ISSN 0005-1179, Automation and Remote Control, 2015, Vol. 76, No. 9, pp. 1704–1712. © Pleiades Publishing, Ltd., 2015.
Original Russian Text © R.N. Zhuchkov, 2013, published in Upravlenie Bol'shimi Sistemami, 2013, No. 46, pp. 147–162.

Application of Predictive Control Approach in Stabilizing Control Design of Networked Plants

R. N. Zhuchkov

Polytechnic Institute of Alekseev State Technical University, Arzamas, Russia
e-mail: roman_jkv@mail.ru

Received June 22, 2013

Abstract—A stabilization problem in a networked control system with packet dropouts is studied and dynamic output-feedback control is constructed. Predictive control approach is employed for obtaining plant state estimates at each step k. This eliminates the need for considering explicitly plant state switching when a packet dropout occurs.

DOI: 10.1134/S0005117915090167

1. BASIC PRINCIPLES OF PREDICTIVE CONTROL

Predictive control belongs to a class of control design algorithms which involve prediction models for future responses of a system. Initially, the approach was intended for satisfying specific control requirements at power plants and petroleum refineries. Nowadays, predictive control finds applications in different fields such as food, automotive and aerospace industries.

The whole essence of predictive control consists in the following. At each control interval, the algorithm tries to optimize the future behavior of a system through constructing a sequence of future control actions. The control sequence is calculated in order to optimize the future behavior of a system within a time interval called prediction horizon. The first control action from the obtained sequence is supplied to the plant and at the next step the control problem is solved all over again using updated measurements.

Predictive control principles date back to the 1960s, see [12]. However, researchers demonstrated a rapidly growing interest in such methods only in the 1980s after the appearance of first publications on predictive control: identification and command (IDCOM) [19], dynamic matrix control (DMC) [9, 10] and the pioneering comprehensive treatment of generalized predictive control (GPC) [6, 7]. Despite that, in the initial form, the underlying ideas of DMC and GPC were similar, DMC serves for multi-dimensional constrained control, whereas GPC mostly fits one-dimensional problems and adaptive control problems.

The term "predictive control" is associated with an explicit model of a plant used for predicting the future behavior of system output. This prediction allows solving optimal control problems in real time, where tracking errors (the deviations between predicted and desired outputs) are minimized during the prediction horizon under input/output constraints. In the case of a linear model, the optimization problem is quadratic (linear) if the performance criterion employs the L_2 norm (the L_1/L_∞ norm, respectively). According to the major principle of predictive control, at step t only the first term from the predicted optimal control sequence is applied to the plant. And then the control problem is solved for step $t+1$.

Figure 1 illustrates the basic principle of predictive control [17]. Our discussion gets confined to single-input single-output systems with discrete time. The current state corresponds to step t. In Fig. 1 readers can observe two trajectories, *viz.*, an ideal trajectory (dashed line) and a predicted trajectory (points). The predicted trajectory begins at step t and defines a trajectory to-be-followed

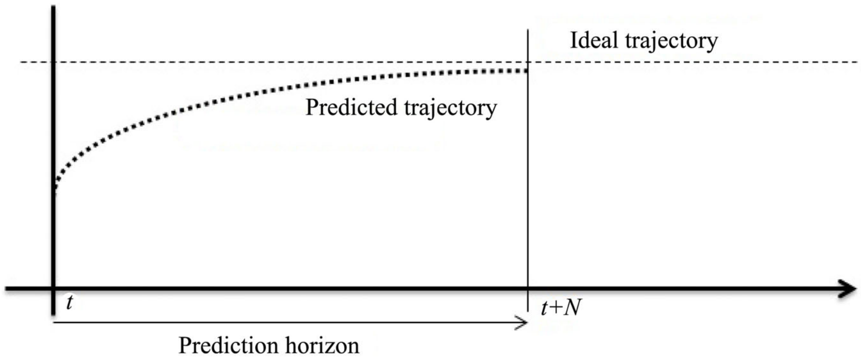

Fig. 1. Predictive control.

by a plant for getting back to the ideal trajectory. A predictive controller has its own model used for predicting system behavior within a prediction horizon. In the elementary case, we can endeavor to choose the predicted trajectory so as to combine it with the ideal one at the end of the prediction horizon.

