Original In: R. Silhavy et al. (eds.), Automation Control Theory Perspectives in Intelligent Systems, Advances in Intelligent Systems and Computing 466, Springer International Publishing Switzerland, 2016. DOI 10.1007/978-3-319-33389-2_2

Modified discrete LQ control algorithm for situations with the scan period variance

Jan Cvejn

University of Pardubice, Faculty of Electrotechnics and Informatics, Studentská 95, 532 10, Pardubice, Czech Republic

E-mail: jan.cvejn@upce.cz

Abstract: Computer-based control systems, especially if they run under general-purpose operating systems, often exhibit variance of the scan period of processing inputs and outputs. Although this fact is usually not taken into account when discrete control algorithms are used, it can cause worse performance of the control loop in comparison to the theoretical case. In this paper we describe a modified discrete LQ control algorithm that takes disturbances of the scan period into account and partially compensates their influence. We also show that such a controller can be implemented even on low-performance hardware platforms, if they are equipped with a sufficient amount of memory.

Keywords: Optimal control, LQ controller, Linear systems, Discrete control

1 Introduction

In the control theory analysis and design of control algorithms in the continuous-time domain and in the discrete-time domain are studied separately. Continuous approach is natural for modelling and analysis of real processes and will be always used for designing controllers on the basis of analog components. Discrete approach seems to be natural for technical implementation of control algorithms on microprocessor-based platforms.

Discrete control algorithms rely upon constant period of processing inputs and outputs. However, constant scan period is often not fully guaranteed in real situations. This phenomenon can occur due to handling asynchronous hardware events in computer systems. Although this problem is typical for general-purpose multitasking operating systems, even the most robust hardware platforms such as PLCs exhibit scan variance. A similar situation could occur at remote control, where the measurements and the control signal are transported over a communication network.

Irregularities of the scan period cause worse performance of the control loop in comparison to the theoretical case. This influence can be neglected if the irregularities

Original In: R. Silhavy et al. (eds.), Automation Control Theory Perspectives in Intelligent Systems, Advances in Intelligent Systems and Computing 466, Springer International Publishing Switzerland, 2016. DOI 10.1007/978-3-319-33389-2_2

occur rarely and the system time constants are large in comparison to the scan period. In the other cases the influence on the closed-loop dynamics can be significant.

In this paper, we describe a modification of the classical linear-quadratic (LQ) discrete control algorithm taking into account irregularities of the scan period. We show that the effect of the scan variance can be partially compensated by mathematical means if a hybrid control law is used, working at discrete steps, but using a continuous-time model for the determination of the control output. In this way the control reliability and performance can be enhanced, especially at time-critical applications.

This problem has been studied already in [6] in a more general form as a stochastic control problem, considering also the optimal estimation. In this paper only the controller part is discussed, but an extended model of the scan period disturbances, which better corresponds to some real situations, is considered. This modification requires a corresponding extension of the control algorithm. The determination of the control action at each step is still not a time-consuming operation, although the enhanced control algorithm needs a table of data stored in the controller memory, which is initialized at the design phase.

2 Motivation

Consider a continuous time-invariant linear system

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \tag{1}$$

where the dimensions of $\mathbf{x}$ and $\mathbf{u}$ are n and m, respectively, $n \geq m$. $\mathbf{A}$ and $\mathbf{B}$ are known matrices of corresponding dimensions. We are looking for a control history such that

$$J = \frac{1}{2}\int_0^\infty \mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t) + \mathbf{u}^T(t)\mathbf{R}\mathbf{u}(t)\, dt \to \min \tag{2}$$

where $\mathbf{Q}$, $\mathbf{R}$ are given symmetric positive definite matrices. We assume that the current state $\mathbf{x}(t)$, or its estimate, is known.

