Discrete-Time Rule-based Time Optimal Controller

Toma L. Dragomir*, Octavian Ștefan*, Sorin Nanu*, Alexandru Codrean**

** Politehnica University Timisoara, 2 Piata Victoriei, Timisoara, 300006, Romania (e-mail: toma.dragomir@upt.ro, octavian.stefan@upt.ro, sorin.nanu@upt.ro)*
*** Technical University of Cluj-Napoca, Cluj-Napoca, 400114, Romania, (e-mail: alexandru.codrean@aut.utcluj.ro)*

Abstract: The article presents a discrete rule-based optimal control (D-RBOC) algorithm, as an alternative to known solutions for discrete time optimal control of a double integrator plant, which is the simplest positioning process**.** The algorithm was designed for a double integrator system, with the step response invariant discretization, using rules developed for the movement in the state space. The comparison of the proposed solution with fhan solutions, which are widely spread in the literature, proves the correctness of the former. The article shows that the developed D-RBOC behavior is very close to other optimal regulators. Because in the real physical system implementation, the control loop is affected by time delay due to processing, conversions, transport, etc. in the second part of article, the D-RBOC algorithm is adapted for this situation. The adaptation is based on time delay compensation and can also be used for other optimal control algorithms. The experiments were performed with two dSPACE systems.

Keywords: Time-optimal control, rule-based controller, delay compensation, positioning systems, discrete-time systems.

1. INTRODUCTION

The optimal transition problem of a system from one state to another, or in particular, the time-optimal control problem, associated to servo-systems, is addressed in the specialized literature for a long time. Textbooks on continuous time optimal control from the 70s and 80s introduce, as case study, the time-optimal control of a double integrator plant considering the limitation of the control signal amplitude (Athans and Falb, 2007; Sage and White, 1977; Weinrich, 1973).

According to (Gao, 2004), in 1999 a solution appeared for the discrete time time-optimal control of a double-integrator plant considering the limitation of the control signal amplitude. The solution, based on isochronic regions, is designed in (Gao, 2004), and its applications in servo control systems is carried on in (Gao and Hu, 2004). The aforementioned articles are comparing the advantages of discrete time versus continuous time control considering the physical realizability and cvasicontinuity of the control signal and the robustness of the system with respect to measurements and various other types of noises. The main aspects of the discrete time servo-systems' capability of following a reference signal are also analyzed.

The discrete time optimal control algorithm from (Gao, 2004) is known in literature as *fhan* (.,.,.,.) because of (Han, 2009), that gathered a large number of citations.

Fhan algorithm's starting point is the use of a discrete time model for the double integrator plant. This is obtained from the continuous time model, using for the derivatives the Newton's difference quotient approximations.

In 2010, in the context of the proposal of „A Complementary form of the Discrete Tracking-Differentiator" in (Li, 2010), and the presentation of an „Optimal control synthesis function of a discrete-time system" in (Sun, 2010), a discrete time model is used for the double integrator plant that considers the control signal as a staircase function with steps at the sampling moments. This type of signal accurately replicates the shape of the control signal in digital control. The derivation of discrete time models for this type of input signals, known as invariant realization to step signal of systems, is considered in numerous books like (Ackerman, 1983) or (Astrom and Wittenmark, 1997).

The result from (Sun, 2010) was generalized in 2015 in (Peng, 2015) using the same discreet time model, for a double integrator plant.

Recently, in (Dragomir and Cîmpeanu, 2018) is emphasized that the solution from (Gao, 2004) is suboptimal due to the discretization procedure for the double integrator plant (forward finite difference). By applying the fhan control signal to a double integrator plant obtained with the invariant realization to step signal method, it can be seen that the system has permanent oscillations in the vicinity of the steady state position. In order to avoid the oscillations, the authors corrected the fhan algorithm, using isochronic regions, in the proximity of the steady state position, for the last two steps.

