detailed below is next used in all MPC algorithms. In place of the augmented model (7.3)-(7.5), basing on the state-space Wiener model defined by Eqs. (2.84)-(2.85), the following model is used for prediction

$$x(k+1) = \boldsymbol{A}x(k) + \boldsymbol{B}u(k) + \nu(k) \tag{7.6}$$

$$y(k) = g(\boldsymbol{C}x(k)) + d(k) \tag{7.7}$$

Unlike the augmented model, disturbances are taken into account in all state and output equations. The state disturbance vector $\nu = \left[\nu_1 \ldots \nu_{n_{\mathrm{x}}}\right]^{\mathrm{T}}$ is determined as the difference between the estimated state, $\tilde{x}(k)$, and the state calculated from the state equation (2.84)

$$v(k) = \tilde{x}(k) - (\boldsymbol{A}\tilde{x}(k-1) + \boldsymbol{B}u(k-1)) \tag{7.8}$$

where $\tilde{x} = \left[\tilde{x}_1 \ldots \tilde{x}_{n_x}\right]^{\mathrm{T}}$ denotes the vector of estimated state variables. Of course, if the state vector may be measured, measurements are used in place of estimations, which gives

$$v(k) = x(k) - (\boldsymbol{A}x(k-1) + \boldsymbol{B}u(k-1)) \tag{7.9}$$

Unfortunately, it may be possible only in very few cases in practice. The output disturbance vector $d = \left[d_1 \ldots d_{n_y}\right]^{\mathrm{T}}$ is calculated as the difference between the measured output vector, $y(k)$, and the output calculated from the output equation (2.85)

$$d(k) = y(k) - g(\tilde{v}(k)) = y(k) - g(\boldsymbol{C}(\tilde{x}(k))) \tag{7.10}$$

When the state vector may be measured, we have

$$d(k) = y(k) - g(v(k)) = y(k) - g(\boldsymbol{C}(x(k))) \tag{7.11}$$

In the MPC-NO strategy, the state variables predicted for the sampling instant $k+1$ at the current instant k are obtained from Eq. (7.6)

$$\hat{x}(k+1|k) = \boldsymbol{A}\tilde{x}(k) + \boldsymbol{B}u(k|k) + v(k) \tag{7.12}$$

When the state is measured, in place of Eq. (7.12), we have

$$\hat{x}(k+1|k) = \boldsymbol{A}x(k) + \boldsymbol{B}u(k|k) + v(k) \tag{7.13}$$

Similarly, using Eq. (7.6) recurrently, the predictions calculated at the sampling instant k for the sampling instants $k+p$ are

$$\hat{x}(k+p|k) = \boldsymbol{A}\hat{x}(k+p-1|k) + \boldsymbol{B}u(k+p-1|k) + v(k) \tag{7.14}$$

where $p = 2, \ldots, N$. Using Eq. (7.7), the output predictions for the sampling instant $k+p$ calculated at the current sampling instant k, are

$$\hat{y}(k+p|k) = g(\boldsymbol{C}\hat{x}(k+p|k)) + d(k) \tag{7.15}$$

for $p = 1, \ldots, N$. In Eqs. (7.12), (7.13) and (7.14) the same state disturbance vector, $v(k)$, is used over the whole prediction horizon. Similarly, in Eq. (7.15) the same output disturbance vector, $d(k)$, is used. Because, typically, variability of future disturbances is not known, they are assumed to be constant over the whole prediction horizon [19].

The presented disturbance modelling was introduced in [20] for linear state-space systems and further extended for nonlinear ones in [21, 22]. A computationally efficient MPC using the considered disturbance modelling was introduced in [8, 7]. The discussed approach to offset-free control has the following advantages:

a) simplicity of development, no need to check all possibilities of disturbance location necessary in the augmented state approach,

b) ability to compensate for the deterministic constant-type disturbances affecting the process, which are crucial in process control because they include unavoidable modelling errors or piecewise-constant disturbances,
c) only the process state must be estimated, not accompanied by the disturbance vector as it is necessary in the case of the conventional augmented state method.