Although the system nature is continuous, we consider that the measurements and the control actions are taken at discrete-time steps $t_0 < t_1 < \dots$. Although the scan period T is assumed to be known and constant, due to external factors the actual difference $t_{i+1} - t_i$ can fluctuate. We assume that each scan is provided with its time mark, which can be usually easily technically realized. Then, from the known sequence $t_1, t_2, \dots, t_k$ of the previous scan instants it is possible to estimate the future sequence $\overline{t}_{k+1}, \overline{t}_{k+2}, \overline{t}_{k+3}, \dots$, which is considered to be equidistant, i.e.

$$\overline{t}_{k+2} - \overline{t}_{k+1} = \overline{t}_{k+3} - \overline{t}_{k+2} = \dots = \overline{T} \; . \tag{3}$$

Original In: R. Silhavy et al. (eds.), Automation Control Theory Perspectives in Intelligent Systems, Advances in Intelligent Systems and Computing 466, Springer International Publishing Switzerland, 2016. DOI 10.1007/978-3-319-33389-2_2

But note that $\overline{T} = T$ need not hold in general, because the response can be delayed permanently in some time interval. Although the scan interrupts are generated by hardware clock with the period T, which does not depend on the previous scan instants, the delay can be caused by omitting some scan instants, e.g. due to service of hardware events in the operating system (this problem is discussed below in more detail). This delay can be detected from the sequence of the past measurements $t_i, i \le k$ and this information also can be used to estimate the future sequence (3), i.e. the expected period $\overline{T}$.

There are several possible ways how to estimate the next scan instant $\overline{t}_{k+1}$. In [6] a simplified model was used, which assumed $\overline{T} = T$. If we denote t_c the instant of the next time interrupt closest to t_k, in most cases the following estimate seems to be more adequate:

$$\overline{t}_{k+1} = t_c + \overline{T} - T . \tag{4}$$

But if $t_c - t_k \le \alpha T$, where $\alpha \in (0,1)$ is a known parameter, the control algorithm should be designed to omit the next scan, i.e. in this case

$$\overline{t}_{k+1} = \left(t_c + \overline{T} - T\right) + T = t_c + \overline{T} . \tag{5}$$

This behavior indeed depends on implementation of the controller on given platform. For instance, $t_c - t_k \le 0.25T$ may indicate that the processor looses the ability to process the hardware events at given moment and in such a case the control algorithm should be designed to drop the following scan and wait until the next one to prevent the system from overloading, which would affect the overall functionality. This modification is especially needed in the case of multi-tasking operating systems, where the control algorithm is a high-priority task, and if processing of the measurements and computation of the control action is a time-consuming operation within the interval $[t_k, t_{k+1}]$.

3 Optimal Control Algorithm

To summarize the considerations of the previous section, at given moment t_k we assume that the estimates of the next scan instant $\overline{t}_{k+1} \ge t_c$ and the future scan period $\overline{T} \ge T$ are known, in general different from the next hardware clock instant t_c and the clock period T.

The criterion (2) value from $t = t_k$ can be expressed as

Original In: R. Silhavy et al. (eds.), Automation Control Theory Perspectives in Intelligent Systems, Advances in Intelligent Systems and Computing 466, Springer International Publishing Switzerland, 2016. DOI 10.1007/978-3-319-33389-2_2

$$J\left(t_k\right) = \frac{1}{2}\sum_{i=k}^{N}\int_{t_i}^{t_{i+1}} \mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t) + \mathbf{u}^T(t)\mathbf{R}\mathbf{u}(t)\, dt \tag{6}$$

where $N \to \infty$ and $\mathbf{u}(t)$ is constant in each interval $\left[t_i, t_{i+1}\right)$, $i \geq k$. Note that unlike common practice the criterion includes the information about complete state history in $\left[0, t_f\right]$, and not only about the values at the discrete points t_i.

