Linear digital controller for high-speed dynamical system

P. Płatek* W. Grega**

* Department of Automatics, AGH University of Science and Technology Al. Mickiewicza 30, 30-059 Cracow, Poland (e-mail: ppi@agh.edu.pl).
** Department of Automatics, AGH University of Science and Technology Al. Mickiewicza 30, 30-059 Cracow, Poland (e-mail: wgr@agh.edu.pl).

Abstract: Traditionally, digital control algorithms are designed without the consideration of their real-time implementation details. Designers do not care very much about real-time, digital implementations of control algorithms, such as closed-loop execution time. The digital controller design methodology often assumes that the controller is implemented exactly, even though in reality a control law can only be realized in finite precision. Due to the finite word length effect, a controller implementation may degrade the designed closed-loop performance or even destabilize the designed stable closed-loop system. If the controller digital implementation structure is not carefully chosen. The closed-loop quality measures of the discrete finite-precision approximation of the digital controller are investigated in this paper. The magnetic levitation control example demonstrated that the proposed design procedure yields more efficient discrete controller approximation.

Keywords: Real-time control, digital control, stability, magnetic levitation.

1. INTRODUCTION

Digital control systems are commonly used for the control of physical processes. In the case of digital linear controllers, a design process consists of developing a continuous-time controller for a continuous model of the process, discretisation of the “continuous” controller and digital implementation of the resulting discrete controller. The performance of a digital control system, besides the sampling period, depends on several variables, such as the control loop execution time and the effects of the discretisation of the controller model. Therefore, a number of authors claim (Grega, 2008; Chen et al., 1999) that these three parameters need particular attention from the real-time, computer implementation of control algorithms. Digital control algorithms are often designed without the consideration of their real-time implementation details. Designers do not care very much about real-time, the digital implementations of control algorithms, such as the closed-loop execution time of the implementation platform. In many cases they do not consider control timing constraints and the typical proposed solution is: “buy a faster computer”. The classical digital controller design methodology often assumes that the controller is implemented exactly, even though in reality a control law can only be realized in finite precision. In the case when the control frequency strongly exceeds the bandwidth of the system (by 50 times and more), digital controllers can lose their stability because of numerical errors. The simplest way of avoiding such a situation is by lengthening the data words used for computation. Such an approach can unfortunately lead to an increase in the overall cost of the control system (often a substantial one). There are also situations when the extension of the word requires the total reconstruction of controller hardware. Another way of improving the properties of digital high-speed controllers is the application of alternative discrete operators and the choice of a proper realisation method (Swider, 2003; Płatek, 2007). Both of these aspects are described in this paper and illustrated with an example of a linear controller for a magnetic levitation control system.

2. DESIGN PROBLEMS FOR DIGITAL CONTROL SYSTEMS

Figure 1, presents the block diagram of a typical digital control system. In this system, the real-time control tasks are executed according to the diagram in Fig. 2 (Piłatek and Grega, 2009).

![Fig. 1. Model of digital control system with digital discrete controller and continuous plant]

SP—set point, A/D—analogue-digital converter, D/A—digital-analog converter, T_0—sampling period

* This research has received support from the AGH University of Science and Technology.
Fig. 2. Tasks in real time control system

Operations are performed cyclically per given time period $T_0$, (the sampling period), which can be understood as the controller discretisation period. After the initialisation of the control cycle the system performs three main operations:

- reading of the plant output (usually equivalent to the use and reading of a analog-digital converter),
- computation of the control value, according to a given algorithm,
- actualisation of the control at the plant input (usually it is the use of digital-analog converters).

The control task period for discrete-time control designs, beyond confirming to the Shannon theorem, can be selected following one of the various so called “engineering rules” (Franklin et al., 1994), depending on the desired performance of the closed loop control system and the estimation of the dynamics of the process to be controlled. The one rule is to choose this period according to the dominant pole positions of the continuous time process model. One accepted rule is that the control task frequency should be $a$ ($a > 1$, $a \in N$) times smaller than the period of the cut-off frequency, approximated in some reasonable way for the process model.

$$T_0^u = \frac{T_{\text{max}}}{a} \quad (1)$$

where:

$$T_{\text{max}} = \max(T_1, T_2, \ldots, T_n),$$

for:

$$A(s) = (T_1 s + 1)(T_2 s + 1) \ldots (T_n s + 1),$$

where:

$A(s)$—denominator of the process linear model.