A unique feature of the proposed prediction method is that the resulting MPC controllers assure offset-free control without the necessity to use an additional observer of the deterministic disturbances. The key factor is the use of properly defined and updated state and output disturbance predictions, $\nu(k)$ and $d(k)$, used in state and output prediction equations, respectively.

Let us derive scalar prediction equations which will be convenient for future transformations. We assume that the estimated state vector, $\tilde{x}$, is used. When the state is measured, it must be replaced by the measured vector, x.

Prediction Using State-Space SISO Wiener Model

At first, let us discuss the state-space SISO case in which the Wiener model depicted in Fig. 2.1 is used. Model matrices $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$ are given by Eq. (2.83). From Eqs. (2.86) and (7.12), the state predictions for the sampling instant $k+1$ are

$$\hat{x}_i(k+1|k) = \sum_{j=1}^{n_\mathrm{x}} a_{i,j}\tilde{x}_j(k) + b_{i,1}u(k|k) + \nu_i(k) \tag{7.16}$$

for $i = 1, \ldots, n_\mathrm{x}$. From Eqs. (2.86) and (7.14), the state predictions for the sampling instant $k+p$ are

$$\hat{x}_i(k+p|k) = \sum_{j=1}^{n_\mathrm{x}} a_{i,j}\hat{x}_j(k+p-1|k) + b_{i,1}u(k+p-1|k) + \nu_i(k) \tag{7.17}$$

for $i = 1, \ldots, n_\mathrm{x}$, $p = 2, \ldots, N$. From Eqs. (2.87) and (7.15), the predictions of the controlled variable are

$$\begin{aligned}\hat{y}(k+p|k) &= g(v(k+p|k)) + d(k)\\ &= g\left(\sum_{i=1}^{n_\mathrm{x}} c_{1,i}\hat{x}_i(k+p|k)\right) + d(k)\end{aligned} \tag{7.18}$$

From Eqs. (2.86) and (7.8), the state disturbances are estimated from

$$\nu_i(k) = \tilde{x}_i(k) - \left(\sum_{j=1}^{n_\mathrm{x}} a_{i,j}\tilde{x}_j(k-1) + b_{i,1}u(k-1)\right) \tag{7.19}$$

for $i = 1, \ldots, n_{\mathrm{x}}$. From Eqs. (2.87) and (7.10), the output disturbance is estimated from

$$d(k) = y(k) - g\left(\sum_{i=1}^{n_{\mathrm{x}}} c_{1,i}\tilde{x}_i(k)\right) \tag{7.20}$$

Prediction Using State-Space MIMO Wiener Model I

Next, we will discuss the case when the state-space MIMO Wiener model I depicted in Fig. 2.2 is used for prediction. Model matrices $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$ are given by Eq. (2.89). From Eqs. (2.91) and (7.12), the state predictions for the sampling instant $k+1$ are

$$\hat{x}_i(k+1|k) = \sum_{j=1}^{n_{\mathrm{x}}} a_{i,j}\tilde{x}_j(k) + \sum_{j=1}^{n_{\mathrm{u}}} b_{i,j}u_j(k|k) + \nu_i(k) \tag{7.21}$$

for $i = 1, \ldots, n_{\mathrm{x}}$. From Eqs. (2.91) and (7.14), the state predictions for the sampling instant $k+p$ are

$$\hat{x}_i(k+p|k) = \sum_{j=1}^{n_{\mathrm{x}}} a_{i,j}\hat{x}_j(k+p-1|k) + \sum_{j=1}^{n_{\mathrm{u}}} b_{i,j}u_j(k+p-1|k) + \nu_i(k) \tag{7.22}$$

for $i = 1, \ldots, n_{\mathrm{x}}$, $p = 2, \ldots, N$. From Eqs. (2.92) and (7.15), the predictions of the controlled variables are

$$\begin{aligned} \hat{y}_m(k+p|k) &= g_m(v_m(k+p|k)) + d_m(k) \\ &= g_m\left(\sum_{i=1}^{n_{\mathrm{x}}} c_{m,i}\hat{x}_i(k+p|k)\right) + d_m(k) \end{aligned} \tag{7.23}$$