Denote for simplicity $\mathbf{x}_k = \mathbf{x}(t_k)$ and $\mathbf{u}_k = \mathbf{u}(t_k)$. For given $\mathbf{x}_k$ and $t > t_k$

$$\mathbf{x}(t) = \mathbf{\Phi}(t - t_k)\mathbf{x}_k + \mathbf{\Psi}(t - t_k)\mathbf{u}_k \tag{7}$$

holds, where

$$\mathbf{\Phi}\left(h\right) = e^{\mathbf{A}h}, \quad \mathbf{\Psi}\left(h\right) = \int_0^h \mathbf{\Phi}(h-\tau)\mathbf{B}\, d\tau = \int_0^h \mathbf{\Phi}(\tau)\, d\tau\, \mathbf{B}\,. \tag{8}$$

The term $\mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t)$ for given $\mathbf{x}_k$ and $t > t_k$ can be written using (8) as

$$\mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t) = \left[\mathbf{x}_k^T, \mathbf{u}_k^T\right]\begin{bmatrix}\mathbf{\Phi}^T(t-t_k)\\ \mathbf{\Psi}^T(t-t_k)\end{bmatrix}\mathbf{Q}\left[\mathbf{\Phi}(t-t_k), \mathbf{\Psi}(t-t_k)\right]\begin{bmatrix}\mathbf{x}_k\\ \mathbf{u}_k\end{bmatrix}. \tag{9}$$

Let us define

$$\begin{aligned}\mathbf{U}(h) &= \int_0^h\left(\begin{bmatrix}\mathbf{\Phi}^T(\tau)\\ \mathbf{\Psi}^T(\tau)\end{bmatrix}\mathbf{Q}\left[\mathbf{\Phi}(\tau), \mathbf{\Psi}(\tau)\right] + \begin{bmatrix}0 & 0\\ 0 & \mathbf{R}\end{bmatrix}\right)dh = \\ &= \int_0^h\begin{bmatrix}\mathbf{\Phi}^T(\tau)\mathbf{Q}\mathbf{\Phi}(\tau) & \mathbf{\Phi}^T(\tau)\mathbf{Q}\mathbf{\Psi}(\tau)\\ \mathbf{\Psi}^T(\tau)\mathbf{Q}\mathbf{\Phi}(\tau) & \mathbf{\Psi}^T(\tau)\mathbf{Q}\mathbf{\Psi}(\tau) + \mathbf{R}\end{bmatrix}d\tau = \begin{bmatrix}\mathbf{U}_{11}\left(h\right) & \mathbf{U}_{12}\left(h\right)\\ \mathbf{U}_{21}\left(h\right) & \mathbf{U}_{22}\left(h\right)\end{bmatrix}\end{aligned} \tag{10}$$

where $\mathbf{U}_{21}\left(h\right) = \mathbf{U}_{12}^T\left(h\right)$. Using (10) we can write

$$J\left(t_k\right) = \frac{1}{2}\left[\mathbf{x}_k^T\ \mathbf{u}_k^T\right]\mathbf{U}\left(\overline{t}_{k+1} - t_k\right)\begin{bmatrix}\mathbf{x}_k\\ \mathbf{u}_k\end{bmatrix} + J\left(\overline{t}_{k+1}\right) \tag{11}$$

where

$$J\left(\overline{t}_{k+1}\right) = \frac{1}{2}\sum_{i=k+1}^{N}\left[\mathbf{x}_i^T\ \mathbf{u}_i^T\right]\mathbf{U}\left(\overline{T}\right)\begin{bmatrix}\mathbf{x}_i\\ \mathbf{u}_i\end{bmatrix}. \tag{12}$$

Let us denote $J^*(t_k)$ the minimal value of $J(t_k)$. By application of Bellman's optimality principle [1]-[3] we obtain