The issues of real-time optimization, stability and performance are well studied for systems described by linear models (e.g., see the books [3–5, 8, 16, 20]). An appreciable advancement in predictive control application for hybrid systems, discrete-time systems, logical systems and heuristic analysis was achieved in [2].

Now, we outline several results on predictive control of networked systems. The paper [15] suggested stabilizing control for a networked system with packet dropouts between a sensor and controller. The authors [11] developed a predictive control method in the case of two-way packet losses; a Lyapunov function dependent on data packet losses was involved to stabilize the closed-loop system. In [13] a networked control system was designed, where predictions serve for compensating signal delays and packet losses during transmission between a plant and a control system. To analyze the properties of the system, the authors introduced the notion of a sequence of predictions which allows defining the properties of a network required for closed-loop system stability.

In the preceding works [1], we treated packet dropouts as a structural state of a networked control system. However, as dropouts occur, a plant does not pass to a new state and continues normal functioning. Switching of a designed control system to a new structural state affects not the plant controlled via closed loop, but information about it.

From this viewpoint, predictive control seems fruitful exactly in the sense of system stay in a same structural state with available information about the system. Yet, one should keep in mind that such information can be predicted, not true.

Let us formulate the key ideas employed in the previous work to design control algorithms for networked control systems. First, we accepted the well-known separation principle allowing to decompose a control system into a plant and an observer. And the gain matrices for the observer and controller were calculated independently. Second, possible packet dropouts were considered for the observer and controller, with construction of either different gain matrices for different structural states or a common gain matrix for all structural states.

Using predictive control principles, we suggest the following modification of the networked control algorithm. Introduce a prediction horizon coinciding with the maximum possible number of lost data packets. This quantity characterizes a network data communication channel and can be chosen with some margin. Next, design two buffers for storing predicted measurements and control actions. In the sequel, we demonstrate a state vector estimation procedure based on a Kalman filter taking into account predictive control. Using predictions of the current and future states of the plant, a sequence of control actions is constructed and supplied to the plant. Here we emphasize an important feature: in contrast to [18], a common output-feedback gain matrix serves for the current and future steps. This is done owing to the assumption that a common output-feedback gain matrix can be found for the system and packet dropouts do not affect its structure.

2. THE PRINCIPLE OF SYSTEM FUNCTIONING

Consider a networked control system illustrated by Fig. 2. It runs in the following way:

1) the plant conducts measurements in the form of current and predicted measurements for several steps ahead (an event horizon);
2) the plant sends a formed packet to the control system;
3) if a packet dropout takes no place, the received packet is stored in a buffer; otherwise, predicted measurements for an appropriate step are taken from a buffer;
4) an observer is constructed;
5) a packet is formed from the state estimate for the current step and the predicted states for several steps ahead;
6) the packet is sent to the plant;
7) if this packet successfully reaches the plant (no dropouts occur), it is stored in the buffer;
8) the control action for the current step or the predictions for the preceding steps are applied.