for $m = 1, \ldots, n_{\mathrm{y}}$, $p = 1, \ldots, N$. From Eqs. (2.91) and (7.8), the state disturbances are estimated from

$$\nu_i(k) = \tilde{x}_i(k) - \left(\sum_{j=1}^{n_{\mathrm{x}}} a_{i,j}\tilde{x}_j(k-1) + \sum_{j=1}^{n_{\mathrm{u}}} b_{i,j}u_j(k-1)\right) \tag{7.24}$$

for $i = 1, \ldots, n_{\mathrm{x}}$. From Eqs. (2.92) and (7.10), the output disturbances are estimated from

$$d_m(k) = y_m(k) - g_m\left(\sum_{i=1}^{n_{\mathrm{x}}} c_{m,i}\tilde{x}_i(k)\right) \tag{7.25}$$

for $m = 1, \ldots, n_{\mathrm{y}}$.

Prediction Using State-Space MIMO Wiener Model II

Finally, we will discuss the case when the state-space MIMO Wiener model II depicted in Fig. 2.3 is used for prediction. Model matrices $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$ are given by Eq. (2.94). Because the state equation is the same in both types of the state-space MIMO Wiener model, state predictions given by Eqs. (7.21) and (7.22) holds true in the second model structure. Analogously, the state disturbances are estimated from Eq. (7.24) in both cases. From Eqs. (2.97) and (7.15), the output predictions for the sampling instant $k + p$ are

$$\begin{aligned}\hat{y}_m(k+p|k) &= g_m(v_1(k+p|k), \ldots, v_{n_\mathrm{v}}(k+p|k)) + d_m(k) \\ &= g_m\left(\sum_{i=1}^{n_\mathrm{x}} c_{1,i}\hat{x}_i(k+p|k), \ldots, \sum_{i=1}^{n_\mathrm{x}} c_{n_\mathrm{v},i}\hat{x}_i(k+p|k)\right) + d_m(k) \qquad (7.26)\end{aligned}$$

for $m = 1, \ldots, n_\mathrm{y}$, $p = 1, \ldots, N$. From Eqs. (2.97) and (7.10), the output disturbances are estimated from

$$d_m(k) = y_m(k) - g_m\left(\sum_{i=1}^{n_\mathrm{x}} c_{1,i}\tilde{x}_i(k), \ldots, \sum_{i=1}^{n_\mathrm{x}} c_{n_\mathrm{v},i}\tilde{x}_i(k)\right) \qquad (7.27)$$

for $m = 1, \ldots, n_\mathrm{y}$.

Optimisation

Taking into account the obtained state and ouput prediction equations, i.e. Eqs. (7.16), (7.17), (7.18) (the state-space SISO Wiener model), (7.21), (7.22), (7.23) (the state-space MIMO Wiener model I), (7.21), (7.22), (7.26) (the state-space MIMO Wiener model II), it is clear that the predicted controlled variables are nonlinear functions of the calculated future increments (1.3). It means that the resulting MPC-NO optimisation problem is also nonlinear. The general formulations of these MPC-NO optimisation problems are the same when input-output and state-space process descriptions are used. If hard constraints are imposed on the controlled variables, we obtain the nonlinear optimisation task (1.35). If soft constraints are used, the nonlinear task is defined by Eq. (1.39). As far as MATLAB implementation is considered, the general structures of the vectors and matrices which define the constraints are the same in both input-output and state-space formulations. The MPC optimisation task is solved in MATLAB by means of the `fmincon` function. All details are given in Chapter 3.2. Of course, the main difference is the way the predicted vector of the controlled variables, $\hat{\boldsymbol{y}}(k)$, is calculated. In the state-space description the satate equations must be used. Of course, we have to use a state estimator when the state cannot be measured.

Applications of the MPC-NO algorithm for Wiener systems are rather rare. An example application of the MPC-NO algorithm based on a state-space Wiener model to a plug-flow tubular reactor is presented in [2]. The MPC-NO algorithm is rather treated as a reference to which alternative, more computationally efficient control schemes are compared.