Original In: R. Silhavy et al. (eds.), Automation Control Theory Perspectives in Intelligent Systems, Advances in Intelligent Systems and Computing 466, Springer International Publishing Switzerland, 2016. DOI 10.1007/978-3-319-33389-2_2

$$J^*\left(t_k\right)=\min_{u_k}\left\{\frac{1}{2}\left[\mathbf{x}_k^T\ \mathbf{u}_k^T\right]\mathbf{U}\left(\bar{t}_{k+1}-t_k\right)\begin{bmatrix}\mathbf{x}_k\\ \mathbf{u}_k\end{bmatrix}+J^*\left(\bar{t}_{k+1}\right)\right\} \tag{13}$$

where

$$J^*\left(\bar{t}_{k+1}\right)=\min_{\{u_{k+1},\dots,u_N\}}\left\{\frac{1}{2}\sum_{i=k+1}^{N}\left[\mathbf{x}_i^T\ \mathbf{u}_i^T\right]\mathbf{U}\left(\bar{T}\right)\begin{bmatrix}\mathbf{x}_i\\ \mathbf{u}_i\end{bmatrix}\right\} \tag{14}$$

subject to the dynamic constraints

$$\mathbf{x}_{i+1}=\mathbf{\Phi}\left(\bar{T}\right)\mathbf{x}_i+\mathbf{\Psi}\left(\bar{T}\right)\mathbf{u}_i,\ \ i\ge k+1\,. \tag{15}$$

Equations (14) and (15) formulate a discrete deterministic linear-quadratic optimal control problem. The minimal cost-function value of this problem for $N\to\infty$ is in the form

$$\begin{aligned}J^*\left(\bar{t}_{k+1}\right)&=\frac{1}{2}\mathbf{x}_{k+1}^T\mathbf{S}\,\mathbf{x}_{k+1}=\\&=\frac{1}{2}\left[\mathbf{x}_k^T,\ \mathbf{u}_k^T\right]\begin{bmatrix}\mathbf{\Phi}^T(\bar{t}_{k+1}-t_k)\\ \mathbf{\Psi}^T(\bar{t}_{k+1}-t_k)\end{bmatrix}\mathbf{S}\left[\mathbf{\Phi}(\bar{t}_{k+1}-t_k),\ \mathbf{\Psi}(\bar{t}_{k+1}-t_k)\right]\begin{bmatrix}\mathbf{x}_k\\ \mathbf{u}_k\end{bmatrix}\end{aligned} \tag{16}$$

where $\mathbf{S}$ is a positive-definite symmetric matrix. If we define

$$\mathbf{Z}(h)=\begin{bmatrix}\mathbf{Z}_{11}(h) & \mathbf{Z}_{12}(h)\\ \mathbf{Z}_{21}(h) & \mathbf{Z}_{22}(h)\end{bmatrix}=\mathbf{U}(h)+\begin{bmatrix}\mathbf{\Phi}^T(h)\\ \mathbf{\Psi}^T(h)\end{bmatrix}\mathbf{S}\left[\mathbf{\Phi}(h),\ \mathbf{\Psi}(h)\right] \tag{17}$$

where $\mathbf{Z}_{21}\left(h\right)=\mathbf{Z}_{12}^T\left(h\right)$, the minimizer of $J\left(t_k\right)$ for fixed $\bar{T}$ can be written as

$$\begin{aligned}\mathbf{u}_k^*&=\arg\min_{\mathbf{u}_k}\left\{\frac{1}{2}\left[\mathbf{x}_k^T\ \mathbf{u}_k^T\right]\mathbf{U}\left(\bar{t}_{k+1}-t_k\right)\begin{bmatrix}\mathbf{x}_k\\ \mathbf{u}_k\end{bmatrix}+\frac{1}{2}\mathbf{x}_{k+1}^T\mathbf{S}\,\mathbf{x}_{k+1}\right\}=\\&=\arg\min_{\mathbf{u}_k}\left\{\frac{1}{2}\left[\mathbf{x}_k^T\ \mathbf{u}_k^T\right]\mathbf{Z}\left(\bar{t}_{k+1}-t_k\right)\begin{bmatrix}\mathbf{x}_k\\ \mathbf{u}_k\end{bmatrix}\right\}.\end{aligned} \tag{18}$$

The solution can be easily found by differentiation in the form

$$\mathbf{u}_k^*=-\mathbf{Z}_{22}^{-1}\left(\bar{t}_{k+1}-t_k\right)\mathbf{Z}_{21}\left(\bar{t}_{k+1}-t_k\right)\mathbf{x}_k=\mathbf{C}\left(\bar{t}_{k+1}-t_k\right)\mathbf{x}_k\,. \tag{19}$$

Note that the matrix inversion in (19) always exists, since $\mathbf{Z}_{22}\left(h\right)$ is positive definite.