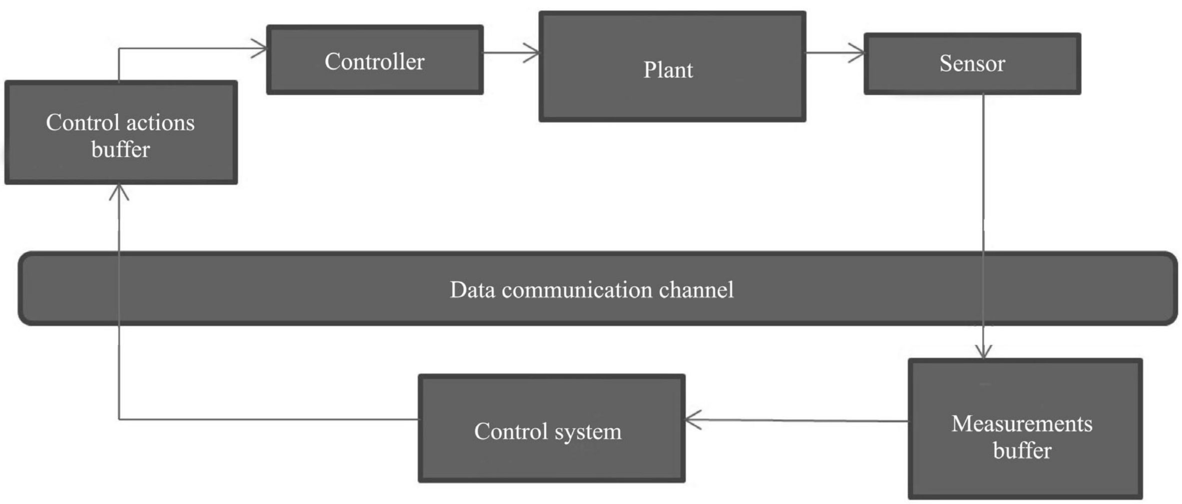

Fig. 2. A networked control system.

3. SYSTEM EQUATIONS

Consider a linear system governed by the difference equation

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k, \\ y_k &= Cx_k, \end{aligned} \tag{1}$$

where x_{k+1} denotes the n-dimensional transition state vector; x_k means the n-dimensional initial state vector; u_k stands for the m-dimensional control vector; y_k is the l-dimensional measurement (output) vector; k specifies discrete time expressed by the number of discretization intervals of length Δt; and finally, $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$ are the state transition matrix and the control gain matrix, respectively.

Similarly to [1], let us accept the separation principle for independent control search and state vector estimation. A valuable property of predictive control is that system switching does not affect stabilizing control design $u_k = -G\hat{x}_k$: even in the case of a packet dropout, one can use information from previous steps stored in the buffer. In what follows, we employ the method of Lyapunov functions and linear matrix inequalities for calculating the output-feedback gain matrix G. In the absence of switching, the formulas become trivial and are omitted here.

Remark 1. As opposed to the earlier paper [1], where the system changes its structural states, applicability of the separation principle can be easily argued. For this, consider the system (1) and the associated observer:

$$\begin{aligned} x_{k+1} &= Ax_k - BG\hat{x}_k, \\ \hat{x}_{k+1} &= Ax_k - BG\hat{x}_k + K(y_k - C\hat{x}_k). \end{aligned} \tag{2}$$

By passing from the estimation vector to the error vector $\tilde{x} = x - \hat{x}$ in the second equation above, we rewrite the system (2) as

$$\begin{aligned} x_{k+1} &= (A - BG)x_k + BG\tilde{x}_k, \\ \tilde{x}_{k+1} &= (A - KC)\tilde{x}_k. \end{aligned} \tag{3}$$

Its equivalent matrix representation is

$$\begin{bmatrix} x_{k+1} \\ \tilde{x}_{k+1} \end{bmatrix} = \begin{bmatrix} A - BG & BG \\ 0 & A - KC \end{bmatrix} \begin{bmatrix} x_k \\ \tilde{x}_k \end{bmatrix}. \tag{4}$$

Obviously, the system (4) possesses a block upper triangular matrix whose eigenvalues equal the eigenvalues of the corresponding block diagonal matrix (or the eigenvalues of the separated system).

Therefore, now the major problem is to estimate the state vector for any step k, regardless of possible dropouts in the network data communication channel. For this, take advantage of predictive control principles and compile the extended state vector

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k, \\ x_{k+2} - Ax_{k+1} &= Bu_{k+1}, \\ x_{k+N+1} - Ax_{k+N} &= Bu_{k+N}; \end{aligned} \tag{5}$$

$$\begin{aligned} y_k &= Cx_k, \\ y_{k+1} &= Cx_{k+1}, \\ y_{k+2} &= Cx_{k+2}, \\ &\vdots \\ y_{k+N} &= Cx_{k+N}. \end{aligned} \tag{6}$$

Here N is the size of the buffer (the event horizon).