7.3 MPC-NO-P Algorithm in State-Space

In the state-space MPC-NO-P algorithm, all prediction equations derived in Chapter 7.2 for the MPC-NO algorithm can be used, it is only necessary to find from Eq. (1.56) the control increments, $\triangle \boldsymbol{u}(k)$, for the actually calculated vector of decision variables, $c(k)$.

Although parameterisation using Laguerre functions makes it possible to reduce the number of decision variables, we still have to solve a nonlinear optimisation task. It is because the predicted vector of the controlled variables, $\hat{\boldsymbol{y}}(k)$, is a nonlinear function of the calculated decision vector, $c(k)$. The general formulations of these MPC-NO optimisation problems are the same when input-output and state-space process descriptions are used. If hard constraints are imposed on the controlled variables, we obtain the nonlinear optimisation task (3.54). If soft constraints are used, the nonlinear task is defined by Eq. (3.66). As far as MATLAB implementation is considered, the general structures of the vectors and matrices which define the constraints are the same in both input-output and state-space formulations. The MPC optimisation task is solved in MATLAB by means of the `fmincon` function. All details are given in Chapter 3.3.

7.4 MPC-NPSL and MPC-SSL Algorithms in State-Space

The MPC-SSL algorithm based on the state-space Wiener model with a neural static block is described in [1] and [10]. Effectiveness of the algorithm is shown for the following simulated processes: a gasifier in the first case and an intensified continuous chemical reactor in the second case. The authors of these works show that the classical LMPC algorithm based on a linear process description results in unsatisfactory control quality and the MPC-SSL scheme gives much better results. Unfortunately, the MPC-NPSL strategy is not discussed, yet it is likely to improve the quality of control. Both MPC-SSL and MPC-NPSL schemes for the state-space MIMO Wiener model I are discussed and compared in [9]. The description presented in this Chapter extends that publication.

Prediction Using State-Space SISO Wiener Model

At first, let us discuss the state-space SISO case in which the Wiener model depicted in Fig. 2.1 is used. The time-varying gain of the nonlinear static part of the model for the current operating point is defined by the general equations (3.69) and (3.70), i.e.

$$K(k) = \frac{\mathrm{d}y(k)}{\mathrm{d}v(k)} = \frac{\mathrm{d}g(v(k))}{\mathrm{d}v(k)} \tag{7.28}$$

where the model signal $v(k)$ is calculated using Eq. (2.88) which gives

$$v(k) = \sum_{i=1}^{n_\mathrm{x}} c_{1,i}\tilde{x}_i(k) \tag{7.29}$$

Prediction Using State-Space MIMO Wiener Model I

If the state-space MIMO Wiener model I depicted in Fig. 2.2 is used, the time-varying gains of the nonlinear static blocks of the model for the current operating point are defined by the general equations (3.102)-(3.103) and (3.104)-(3.105), i.e.

$$K_m(k) = \frac{\mathrm{d}y_m(k)}{\mathrm{d}v_m(k)} = \frac{\mathrm{d}g_m(v_m(k))}{\mathrm{d}v_m(k)} \tag{7.30}$$

for $m = 1, \ldots, n_\mathrm{y}$, where the model signals $v_m(k)$ are calculated using Eq. (2.93) which gives

$$v_m(k) = \sum_{i=1}^{n_\mathrm{x}} c_{m,i}\tilde{x}_i(k) \tag{7.31}$$

Prediction Using State-Space MIMO Wiener Model II

If the state-space MIMO Wiener model II depicted in Fig. 2.2 is used, the time-varying gains of the nonlinear static blocks of the model for the current operating point are defined by the general equations (3.125)-(3.126) and (3.127)-(3.128), i.e.