By analyzing the time-optimal control laws as they appear in (Athans and Falb, 2007; Sage and White, 1977; Weinrich, 1973; Gao, 2004), it can be seen that they use distinct calculus formulas of control signal u, dependent on x_1Ox_2 state space. Let an arbitrary region ρ_i, $[x_1, x_2]^T$ be the current state of the process in this region and $f_i(x_1,x_2)$ the compute formula of the control signal in the region. In this case, the optimal controller manages the different formulas in the x_1Ox_2 plane through an ensemble of rules of the type:

$$\text{IF } [x_1, x_2]^T \in \rho_i \text{ THEN } u = \dots f_i(x_1,x_2) \tag{1}$$

In this context, it can be stated that time optimal controllers of the double integrator plant are „rule-based optimal controllers" (RBOC). They are classified as „continuous-time rule-based optimal controller" (C-RBOC) and „discrete-time rule-based optimal controller" (D-RBOC). In this paper a D-RBOC algorithm is proposed.

This paper proposes a time-optimal control algorithm for the double-integrator plant, modeled using the invariant realization to step signal method and proposes a solution for time delay compensation, delay that appears in physical implementations.

Next, the structure of this paper is: the introduction of the models used for the double integrator plant (Chapter 2), the derivation of the formulas for the optimal control law (Chapter 3), the synthesis of the new D-RBOC (Chapter 4), the comparison between the effects of the new D-RBOC and the ones from previous studies (Chapter 5), the new D-RBOC algorithm extended for time delay systems (Chapter 6) and, finally, some conclusions (Chapter 7).

2. DOUBLE INTEGRATOR PLANT

2.1 Continuous time model

The state space equations of the double integrator plant are:

$$\begin{cases} \dot{x}_1(t) = x_2(t), & x_1(0) \quad \textit{fixed} \\ \dot{x}_2(t) = u(t), & x_2(0) \quad \textit{fixed} \end{cases} \tag{2}$$

In (2) $t \in R_+$ is the time. Usually, the variables x_1 and x_2 mean position and speed respectively, while the input u is associated to acceleration. $x_1(0)$ and $x_2(0)$ are the initial conditions of the plant.

Considering $u(t)=u_o$=const., the state trajectories $x_2(x_1)$ are either descending parabolas (when $u_o<0$), straight lines (when $u_o=0$) or ascending parabolas (when $u_o>0$), like in Fig. 1.

Parabolas 1, …, 2′, 4, …, 5′ have the equations (3), and straight lines 3 and 3′ have the equations (4).

$$x_1(t) = \frac{1}{2 \cdot u_o} \cdot x_2^2(t) - \frac{1}{2 \cdot u_o} \cdot x_2^2(0) + x_1(0) \tag{3}$$

$$x_2(t) = x_2(0), \quad (x_1(t) \in R) \tag{4}$$

Considering the state trajectories in Fig.1, it is important to emphasize the following:

i) If $|u_o|$ has small values, the parabolas have a wider opening and if $|u_o|$ has large values, the parabolas have more pointed shapes (curves 1 and 1′ versus curves 2 and 2′ etc.).

ii) For a fixed u_o we obtain a family of parabolas that, depending on $x_1(0)$ and $x_2(0)$, can be obtained one from another by translating along Ox_1 axis (pairs 1 - 1′, 2 - 2′, 4 - 4′, 5 - 5′ represent this kind of families). Any parabolas family fully covers the state space.

iii) For a family of parabolas corresponding to a fixed value of $|u_o|$, the movement on all the segments of the trajectories that have projections of equal length on the Ox_2 axis, uses the same amount of time. In Fig. 1, the movement on both arcs MN and PQ uses the same amount of time, as the trajectories 1 and 1′ belong to the same family and their projection on Ox_2 axis is the segment AB.

iv) For two parabolas' families corresponding to two fixed values of u_o, $|u'_o| < |u''_o|$, the movement on the segments of the trajectories that have the same projection length on Ox_2 axis, is faster on trajectories of parameter u''_o than on trajectories of parameter u'_o. E.g. in Fig. 1 arc MN is covered faster than arc RS.

v) The movement direction along trajectories 3 and 3′ is established by the sign of $x_2(0)$.

vi) If d and d′ are two straight lines that pass through the origin of the x_1Ox_2 plane, then the movement on any trajectory 3 and 3′ between d and d′ is performed in the same amount of time.