If we put $\bar{t}_{k+1}-t_k=\bar{T}$ in (13) and (19), we have to obtain for optimal $\mathbf{u}_k$

Original In: R. Silhavy et al. (eds.), Automation Control Theory Perspectives in Intelligent Systems, Advances in Intelligent Systems and Computing 466, Springer International Publishing Switzerland, 2016. DOI 10.1007/978-3-319-33389-2_2

$$J^*\left(t_k\right)=\frac{1}{2}\mathbf{x}_k^T\mathbf{S}\,\mathbf{x}_k\,. \tag{20}$$

This condition can be used to determine $\mathbf{S}$. By substituting (20) into (13) and (16) we obtain:

$$\begin{aligned}
J^*\left(t_k\right)&=\frac{1}{2}\left[\mathbf{x}_k^T\ \mathbf{u}_k^T\right]\mathbf{Z}\left(\bar{T}\right)\begin{bmatrix}\mathbf{x}_k\\ \mathbf{u}_k\end{bmatrix}=\\
&=\frac{1}{2}\mathbf{x}_k^T\left[\mathbf{I}\quad -\mathbf{Z}_{22}^{-1}\left(\bar{T}\right)^{-1}\mathbf{Z}_{21}\left(\bar{T}\right)\right]\mathbf{Z}\left(\bar{T}\right)\begin{bmatrix}\mathbf{I}\\ -\mathbf{Z}_{22}^{-1}\left(\bar{T}\right)^{-1}\mathbf{Z}_{21}\left(\bar{T}\right)\end{bmatrix}\mathbf{x}_k=\\
&=\frac{1}{2}\mathbf{x}_k^T\left[\mathbf{I}\quad -\mathbf{Z}_{22}^{-1}\left(\bar{T}\right)^{-1}\mathbf{Z}_{21}\left(\bar{T}\right)\right]\begin{bmatrix}\mathbf{Z}_{11}\left(\bar{T}\right)-\mathbf{Z}_{12}\left(\bar{T}\right)\mathbf{Z}_{22}^{-1}\left(\bar{T}\right)\mathbf{Z}_{21}\left(\bar{T}\right)\\ \mathbf{0}\end{bmatrix}\mathbf{x}_k=\\
&=\frac{1}{2}\mathbf{x}_k^T\left(\mathbf{Z}_{11}\left(\bar{T}\right)-\mathbf{Z}_{12}\left(\bar{T}\right)\mathbf{Z}_{22}^{-1}\left(\bar{T}\right)\mathbf{Z}_{21}\left(\bar{T}\right)\right)\mathbf{x}_k\,.
\end{aligned} \tag{21}$$

This shows that

$$\mathbf{S}=\mathbf{Z}_{11}\left(\bar{T}\right)-\mathbf{Z}_{12}\left(\bar{T}\right)\mathbf{Z}_{22}^{-1}\left(\bar{T}\right)\mathbf{Z}_{21}\left(\bar{T}\right) \tag{22}$$

must hold. By substituting for $\mathbf{Z}_{ij}\left(\bar{T}\right)$ from (17) it is easily seen that (22) can be rewritten into an algebraic matrix Riccati equation [2], [3].

4 Implementation of the Controller

Obtained expressions are indeed rather complicated to be computed at each control step. The controller matrix $\mathbf{C}\left(\bar{t}_{k+1}-t_k\right)$ in (19) is dependent on the expected distance of the next scan $\bar{t}_{k+1}-t_k$ and on $\mathbf{S}$, while $\mathbf{S}$ depends on $\bar{T}$.