Reexpress the system (5) in the form

$$\begin{gathered}\begin{bmatrix} I & 0 & \dots & 0 & 0 \\ -A & I & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & -A & I \end{bmatrix} \begin{bmatrix} x_{k+1} \\ x_{k+2} \\ x_{k+3} \\ \vdots \\ x_{k+N+1} \end{bmatrix} \\ = \begin{bmatrix} A & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} x_k \\ x_{k+1} \\ x_{k+2} \\ \vdots \\ x_{k+N} \end{bmatrix} + \begin{bmatrix} B & 0 & 0 & \dots & 0 \\ 0 & B & 0 & \dots & 0 \\ 0 & 0 & B & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & B \end{bmatrix} \begin{bmatrix} u_k \\ u_{k+1} \\ u_{k+2} \\ \vdots \\ u_{k+N} \end{bmatrix},\end{gathered} \tag{7}$$

$$\begin{bmatrix} y_k \\ y_{k+1} \\ y_{k+2} \\ \vdots \\ y_{k+N} \end{bmatrix} = \begin{bmatrix} C & 0 & 0 & \dots & 0 \\ 0 & C & 0 & \dots & 0 \\ 0 & 0 & C & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & C \end{bmatrix} \begin{bmatrix} x_k \\ x_{k+1} \\ x_{k+2} \\ \vdots \\ x_{k+N} \end{bmatrix}. \tag{8}$$

Introduce the extended state and measurement vectors $X = [x_k\ x_{k+1}\ \dots\ x_{k+N}]$, $Y = [y_k\ y_{k+1}\ \dots\ y_{k+N}]$. Then the system (7) takes the form

$$\begin{gathered} A_+ X_{k+1} = A_0 X_k + B_0 u_k, \\ Y_k = C_0 X_k. \end{gathered} \tag{9}$$

Note that in the vector Y_k only the measurements y_k are the real output of the system (1); the rest components $y_{k+1}, \dots, y_{k+N}$ represent the predicted measurements, e.g., constructed using the estimates of the extended state vector.

Equations (5)–(9) incorporate no external disturbances, though the system (1) admits their appearance. This can be explained as follows. While predicting future states and measurements, we believe that noises in the original system (if any) have zero mean. In this case, such noises are compensated during state vector estimation by a Kalman filter.

To estimate the state of the system (9), apply the linear Kalman filter

$$\hat{X}_{k+1} = A_+^{-1} A_0 \hat{X}_k + A_+^{-1} B U_k + K(Y - C_0 \hat{X}_k). \tag{10}$$

The matrix C_0 remains invariable if a packet dropout occurs.

Remark 2. It would seem that, in the case of a packet dropout, a reasonable approach is to use the matrix C_0 of the following form:

$$Y = \begin{bmatrix} C & 0 & 0 & \dots & 0 \\ 0 & C & 0 & \dots & 0 \\ 0 & 0 & C & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}. \tag{11}$$

Here the missed packet is replaced by zeros. However, the numerical experiments have demonstrated that such approach impairs the convergence characteristics of the Kalman filter, as its original version is sensitive to structural switching of the system.

Invariance of the matrix C_0 indicates that the last obtained measurements (not zero data) replace the lost measurements.

And finally, we emphasize that within the prediction horizon the predicted states and measurements are chosen for each system individually. They can depend on such factors as system dynamics and external disturbances. Once again, recall the basic principle of predictive control: within the prediction horizon, the behavioral model of the system is selected on the basis of its desired dynamics and may vary rather widely.

4. EXAMPLE

As an example, consider stabilizing control design for a quadcopter. The linear system characterizing small deviations from an equilibrium in the angle of roll and zero position in the $3D$ space was adopted from [14].

In the continuous-time setting, the system takes the form

$$\ddot{\phi} = \frac{1}{I}\Gamma, \tag{12}$$

$$\ddot{x} = -g\phi + \frac{1}{Lm}\Gamma + \frac{1}{m}d_t. \tag{13}$$

Here m is the weight of the quadcopter; I designates the inertia moment of axis X; L gives the distance between the propeller plane and the engine; d_t represents a random disturbance (e.g., wind force).

Compile the new vector of unknown variables $X = \begin{bmatrix}\phi & \dot{\phi} & x & \dot{x}\end{bmatrix}$ and choose $\dot{\phi}$ and x as measurable quantities.

To introduce disturbances into the system, apply the function d_t of the following form (see Fig. 3).