$$K_{m,n}(k) = \frac{\mathrm{d}y_m(k)}{\mathrm{d}v_n(k)} = \frac{\mathrm{d}g_m(v_1(k), \ldots, v_{n_\mathrm{v}}(k))}{\mathrm{d}v_n(k)} \tag{7.32}$$

for $m = 1, \ldots, n_\mathrm{y}$, $n = 1, \ldots, n_\mathrm{v}$. The model signals v_n are calculated using Eq. (2.98) which gives

$$v_n(k) = \sum_{i=1}^{n_\mathrm{x}} c_{n,i}\tilde{x}_i(k) \tag{7.33}$$

As a result of linearisation, taking into account the serial structure of the Wiener model, we may easily conclude that in the SISO case, the model output may be expressed as multiplication of the time-varying gain $K(k)$ and the auxiliary signal $v(k)$, as defined by Eq. (3.71). In the case of the MIMO Wiener model I, we may also notice that the signals $y_1(k), \ldots, y_{n_\mathrm{y}}(k)$ may be easily found as multiplications of the corresponding time-varying gains $K_1(k), \ldots, K_{n_\mathrm{y}}(k)$ and the auxiliary signals $v_1(k), \ldots, v_{n_\mathrm{y}}(k)$, respectively, as defined by Eqs. (3.106)-(3.107). The diagonal gain matrix $\boldsymbol{K}(k)$, of dimensionality $n_{\mathrm{y}} \times n_{\mathrm{y}}$, is defined by Eq. (3.108). Hence, the linear approximation of the state-space Wiener model (2.84)-(2.85) is

$$x(k+1) = \boldsymbol{A}x(k) + \boldsymbol{B}u(k) \tag{7.34}$$

$$y(k) = \boldsymbol{K}(k)v(k) = \boldsymbol{K}(k)\boldsymbol{C}x(k) \tag{7.35}$$

When the MIMO Wiener model II is used, the signals $y_1(k), \ldots, y_{n_{\mathrm{y}}}(k)$ may be easily found as multiplications of the corresponding time-varying gains $K_{1,1}(k), \ldots, K_{n_{\mathrm{y}},n_{\mathrm{v}}}(k)$ and the auxiliary signals $v_1(k), \ldots, v_{n_{\mathrm{v}}}(k)$. In contrast to the MIMO Wiener model I, all input-output channels must be taken into consideration, as defined by Eqs. (3.129)-(3.130). The resulting linear approximation of the MIMO Wiener model II is also defined by Eqs. (7.34)-(7.35), but now $v(k)$ is the vector of length n_{v} and the matrix $\boldsymbol{K}(k)$, of dimensionality $n_{\mathrm{y}} \times n_{\mathrm{v}}$, is defined by Eq. (3.131). All things considered, Eqs. (7.34)-(7.35) are used in all three cases of the state-space Wiener models, i.e. for the SISO structure as well as MIMO representations I and II. One may easy notice that the structure of the obtained linearised model (7.34)-(7.35) is similar to that of the classical linear state-space models, but a time-varying matrix $\boldsymbol{K}(k)$ is used in the output equation.

Using recurrently the general state prediction formula (7.6) and the state equation (7.34), it is possible to calculate the predicted state vector for the whole prediction horizon ($p = 1, \ldots, N$)

$$\hat{x}(k+1|k) = \boldsymbol{A}x(k) + \boldsymbol{B}u(k|k) + \nu(k) \tag{7.36}$$

$$\hat{x}(k+2|k) = \boldsymbol{A}\hat{x}(k+1|k) + \boldsymbol{B}u(k+1|k) + \nu(k) \tag{7.37}$$

$$\hat{x}(k+3|k) = \boldsymbol{A}\hat{x}(k+2|k) + \boldsymbol{B}u(k+2|k) + \nu(k) \tag{7.38}$$

$$\vdots$$

The state predictions can be expressed as functions of the increments of the future control increments (similarly to Eqs.(3.89)-(3.91), the influence of the past is not taken into account)

$$\hat{x}(k+1|k) = \boldsymbol{B}\triangle u(k|k) + \ldots \tag{7.39}$$

$$\hat{x}(k+2|k) = (\boldsymbol{A}+\boldsymbol{I})\boldsymbol{B}\triangle u(k|k) + \boldsymbol{B}\triangle u(k+1|k) + \ldots \tag{7.40}$$