For given $\bar{T}$ the solution to the Riccati equation (22) can be obtained off-line as a part of the controller design. Denote $\mathbf{Z}\left(\bar{T};\mathbf{S}_i\right)$ the value of $\mathbf{Z}\left(\bar{T}\right)$ for $\mathbf{S}=\mathbf{S}_i$. A basic method of obtaining **S** consists in solving

$$\mathbf{S}_{i+1}=\mathbf{Z}_{11}\left(\bar{T};\mathbf{S}_i\right)-\mathbf{Z}_{12}\left(\bar{T};\mathbf{S}_i\right)\mathbf{Z}_{22}^{-1}\left(\bar{T};\mathbf{S}_i\right)\mathbf{Z}_{21}\left(\bar{T};\mathbf{S}_i\right) \tag{23}$$

iteratively until $\left\|\mathbf{S}_{i+1}-\mathbf{S}_i\right\|<\varepsilon$, where ε is sufficiently small [2]. More sophisticated methods, preferable both from numerical point of view and for improved efficiency, were proposed in [4] and [5].

Original In: R. Silhavy et al. (eds.), Automation Control Theory Perspectives in Intelligent Systems, Advances in Intelligent Systems and Computing 466, Springer International Publishing Switzerland, 2016. DOI 10.1007/978-3-319-33389-2_2

If we assume $\bar{T} \in [T, T_{\max}]$, $T_{\max} \geq 2T$, the solutions to (23) can be obtained for the values in this interval with a sufficiently small discrete step $\Delta\bar{T}$ at the initialization phase. These solutions can be stored in the controller memory and in the real-time operation the values corresponding to the current estimate of $\bar{T}$ are being picked from this table. Although it may seem that the initialization phase could be computationally very demanding, if the value of $\mathbf{S}$ is known of some $\bar{T}$, it can be used as a very good estimate of $\mathbf{S}$ for the iterative computation based on (23) corresponding to $\bar{T} + \Delta\bar{T}$, because $\mathbf{S}$ depends only moderately on $\bar{T}$. Therefore, the computation of the whole table of the values of $\mathbf{S}$ for $\bar{T} \in [T, T_{\max}]$ is not a time-demanding operation.

In the same way, the values of the matrices $\mathbf{\Phi}(h), \mathbf{\Psi}(h)$ and $\mathbf{U}(h)$ have to be computed for $0 < h \leq h_{\max}$, where $h_{\max} \geq 2T$ is known, with a sufficiently small discrete step of h and stored in the controller memory. Formally written, it is needed to solve the following set of differential equations in the interval $h \in [0, h_{\max}]$:

$$\frac{d}{dh}\mathbf{\Phi}(h) = \mathbf{A}\mathbf{\Phi}(h) \tag{24}$$

$$\frac{d}{dh}\mathbf{\Psi}(h) = \mathbf{\Phi}(h)\mathbf{B} \tag{25}$$

$$\frac{d}{dh}\mathbf{U}(h) = \begin{bmatrix} \mathbf{\Phi}^T(\tau)\mathbf{Q}\mathbf{\Phi}(\tau) & \mathbf{\Phi}^T(\tau)\mathbf{Q}\mathbf{\Psi}(\tau) \\ \mathbf{\Psi}^T(\tau)\mathbf{Q}\mathbf{\Phi}(\tau) & \mathbf{\Psi}^T(\tau)\mathbf{Q}\mathbf{\Psi}(\tau) + \mathbf{R} \end{bmatrix} \tag{26}$$

with the initial conditions

$$\mathbf{\Phi}(0) = \mathbf{I}_n, \quad \mathbf{\Psi}(0) = \mathbf{0}, \quad \mathbf{U}(0) = \mathbf{0}. \tag{27}$$

It is important to mention here that the controller implementation is significantly more efficient in the simplified version where $\bar{T} = T$ is fixed. In this case $\mathbf{S}$ is constant and the controller matrix $\mathbf{C}(h)$ in (19) can be computed in forward and stored, so the matrix inversion in (19) need not be computed in real time.