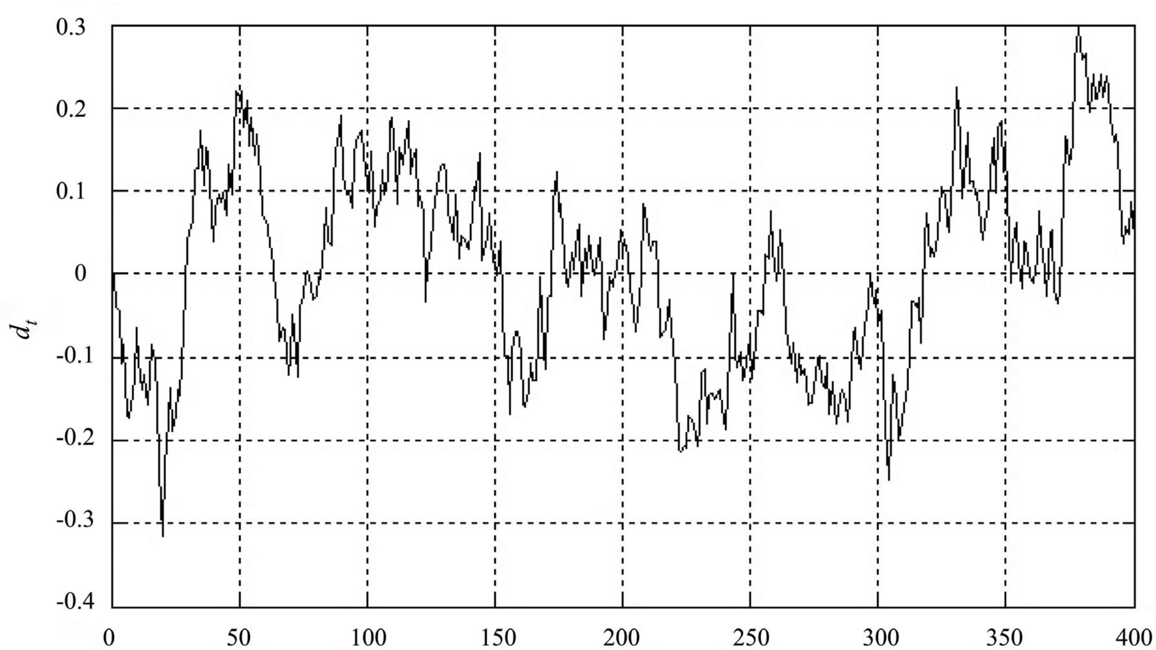

Fig. 3. Random disturbance.