Further on, the state trajectories (3) that pass through the origin of the x_1Ox_2 plane (when $u=r$, r being the maximum value of u) are called *straight paths*. The evolution of the current point on the straight paths is described by (5). Consequently, the initial state on straight paths must fulfill the condition (6):

$$x_1(t) = \frac{1}{2 \cdot r} \cdot x_2^2(t) \tag{5}$$

$$x_1(0) = \frac{1}{2 \cdot r} \cdot x_2^2(0) \tag{6}$$

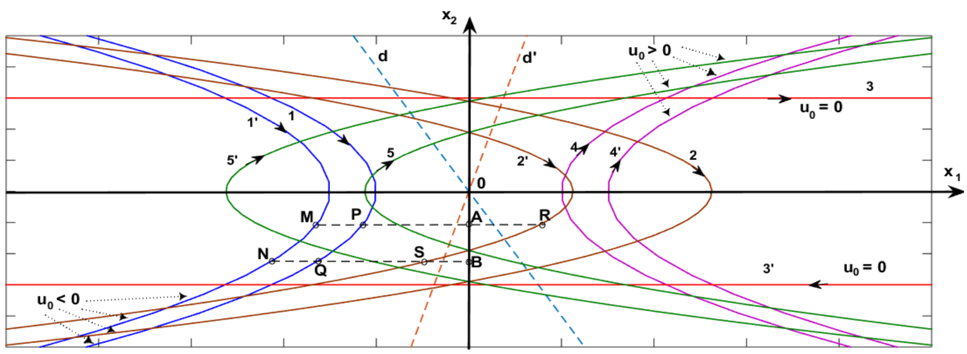


Fig. 1. Simple state trajectories of system (2) when $u(t)=u_o$: {1, 1′, 2, 2′} for $u_o<0$, {3, 3′} for $u_o=0$, and {4, 4′ 5, 5′} for $u_o>0$.

2.2 Discrete time model

For the digital control of the double integrator plant (2), the control signal $u(t)$ is a staircase function like (7), generated by the digital to analog converter by processing the control sequence (8) sent at moments $t_k=k\cdot h$, $k\in N$. The parameter $h>0$ is the sampling step generated by the control device.

$$u(t) = u_{k*} = const., \quad t \in [k\cdot h,\ (k+1)\cdot h), \quad k \in N \tag{7}$$

$$\{u_{k*}\}_{k\in N} = \{u_{0*}, u_{1*}, u_{2*}, \Lambda\} \tag{8}$$

The behavior of the double integrator plant at sampling time t_k is described by equations (9):

$$\begin{cases} x_1((k+1)\cdot h) = x_1(k\cdot h) + h\cdot x_2(k\cdot h) + 0.5\cdot h^2\cdot u(k\cdot h), \\ x_2((k+1)\cdot h) = x_2(k\cdot h) + h\cdot u(k\cdot h)\ . \end{cases} \tag{9}$$

The system (9) represents the invariant realization to step signal of system (2). As $u(t)$ is constant within the time interval $t\in[k\cdot h,(k+1)\cdot h)$ and is equal to u_{k*}, the points $(x_1(k\cdot h), x_2(k\cdot h))$ and $(x_1((k+1)\cdot h), x_2((k+1)\cdot h)$ will be found always on the boundary of a segment of a trajectory like in Fig. 1, corresponding to the control u_k. Consequently, the statements i) – vi) are valid mutatis mutandis also for system (9).