5 Example

Consider the double-integrator system in the form (1) where

$$\mathbf{A} = \begin{bmatrix} 0 & 0 \\ 2 & 0 \end{bmatrix}, \; \mathbf{B} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{28}$$

Original In: R. Silhavy et al. (eds.), Automation Control Theory Perspectives in Intelligent Systems, Advances in Intelligent Systems and Computing 466, Springer International Publishing Switzerland, 2016. DOI 10.1007/978-3-319-33389-2_2

with the initial condition

$$\mathbf{x}(0) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}. \tag{29}$$

The criterion (6) parameters were chosen as

$$\mathbf{Q} = \mathbf{I}_2, \quad \mathbf{R} = 1. \tag{30}$$

The scan period is $T = 2s$, but each fourth scan is delayed of 50% and the following scan is omitted. In the initialization phase it was needed to obtain the solution to (24)-(26) for $h \in [0, 2T]$ and to solve (23) for the sequence of values of $\overline{T} \in [T, 2T]$ with the step size of $\Delta\overline{T} = 0.01$. The initialization was not a time-demanding operation.

Figure 1 shows the response of the system when the standard LQ controller is used, i.e. if $\overline{t}_{k+1} = t_k + T$ and $\overline{T} = T$, while Fig. 2 shows the responses if the modified controller (19) is used for $\overline{T} = 5T/4$. This estimate of $\overline{T}$ corresponds to the fact that each fifth scan is omitted.

It can be seen that the control loop behavior was significantly enhanced, although a similar effect indeed could be achieved by decreasing the scan period, if it was technically possible. Even bigger differences can be observed if the controller is equipped with the optimal state estimator, which can be also designed so that it takes the variance of the scan period into account, as described in [6].

Fig. 3 shows that the responses obtained for the simplified version where $\overline{T} = T$ are similar to the full version. This indicates that the simplified version, which is preferable from the implementation point of view, would be usually sufficient and recommendable for practical use.

Original In: R. Silhavy et al. (eds.), Automation Control Theory Perspectives in Intelligent Systems, Advances in Intelligent Systems and Computing 466, Springer International Publishing Switzerland, 2016. DOI 10.1007/978-3-319-33389-2_2

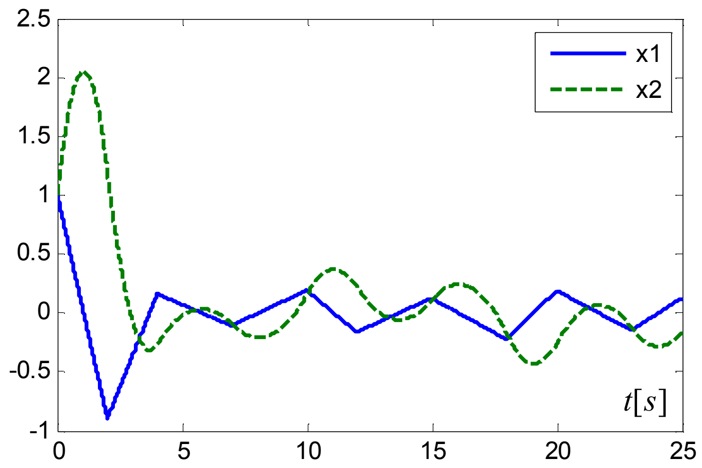

Fig. 1. The history of the state variables – standard LQ controller

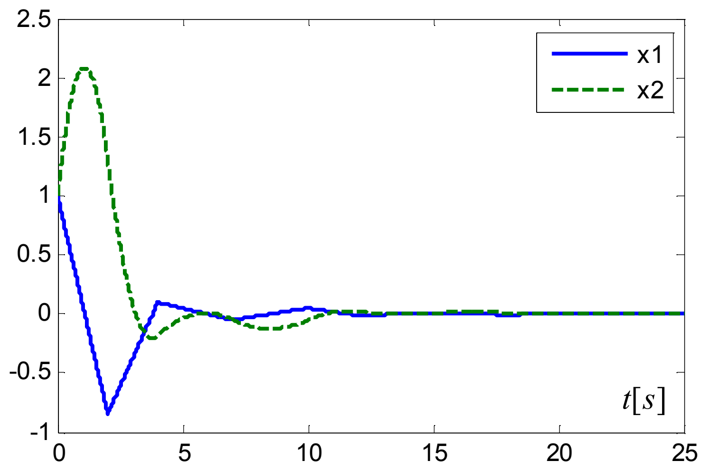

Fig. 2. The history of the state variables – the modified LQ controller, full version