There is some flexibility in the selection:

- $a = 10$, giving $T_0^u$ as the control task period for an “ideal” (not disturbed) control system, where modelling and identification errors, as well as time delays and variations of sampling periods are negligible,
- $a > 10$, giving $T_0^u$— the period guaranteeing the robust operation of the control system, if the system is under the influence of external and internal disturbances.

If we assume that the performance of the closed-loop control system is a strictly monotonic function of $T_0$ then any sampling period $T_0 < T_0^u$ improves the control performance. For $T_0 < T_0^u$ improvement is not observed. Finally, the control task period can be estimated as $T_0 \in [T_0^u, T_0^u]$.

The applied control platform (processor, peripherals hardware and operating systems) are characterized by a minimal (a shortest accessible) closed-loop execution time, estimated as $\tau_c = [\tau_{c1}, \tau_{c2}]$, where $\tau_{c1}$—is the lower boundary of the execution time for simple control algorithms, $\tau_{c2}$—is the execution time for complex control algorithms. The control algorithm is classified as “simple”, if the pseudocode of the controller task includes no more than 5-10 operations (loops are excluded). The examples of “simple” algorithms are incremental PID or state feedback controller. If the pseudocode of the controller includes more than 10 operations, or loops are included, then the algorithm is classified as “complex”. The examples of “complex” control algorithms are: time-optimal or model-reference controllers. Figure 3 presents 2-dimensional design space ($T_0, \tau_s$). The example presents the situation $\tau_s < T_0$ e.g. when the interval of controller execution times is lower than the interval of control task periods. We can move $T_0$ below $T_0$—see point $A'$ in Fig. 3. This opens the possibility of using a simplified controller design technique, known as “simulation design” (Swider, 2003). Using this method, a model for the controller is first developed in the continuous—time domain and then transformed into the discrete time domain, using some simple numerical approximation methods (Tustin approximation is the most popular).

Fig. 3. Two dimensional real-time control system design space

<table>
<thead>
<tr>
<th>$T_0^u$</th>
<th>$T_0^u$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_{\text{max}}$</td>
<td>$T_{\text{max}}$</td>
</tr>
<tr>
<td>$T_0^u$</td>
<td>$T_0^u$</td>
</tr>
</tbody>
</table>

Fig. 4. Control quality, sampling and discretisation effects

However, different controller discretisations possess different degrees of robustness to finite-precision implementation errors. If one maps the selected parameters of the controller (defined by the point $A'$ in Fig. 3) into control quality space (defined by the value of an objective function), then he will receive a different control system performance for different discretisations methods (Fig. 4). So, if taking into account finite-precision implementation considerations, then the design problem is to select an
optimal discretisations scheme for the given control law and control quality criteria.

3. FINITE WORD LENGTH EFFECT

Digital controller formula is usually the implementation of control algorithm e.g. a linear controller. Such a controller should be properly chosen for the system requirements and possibilities of control. The design of the controller and the selection of its parameters is usually described as a separate stage of the control system development. In the control of plants and physical processes, which are continuous in nature, we have two ways of discrete controller design (Grega, 2004).

(1) Design of a continuous-times controller for a continuous-times process model and the subsequent discretisation of the resulting controller.

(2) Discretisation of a “continuous” process model, and the design of a discrete controller using the modelling methods for discrete systems.

In this paper we consider the first of these two methods. It allows for the application of methods well known for continuous systems (Kaczorek, 1990). The most important disadvantage is the possibility of losing the stability or changing the dynamical properties of the controller through discretisation.

Discretisation of the continuous-time model of a controller consists of finding its equivalent in the domain of discrete time. For linear controllers it is convenient to use the complex variable domain and the description of the controller with a transfer function. In practical applications linear SISO (Single Input Single Output) controllers are frequently used. Their relatively low computational complexity and the need for only basic arithmetic operations make them feasible for implementation in systems with strict speed requirements or low computational capabilities.