For a simplified presentation, the equations (9) will be rewritten using for state and input signals, scaled dimensionless variables (10) obtained by dividing variables (9) with the base values (11), where r is, according to (12) the absolute maximum value of $u(t)$:

$$u_k = \frac{u(k\cdot h)}{u_b}, \quad x_{1,k} = \frac{x_1(k\cdot h)}{x_{1b}}, \quad x_{2,k} = \frac{x_2(k\cdot h)}{x_{2b}}, \tag{10}$$

$$u_b = r, \quad x_{1b} = 0.5\cdot r\cdot h^2, \quad x_{2b} = r\cdot h \tag{11}$$

$$u \in [-r,\quad r]\ . \tag{12}$$

The system (13) with the restriction (14) is obtained:

$$\begin{cases} x_{1,k+1} = x_{1,k} + 2\cdot x_{2,k} + u_k, \quad x_{1,0} \quad \text{fixed} \\ x_{2,k+1} = x_{2,k} + u_k, \quad x_{2,0} \quad \text{fixed} \end{cases}, \tag{13}$$

$$u_k \in [-1,\quad 1]\ . \tag{14}$$

If at the input of the system (13) the constant control sequence $\{u_k\}=\{u_0=u_o,\ u_1=u_o,\ u_2=u_o,\ \ldots,\ u_{n-1}=u_o\}$ is applied, the states will move through points of the trajectory (15), homologous of trajectory (3), and eventually will reach the state $[x_{1,n}, x_{2,n}]^T$ given by (16):

$$x_1 = \frac{1}{u_0}\cdot x_2^2 - \frac{1}{u_0}\cdot x_{2,0}^2 + x_{1,0} \tag{15}$$

$$\begin{cases} x_{1,n} = x_{1,0} + 2\cdot n\cdot x_{2,0} + n^2\cdot u_0, \\ x_{2,n} = x_{2,0} + n\cdot u_0 \end{cases} \tag{16}$$

3. THE SYNTHESIS OF D-RBOC FORMULAS

The objective of time-optimal control is to generate the control sequence $\{u_k\}_{k\in N}=\{u_0,\ u_1,\ u_2,\ \ldots\}$ under restriction (14), so as the process (13) reaches the steady state $[0,\ 0]^T$ from any initial state $[x_{1,0},\ x_{2,0}]^T$, in a minimum number of steps. This means that the number of values in the control sequence has to be minimal.

Using scaled values, the equation of a straight path is (17). In Fig. 2 are represented with black lines the straight paths (of parameter $u_k=\pm1$ or $u(t)=\pm r$), and with red line other trajectories that pass through the origin.

$$x_{1,k} = \pm x_{2,k}^2 \tag{17}$$

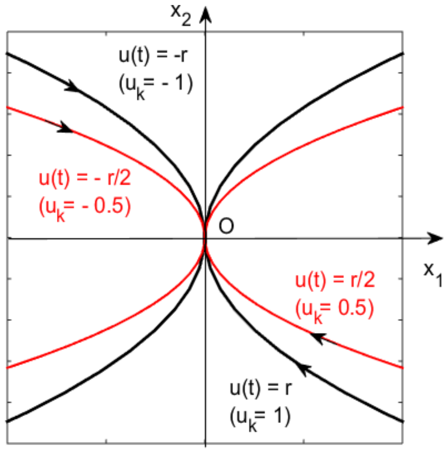


Fig. 2. State trajectories of system (1) passing through the origin.

Let $P(x_{1,k},x_{2,k})$ be the notation for the point into the state space where the system is currently found. From this point on, the system evolves under the control signal u_k. In order to establish the proper value of u_k that has to be applied at the input of the plant (13) to reach the steady state in minimum time, four curved regions are marked within the plane x_1Ox_2 (Fig. 3), relative to the straight paths (17), and the Ox_1 axis. Two of them are represented striped and two are clear. The striped regions include the boundaries as well. The steady state O(0,0) does not belong to any region.

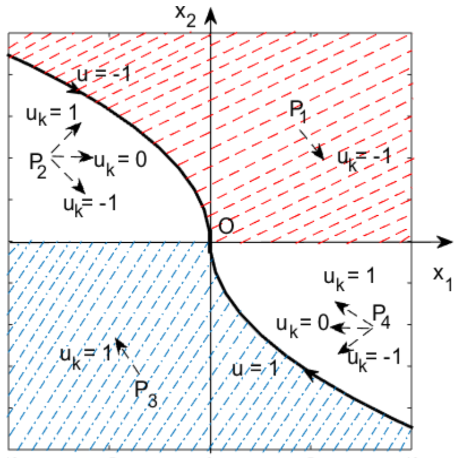


Fig. 3. Curved regions, generic points and the evolution tendencies of states for every region.