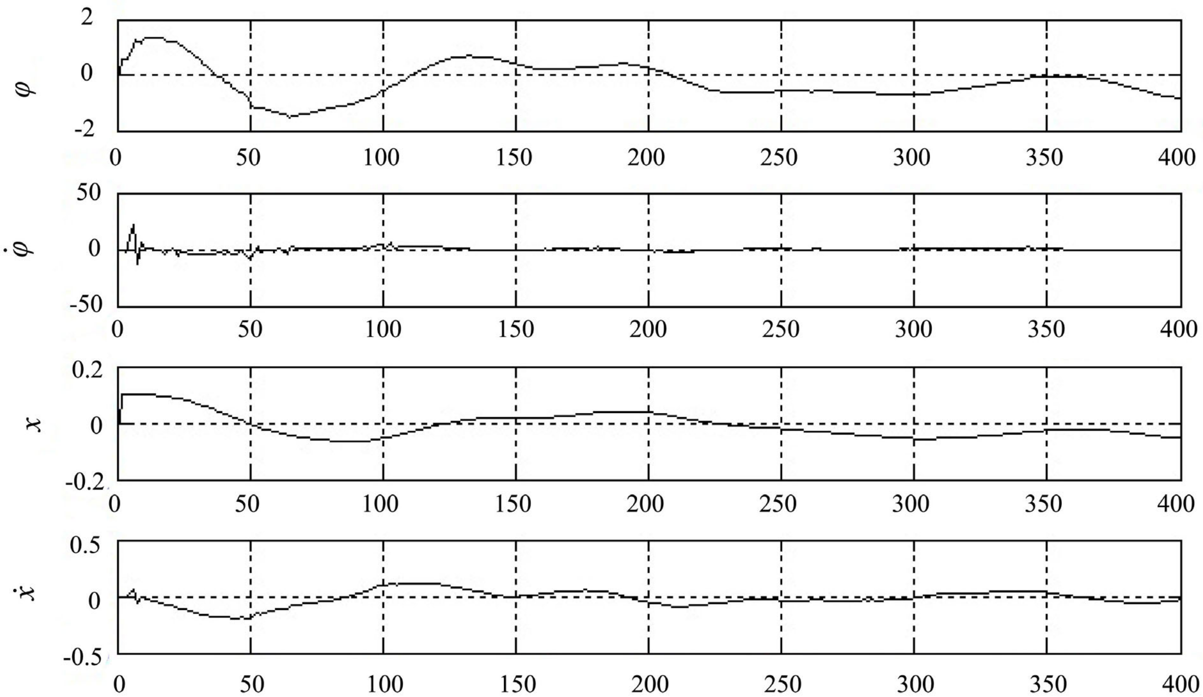

Fig. 4. The trajectories of the stabilized system.

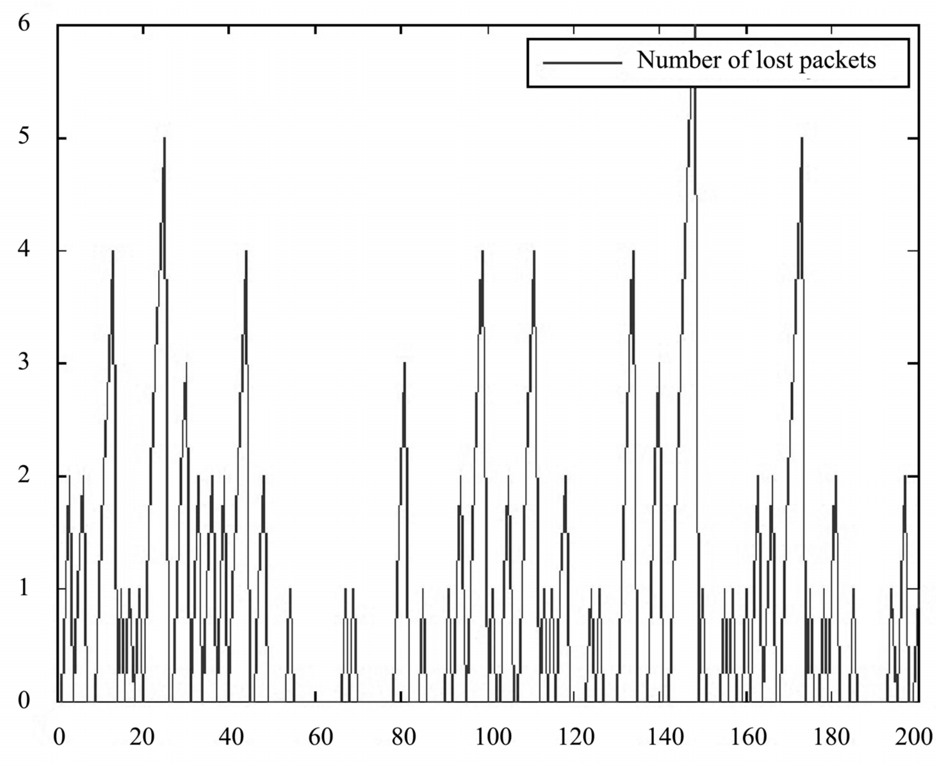

Fig. 5. The number of data packets lost.

Choose $m = 6$ kg, $L = 200$ mm, $I = 0.24$ kg$\times$m^2 as the parameter values of the system (12). Then the matrices of the discrete-time system are defined by

$$A = \begin{bmatrix} 1.0000 & 0.0200 & 0 & 0 \\ 0 & 1.0000 & 0 & 0 \\ -0.0020 & -0.0000 & 1.0000 & 0.0200 \\ -0.1962 & -0.0020 & 0 & 1.0000 \end{bmatrix},$$

$$B = \begin{bmatrix} 0.0008 \\ 0.0833 \\ 0.0002 \\ 0.0166 \end{bmatrix}.$$

According to the paper [14], the system (12) is instable under the above parameters.

Set $N = 5$ as the buffer size for system design. In the general case, the buffer size should be selected depending on the predicted number of sequentially lost data packets and the capabilities of the computing system. If the predicted number of lost data packets is exceeded, the system may leave the stable trajectory.

Figure 4 presents the trajectories of the stabilized system.