A continuous, generalised linear SISO controller is given by the transfer function (2).

$$G_R(s) = \frac{U(s)}{E(s)} = \frac{b_0 s^n + \cdots + b_2 s^2 + b_1 s + b_0}{a_d s^d + \cdots + a_2 s^2 + a_1 s + a_0} \quad (2)$$

where:

- $n$ and $d$—are degrees of polynomials
- $a_i, b_j$—for $i = 1, 2, \ldots, d$ and $j = i = 1, 2, \ldots, n$ are the controller parameters chosen for control of given plant.

One of SISO controllers is PID controller, which is widely used in industrial applications. Most popular discretisation methods of SISO controllers are:

- application of ZOH (Zero Order Hold) on the input of continuous transfer function model (Step Invariant) (Middleton and Goodwin, 1990; Grega, 2004),
- methods using different types of numerical integration:
  - explicit Euler method (forward polygons) (Franklin et al., 1994; Grega, 2004),
  - implicit Euler method (backward polygons) (Grega, 2004),
  - trapezoidal rule, so called Tustin scheme (Grega, 2004),
- matching of zeros and poles.

The consequence of some of the approximation methods is the different mapping of the left open half plane of complex variable $s$ into the complex plane of $z$ variable. It can lead to a situation when from the discretisation of a stable continuous system we get an unstable discrete one. It is then the use of methods preserving stability (backward polygons or Tustin scheme) is suggested, or the verification of the stability of models discretised with a method that does not have this property e.g. forward polygon method.

An important matter that has to be considered at the stage of design is the influence of numerical errors on the operation of a discrete controller. The computation of control values are always performed with finite word lengths. In the case of the implementation of linear discrete controllers, the following operations are necessary: multiplication (signals by coefficients), addition (of signals or values of indirect computations) and $z^{-1}$ (shift by one sample) or $\delta^{-1}$ (numerical integration). Sources of numerical errors in those systems are (Gevers and Li, 1993; Swider, 2003, 2000a):

- quantisation of continuous signals by analog-digital converters—they depend on the resolution of used A/D converter,
- overflow errors resulting from too short word lengths. In practice, it occurs only in fixed point arithmetic systems. Overflow errors can be eliminated through the proper scaling of signals and controller coefficients, which can lead to additional rounding errors,
- rounding errors of arithmetic operations—addition, multiplication and digital integrator ($\delta^{-1}$ operator) (Middleton and Goodwin, 1990),
- quantisation errors of controller coefficients which are caused by writing the coefficients with finite words.

The influence of rounding and quantisation errors on the result of computations cannot be fully eliminated. Effects associated with those kinds of errors are called FWL (Finite Word Length) effects (Gevers and Li, 1993). The influence of these effects can be limited by the increase of word length and by changes in model structure. The change of word length is not always possible. In computer-based control systems, usually two or three lengths are available, and in simple microprocessor units there is only one. Relatively simple increases in precision are possible only in data types supported by a given architecture. Additional increases (above the precision supported by computer instructions) happen at the cost of a substantial increase in the number of instructions required for the computation of the model. In the case of control system realisation in dedicated structures e.g. with FPGA circuits, the word length can be chosen almost freely. However, it increases the usage of circuit resources, which can be interpreted as a rise in the cost of the controller realisation (Piątek, 2007). Another way of limiting the FWL influence on the control quality is the modification of the model structure. A more precise description of this problem is given in the further part of the paper.

In discrete control it is often the practice that overly short sampling times are used. This is called over-sampling. In many cases it is justified by the need to protect the control system from expected or unexpected disturbances or fac-
tors influencing its operation, for example inaccuracies in modelling or identification. It is however associated with the risk of the destabilisation of the discrete controller. Usually, with an increase of sampling frequency the coefficients of the controller model become larger. This is illustrated by the following example. Formula (3) illustrates the transfer function of the discrete PID controller achieved by the approximation of the continuous PID controller with a realisable derivative part through the backward polygon method [Piątek, 2007].

\[
G_{\text{back}}(z) = \frac{U(z)}{E(z)} = \frac{b_2z^2 + b_1z + b_0}{a_2z^2 + a_1z + a_0}
\]  

(3)

where:

\[
b_2 = \frac{K_pT_0^2 + (K_pT_1 + K_p)T_0 + K_pT_1 + K_d}{T_0 + T_1},
\]

\[
b_1 = \frac{(-K_p - K_p)T_0 - 2K_p - 2K_pT_1}{T_0 + T_1},
\]

\[
b_0 = \frac{K_pT_1 + K_d}{T_0 + T_1},
\]

\[a_2 = 1,\]

\[a_1 = \frac{-T_0 - 2T_1}{T_0 + T_1},\]

\[a_0 = \frac{T_1}{T_0 + T_1},\]

where:

\[K_p, K_i, K_d, T_1\]—are parameters of the continuous-time PID controller.