The generic positions of the point P in these four regions are noted as P_1, ..., P_4. The types of control signals that can be taken into consideration for a minimal time control are indicated in each region from Fig. 3, based on the assumptions from section 2. Thus:

• For P_1 and P_3 points, the approach towards steady state will be done with maximum acceleration over the simple trajectories of type 1 and 4, by applying $u_k = -1$, respectively $u_k = 1$ (Fig. 1). This conclusion leads to rule R3, respectively rule R4 from the algorithm (25).

• For P_2 and P_4 points, the approach towards steady state will be done through the points over the trajectories composed of arcs. The arcs are obtained by alternating in different time intervals, the control signal u_k in sequence ..., -1, u^*, 1, ... around the point P_4, respectively in sequence ..., 1, u^*, -1, ... around the point P_2. The value u^* represents a control that abides the restriction (14).

In order to establish the control strategy for the point P_4, the representation in Fig. 4 will be used. The figure shows comparatively 3 particular evolutions in two steps of the system (2), provoked by a control sequence $\{u_0, u_1\}$ applied when the system is placed in the point P_4. If the control sequence $\{u_0, u_1\}=\{1, -1\}$ is applied, the system eventually gets to the state Q_1. If the control sequence $\{u_0, u_1\}=\{0, 0\}$ is applied, the system eventually gets to the state Q_2, and if the control sequence $\{u_0, u_1\}=\{-1, 1\}$ is applied, the system eventually gets to the state Q_3. The points P_4, Q_1, Q_2 and Q_3 are collinear. For a minimal time control strategy, the concern is that the points Q_i , i = 1, 2 or 3, to be as close as possible to the straight path (16), in our case, the trajectory $x_{1,k}=x_{2,k}^2$, respectively the point Q, where the line P_4Q_3 crosses the straight path. As it can be seen, the most favorable case corresponds to control sequence $\{u_0, u_1\}=\{-1, 1\}$.

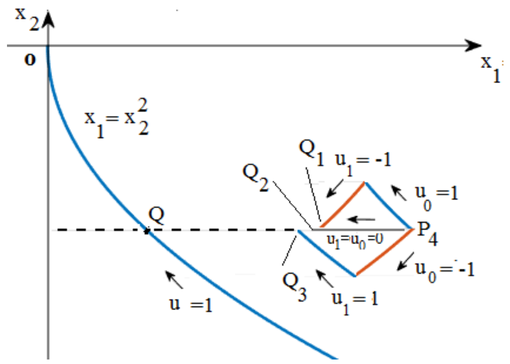


Fig. 4. Alternative state trajectories for the system (13) evolution being initially in state P_3.

This can be generalized as follows: when the control sequence (18), formed of j values of -1 and j values of 1, is applied to the system (13) found in the point P_4, the system gets on a point Q_m belonging to the segment P_4Q (Fig. 5).

$$\{u_0, \ldots, u_{j-1}, u_j, \ldots, u_{2j-1}\} = \{-1, \ldots, -1, 1, \ldots, 1\} \tag{18}$$

Proceeding with the previous reasoning, it can be concluded that in order to obtain a time-optimal control, the value of j must be selected such as the point Q_m to be placed on the closed segment P_4Q closest to Q, and then, from Q_m, the system to move on the straight path OQ towards the steady state. Considering both statement iii) from section 2a and the fact that the control sequences equal to 1 determine movement on trajectories SQ_m and OQ, it follows that the transition from S on the straight path OQ is more advantageous than movement on the trajectory SQ_m followed by a transition from Q_m to the straight path. The explanation is that the movement on the same distance on Ox_1 is faster with greater speed x_2 and the reference to the start point P_4 is no longer necessary.