Original In: R. Silhavy et al. (eds.), Automation Control Theory Perspectives in Intelligent Systems, Advances in Intelligent Systems and Computing 466, Springer International Publishing Switzerland, 2016. DOI 10.1007/978-3-319-33389-2_2

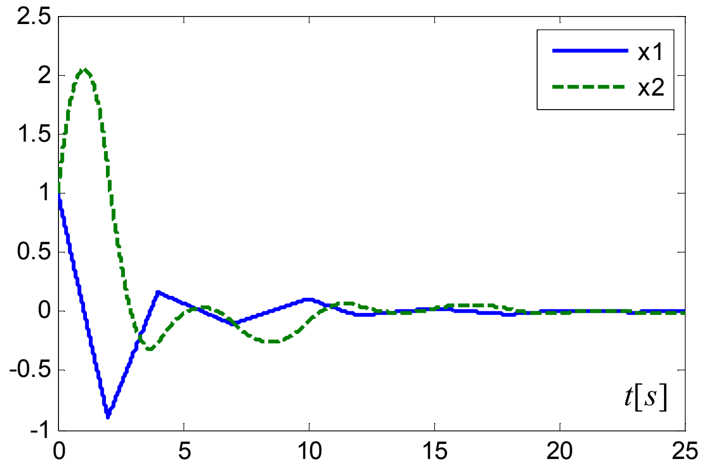

Fig. 3. The history of the state variables – the modified LQ controller, simplified version

6 Conclusions

The modification of the LQ control algorithm described in this paper tries to reduce the influence of the scan-period variance, which can occur in computer-based control systems, on the closed-loop control performance. Although obtained expressions for the control output may be rather complicated to be computed in real time, if sufficient memory in the control system is available, it is possible to carry out most of these computations in forward and the determination of the control output is not a complicated or time-consuming operation. Consequently, such a control algorithm can be implemented even on low-performance hardware platforms. In the simplified version, which seems to be sufficient for practical purposes, the computation of the control action is as demanding as a matrix-vector product, like in the case of standard LQ control algorithm. However, the controller has to be equipped with a sufficient amount of memory to store a table of the parameter-dependent control matrices, generated with a sufficiently short discrete step.

References

1. Bryson, A.E. , Ho, Y.C.: Applied Optimal Control. Hemisphere Corp., New York (1975)
2. Stengel, R.F.: Stochastic Optimal Control: Theory and Application. Wiley, New York, (1986)
3. Fleming, W.H., Rishel, R.W.: Deterministic and Stochastic Optimal Control. Springer-Verlag, Berlin (1995)

Original In: R. Silhavy et al. (eds.), Automation Control Theory Perspectives in Intelligent Systems, Advances in Intelligent Systems and Computing 466, Springer International Publishing Switzerland, 2016. DOI 10.1007/978-3-319-33389-2_2

4. Pappas, T., Laub, A. J., Sandell, N.R.Jr.: On the Numerical Solution of the Discrete-Time Algebraic Riccati Equation. IEEE Transactions on Automatic Control 25, 631-641 (1980)
5. Kleinman, D.L.: Stabilizing a Discrete, Constant Linear System with Application to Iterative Methods for Solving the Riccati Equation. IEEE Transactions on Automatic Control, 19, 252-254 (1974)
6. Cvejn, J.: Compensation of the Scan-Period Irregularities in LQG Control Systems. Archives of control sciences. vol. 18, 3 (2008)