As it can be seen the coefficients depend on the sampling time \(T_0\), in such a way that they increase when \(T_0\) becomes lower. So with the faster control, more data words are needed to code the coefficients. The application of too short words for controller coefficients can lead to unstability. It is known, that the discrete controller is stable when its poles are inside the unit circle on the complex plane. The increase of sampling frequency moves the poles close to the border of the region of stability. The rounding of the coefficients can then lead to a situation when the poles will be shifted outside the unit circle, and in consequence convert the model to an unstable one.

4. AN ALTERNATIVE DISCRETE OPERATORS

One of the methods of increasing the numerical robustness of discrete models is to use discrete operators different to the shift operator \(q\). That is why the concept of \(\delta\) operator was introduced [Middleton and Goodwin, 1990; Gevers and Li, 1993; Swider, 2003], which corresponds better to the differentiation operator \(\frac{d}{dt}\) than the shift operator. In the work Middleton and Goodwin (1990) a comparison of \(q\) and \(\delta\) operators and an analysis of their properties was given. It was shown that models using \(\delta\) operator exhibit a lower sensitivity of poles to numerical errors than the corresponding models using the \(q\) operator. In Fang et al. (2005) the realisation and comparison of discrete controllers designed with \(q\) and \(\delta\) operators for a magnetic bearing control system were described.

The operator \(\delta\) can be described as a special case of generalised ADTO (Alternative Discrete-Time Operator) of the first order [Back et al., 1999]. Alternative operators are used for the improvement of the numerical properties of the model and the robustification of parameters for FWL effects. Generalised ADTO proposed in Back et al. (1999) is given by formula (4).

\[
\nu = \frac{q - c_1}{c_2}
\]  

(4)

where:

\[c_1\] and \(c_2\)—are constant parameters.

The operator \(\delta\) is defined with shift operator \(q\), as per equalities (5). When considering the \(\delta\) operator as a special case of generalised ADTO (4) it can be seen that \(c_1 = 1\), \(c_2 = T_0\).

\[
\delta = \frac{q - 1}{T_0}, \quad q = 1 + \delta T_0
\]  

(5)

The \(\delta\) operator is a discrete equivalent of the differentiation operator. The fundamental merit is a closer relation to continuous systems and better numerical properties of the resulting equations (Gevers and Li, 1993; Swider, 2003). The inverse operator \(\delta^{-1}\) represents numerical integration via the polygon rule. With the \(\delta\) operator there is an associated discrete transform \(\Gamma\) of the complex variable \(\gamma\) (6) with a discrete transfer function \(G_\gamma(\gamma)\).

\[
\gamma = \frac{z - 1}{T_0}, \quad G_\gamma(\gamma) = \frac{T_0\gamma}{1 + T_0\gamma} L^{-1}\left\{\frac{G_s(s)}{s}\right\}
\]  

(6)

A very important property of the discrete transfer function \(G_\gamma(\gamma)\) achieved by the direct replacement of the complex variable \(s\) via the complex variable \(\gamma\), is its convergence to the continuous transfer function \(G_s\) with a sampling time \(T_0\) tending towards zero (8).

\[
\lim_{T_0 \to 0} G_\gamma(\gamma) = G_s(s)
\]  

(8)

It means, that for very short control periods we can replace the continuous transfer function \(G_s\) with a discrete one \(G_\gamma\), through a simple replacement of \(s\) by \(\gamma\). This procedure is called emulation (Swider, 2003). This situation takes place when fast controllers based for example on FPGA circuits are used. In these controllers, very short control times can be achieved, which in some cases can result in fulfilling (8) even for FCS (Fast Control System) type systems [Piątek and GREGA, 2009].