Consequently, in the case of point P_4, the solution of the minimal time control problem can be reduced to the identification of the moment when the system state can be considered to have the characteristic of the point S, i.e. the moment when with a control $u_j \in [-1, 1]$ the system can reach the straight path (17). The situation is exemplified in Fig. 6. The segment R′R″ contains all the points (states) the system can reach from the point S with $u_j \in [-1, 1]$. In order to continue the move in minimal time, a control signal must be applied to get the system in point R.

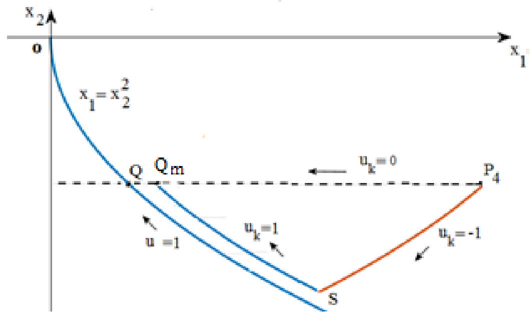


Fig. 5. State trajectory P_4SQ_m when control (18) is applied.

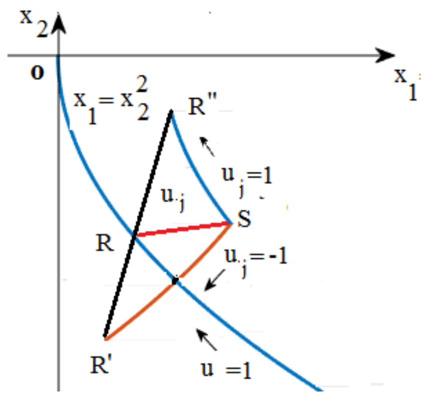


Fig. 6. Possible evolutions from point S when control $u_j \in [-1, 1]$ *is applied.*

Let $[x_{1,j}, x_{2,j}]^T$ be the state corresponding to the point S. The position of the point S regarding the straight path (17) has the property

$$x_{1,j} - x_{2,j}^2 > 0\,. \tag{19}$$

The system transition from S to a point on the segment R′R″ is described by the equations:

$$\begin{cases} x_{1,j+1} = x_{1,j} + 2 \cdot x_{2,j} + u_j, & u_j \in [-1, 1] \\ x_{2,j+1} = x_{2,j} + u_j, \end{cases} \tag{20}$$

In order to determine the values of u_j that provide the transition, it is taken into consideration that in positions R′, R and R″ one has respectively:

$$x_{1,j+1} - x_{2,j+1}^2 < 0\,, \tag{21.1}$$

$$x_{1,j+1} - x_{2,j+1}^2 = 0\,, \tag{21.2}$$

$$x_{1,j+1} - x_{2,j+1}^2 > 0\,. \tag{21.3}$$

The value of the control u_j is obtained from (21.2) after replacing $x_{1,j+1}$ and $x_{2,j+1}$ with the equations (20). The second order equation has the solutions:

$$u_j = 0.5 - x_{2,j} \;\mu\; \sqrt{0.25 + x_{1,j} + x_{2,j}}\,.$$

The only acceptable solution (such that $u_j \in [-1,\ \ 1]$) is:

$$u_j = 0.5 - x_{2,j} - \sqrt{0.25 + x_{1,j} + x_{2,j}}\,. \tag{22}$$

Likewise, for the point P_2 in Fig. 3:

$$u_j = -0.5 - x_{2,j} + \sqrt{0.25 - (x_{1,j} + x_{2,j})} \tag{23}$$

The formulas (21) and (22) are integrated in rules R7 and R8 of the algorithm (25).