In addition, we illustrate the distribution of the number of data packets lost, see Fig. 5.

Clearly, the system is stabilized despite many data packets lost.

ACKNOWLEDGMENTS

This work was supported in part by the Russian Foundation for Basic Research, project no. 13-08-01092_a, and the Ministry of Education and Science of the Russian Federation, Federal Target Program "Personnel" (agreement no. 8846).

REFERENCES

1. Zhuchkov, R.N. and Pakshin, P.V., Stabilizing Networked Control of Linear Discrete Systems with Packet Dropouts, *Autom. Remote Control*, 2013, vol. 74, no. 3, pp. 529–536.

2. Bemporad, A. and Morari, M., Control of Systems Integrating Logic, Dynamics, and Constraints, *Automatica*, 1999, vol. 35, pp. 407–427.

3. Berber, R., *Methods of Model Based Process Control*, NATO ASI Series E: Applied Sciences, vol. 293, Dortrecht: Kluwer, 1995.

4. Bitmead, R.R., Gevers, M., and Wertz, V., *Adaptive Optimal Control. The Thinking Man's GPC*, Sydney: Prentice-Hall, 1990.

5. Camacho, E.F. and Bordons, C., *Model Predictive Control in the Process Industry*, New York: Springer, 1995.

6. Clarke, D.W., Mohtadi, C., and Tuffs, P.S., Generalized Predictive Control. I. The Basic Algorithm, *Automatica*, 1987, vol. 23, pp. 137–148.

7. Clarke, D.W., Mohtadi, C., and Tuffs, P.S., Generalized Predictive Control. II. Extensions and Interpretations, *Automatica*, 1987, vol. 23, pp. 149–160.

8. Clarke, D.W., *Advances in Model-Based Predictive Control*, Oxford: Oxford Univ. Press, 1994.

9. Cutler, C.R. and Ramaker, B.L., Dynamic Matrix Control—A Computer Control Algorithm, *AIChE 86th National Meeting*, Houston, 1979.

10. Cutler, C.R. and Ramaker, B.L., Dynamic Matrix Control—A Computer Control Algorithm, *Joint Automatic Control Conf.*, San Francisco, 1980.

11. Ding, B., Stabilization of Linear Systems over Networks with Bounded Packet Loss and Its Use in Synthesizing Model Predictive Control, *Proc. 8th. Int. Conf. on Control and Automation*, 2010, pp. 2258–2263.

12. Garcia, C.E., Prett, D.M., and Morari, M., Model Predictive Control: Theory and Practice—A Survey, *Automatica*, 1989, vol. 25, pp. 335–348.

13. Grüne, L., Pannek, J., and Worthmann, K., A Networked Unconstrained Nonlinear MPC Scheme, in *Proc. Eur. Control Conf.*, Budapest, 2009, pp. 371–376.

14. Hua Minh-Duc, Hamel, T., and Morin, P., Introduction to Feedback Control of Underactuated VTOL Vehicles: A Review of Basic Control Design Ideas and Principles, *Control Systems, IEEE*, 2013, vol. 33, no. 1, pp. 61–75.

15. Li, Z.J., Sun, D.H., Shi, Y.T., et al., A Stabilizing Model Predictive Control for Network Control System with Data Packet Dropout, *J. Control Theory Appl.*, 2009, vol. 7, pp. 281–284.

16. Martin Sanchez, J.M. and Rodellar, J., *Adaptive Predictive Control*, International Series in Systems and Control Engineering, New York: Prentice-Hall, 1996, pp. 263–283.

17. Bemporad, A. and Morari, M., Robust Model Predictive Control: A Survey, in *Robustness in Identification and Control*, 1999, vol. 245, pp. 207–226.

18. Nguyen, Q.T. and Vesely, V., Design of Robust Networked Predictive Control Systems with Packet Loss, *Preprints of 4th IFAC Nonlinear Model Predictive Control Conf.*, 2012, pp. 362–357.

19. Richalet, J., Rault, A., Testud, J.L., et al., Model Predictive Heuristic Control: Applications to Industrial Processes, *Automatica*, 1978, vol. 14, pp. 413–428.

20. Soeterboek, R., *Predictive Control—A United Approach*, Int. Series in Systems and Control Engineering, New York: Prentice-Hall, 1992.