In the case of linear systems, the description of the control algorithm in the form of the transfer function is the equivalent to the infinite number of descriptions in state space form. State space representations are equivalent, on the condition of the infinite precision of computations and no quantisation of parameters. In practical cases the realisations differ, because of FWL effects. One can attempt to find such a realisation that will minimise the rounding errors and the quantisation effects on the quality. An extensive discussion of the minimisation of FWL effects in fixed point architecture realisations of linear systems was described in Gevers and Li (1993) and for floating point architectures in Swider (2003), Swider (2000b) and Swider (2000a). These works also describe methods of
finding realizations, minimising the influence of errors for systems modelled with the use of $\delta$ operator.

5. EXAMPLE

Simulations and experiments of laboratory magnetic suspension (Piątek, 2002; Baranowski and Piątek, 2008) control will be presented as an illustration of the applications and merits of the $\delta$ operator. Figure 5 illustrates a block diagram of the control system applied during the experiments.

![Diagram of the control system](image)

**Fig. 5.** Magnetic Levitation control system

For the control of the magnetic levitation system, the PID controller was selected. The discrete controllers used during the experiments were created through the discretisation of the continuous controller via different approximation methods (Piątek, 2007). Controller algorithms obtained in this way were implemented with canonical observer realisation (Kaczorek, 1990) as a code in VHDL (Very High Speed Integrated Circuits Hardware Description Language). For every implemented controller the procedure of selecting the data word length was performed. For controllers obtained using the complex variable $z$ two types of words were used for parameters (PWL — Parameter Word Length) and for signals (SWL — Signal Word Length). In the case of complex variable $\delta$ there was applied an additional word length for writing the discretisation period $T_{\delta}$ in the $\delta^{-1}$ integrators (TWL — Time Word Length). The procedure of word length selection was based on the series of simulation experiments for the model of the closed loop system with discrete controller. In one set of experiments only one discretisation method was used. The goal was to determine the minimal length of PWL, SWL and TWL for which the control of MagLev is possible with the appropriate quality. By "appropriate quality" of control we understand a situation when changing the word length does not lead to an improvement in quality. The simulation model of the closed loop system included the resolutions of A/D and D/A converters, the structure and arithmetic of the controller. However, the accuracy of the converters and internal and exogenous disturbances were not modelled. These effects are present in the real control system, so they influence the required word length. Additionally, the simulation (non-linear) model does not precisely imitate the real machine. That is why the simulation results obtained were verified by experiments on the real plant. During the experiments, SWL obtained from simulations were corrected. PWL did not require modifications, because the effects of quantisation were fully modelled in simulations. Experiments were performed with the application of the control system in Fig. 5. The experimental choice of SWL was similar to the simulation one.

The most interesting results of the experiments for the chosen controllers and sampling periods are included in Tab. 1 and Tab. 2. The tables also include information regarding the usage of FPGA resources for different controllers. This information is given as a percentage of occupied Slices (Slice—the most fundamental element of the FPGA circuit). This parameter is available from the reports generated by the software used for the design of FPGA logic. However, it cannot be used as a justification of the actual resource needs of the tested controller.

<table>
<thead>
<tr>
<th>Sample Time</th>
<th>Approximation</th>
<th>Approximation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_0 = 700 \mu s$</td>
<td>PWL = 15</td>
<td>PWL = 16</td>
</tr>
<tr>
<td>SWL = 18</td>
<td>SWL = 22</td>
<td></td>
</tr>
<tr>
<td>TWL = 47%</td>
<td>TWL = 50%</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sample Time</th>
<th>Approximation</th>
<th>Approximation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_0 = 40 \mu s$</td>
<td>PWL = 25</td>
<td>PWL = 26</td>
</tr>
<tr>
<td>SWL = 38</td>
<td>SWL = 40</td>
<td></td>
</tr>
<tr>
<td>TWL = 52%</td>
<td>TWL = 56%</td>
<td></td>
</tr>
</tbody>
</table>

6. CONCLUSION

In the real-time computer implementation of control algorithms, control task period (sampling, actuation), the computational efficiency of the applied controller platform, and the effects of the finite-precision implementation can not be considered separately.

In many cases, increasing the sampling/actuating frequency in the control-loop is justified or even needed. The important benefits of such an approach can be:

- improvement of control quality,
- greater robustness to disturbances,
- increasing the range of the proper operation of linear controllers controlling nonlinear systems.

Increasing the sampling frequency can however numerically destabilise the controller, which can lead to the