$$\begin{aligned}\hat{x}(k+3|k) = {} & (\boldsymbol{A}^2+\boldsymbol{A}+\boldsymbol{I})\boldsymbol{B}\triangle u(k|k) \\ & + (\boldsymbol{A}+\boldsymbol{I})\boldsymbol{B}\triangle u(k+1|k) + \boldsymbol{B}\triangle u(k+2|k) + \ldots\end{aligned} \tag{7.41}$$

$$\vdots$$

Let us define the predicted state trajectory over the whole prediction horizon, the vector of length $n_\mathrm{x}N$

$$\hat{\boldsymbol{x}}(k) = \begin{bmatrix} \hat{x}(k+1|k) \\ \vdots \\ \hat{x}(k+N|k) \end{bmatrix} \tag{7.42}$$

From Eqs. (7.39)-(7.41), the predicted state vector can be expressed in the following way

$$\hat{\boldsymbol{x}}(k) = \boldsymbol{P}\triangle\boldsymbol{u}(k) + \boldsymbol{x}^0(k) \tag{7.43}$$

where the matrix

$$\boldsymbol{P} = \begin{bmatrix} \boldsymbol{B} & \ldots & \boldsymbol{0}_{n_\mathrm{x}\times n_\mathrm{u}} \\ (\boldsymbol{A}+\boldsymbol{I})\boldsymbol{B} & \ldots & \boldsymbol{0}_{n_\mathrm{x}\times n_\mathrm{u}} \\ \vdots & \ddots & \vdots \\ \left(\sum_{i=1}^{N_\mathrm{u}-1}\boldsymbol{A}^i+\boldsymbol{I}\right)\boldsymbol{B} & \ldots & \boldsymbol{B} \\ \left(\sum_{i=1}^{N_\mathrm{u}}\boldsymbol{A}^i+\boldsymbol{I}\right)\boldsymbol{B} & \ldots & (\boldsymbol{A}+\boldsymbol{I})\boldsymbol{B} \\ \vdots & \ddots & \vdots \\ \left(\sum_{i=1}^{N-1}\boldsymbol{A}^i+\boldsymbol{I}\right)\boldsymbol{B} & \ldots & \left(\sum_{i=1}^{N-N_\mathrm{u}}\boldsymbol{A}^i+\boldsymbol{I}\right)\boldsymbol{B} \end{bmatrix} \tag{7.44}$$

is of dimensionality $n_\mathrm{x}N \times n_\mathrm{u}N_\mathrm{u}$ and the free state trajectory vector

$$\boldsymbol{x}^0(k) = \begin{bmatrix} x^0(k+1|k) \\ \vdots \\ x^0(k+N|k) \end{bmatrix} \tag{7.45}$$

is of length $n_\mathrm{x}N$. Using the obtained linearised output equation (7.35) and the state prediction equation (7.43), we derive the predicted trajectory of the controlled variables, defined by Eq. (1.22), as

$$\hat{\boldsymbol{y}}(k) = \widetilde{\boldsymbol{K}}(k)\boldsymbol{P}\triangle\boldsymbol{u}(k) + \boldsymbol{y}^0(k) \tag{7.46}$$

where the matrix of dimensionality $n_\mathrm{y}N \times n_\mathrm{x}N$ is

$$\widetilde{\boldsymbol{K}}(k) = \mathrm{diag}(\boldsymbol{K}(k)\boldsymbol{C}, \ldots, \boldsymbol{K}(k)\boldsymbol{C}) \tag{7.47}$$

Let us remind that in the input-output process description we use the prediction equation (3.93), i.e. $\hat{\boldsymbol{y}}(k) = \boldsymbol{G}(k)\triangle\boldsymbol{u}(k) + \boldsymbol{y}^0(k)$. The relation obtained for the state-space approach, i.e. Eq. (7.46), may be easily transformed to Eq. (3.93) equating $\boldsymbol{G}(k) = \widetilde{\boldsymbol{K}}(k)\boldsymbol{P}$. It means that in the state-space process description we may use the same prediction equations derived for the input-output case. The output free trajectory vector is defined by Eq. (3.88). In the MPC-NPSL algorithm, the consecutive