Note: Let n_S be the number of steps n in which the system (13) reaches from point P_4 to point S. It can be calculated by observing that the arcs P_4S and Q_mS in Fig. 5 are symmetrical to the vertical line that passes through the point S. Considering that the point P_4 has the coordinates $(x_{1,k}, x_{2,k})^T$, the point Q will have the coordinates $((x_{2,k})^2, x_{2,k})^T$, and the point S the coordinates $(0.5\,(x_{2,k})^2, x_{2,nS})^T$. Thus, according to first equation in (16), the following equation is obtained:

$\frac{1}{2}x_{2,k}^2 = x_{1,k} + 2\cdot n_S \cdot x_{2,k} - n_S^2$, with the solution:

$$n_S = \begin{cases} \left\lceil x_{2,k} + \sqrt{\frac{1}{2}x_{2,k}^2 + x_{1,k}} \right\rceil, \text{if } x_{2,k} + \sqrt{\frac{1}{2}x_{2,k}^2 + x_{1,k}} \ge 0 \\ 0, \textit{ if } x_{2,k} + \sqrt{\frac{1}{2}x_{2,k}^2 + x_{1,k}} < 0 \end{cases}$$

Likewise, for the point P_2:

$$n_S = \begin{cases} \left\lceil -x_{2,k} + \sqrt{\frac{3}{2}x_{2,k}^2 - x_{1,k}} \right\rceil, \text{if} -x_{2,k} + \sqrt{\frac{3}{2}x_{2,k}^2 - x_{1,k}} \ge 0 \\ 0, \text{ if } -x_{2,k} + \sqrt{\frac{3}{2}x_{2,k}^2 - x_{1,k}} < 0 \end{cases}$$

The results can be used to approximate the number of steps in which the system reaches the steady state.

4. D-RBOC CONTROL LAW

The D-RBOC control law, which is the subject of this paper, is intended for the real time implementation of the closed-loop control structure from Fig.7. Fig.7.a emphasizes the main elements of the physical structure, while Fig.7.b. contains the algorithmic elements.

Due to measurement and calculus errors, the evolution of the plant (13) on the straight path, has only theoretical value. In reality, the system will move only on nearby trajectories. In this context, in order to reach the steady state, when the system gets near it on nearby trajectories, the first two rules of the algorithm developed in the paper (Dragomir and Cîmpeanu, 2018) will be used. Using the current notations, these two rules are:

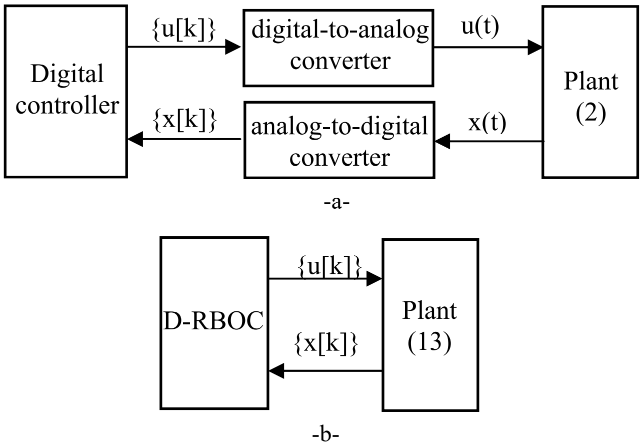


Fig. 7. The structure of the minimal time control system

$$\begin{cases} \text{R1: IF } x_{1,k} + x_{2,k} = 0 \vee \left|x_{2,k}\right| \le 1, \text{ THEN } u_k = -x_{2,k} \\ \text{R2: IF } \left|x_{1,k} + x_{2,k}\right| \le 1 \vee \left|\frac{1}{3}x_{1,k} + x_{2,k}\right| \le \frac{2}{3}, \\ \qquad\qquad \text{THEN } u_k = -\frac{1}{2}x_{1,k} - \frac{3}{2}x_{2,k} \end{cases} \tag{24}$$

These two rules provide the reach of the steady state in one or two steps, when system state is nearby the origin, into the region obtained from the graphic interpretation of the premise of the two rules.

The new D-RBOC proposed in the present paper, referred as PM (proposed method), adds to that two rules, based on the reasoning from the chapter 4, the following 6 rules:

$$\begin{cases} \text{R3: IF } x_{1,k} + x_{2,k}^2 \ge 0 \ \wedge\ x_{1,k} \ne 0 \ \wedge\ x_{2,k} \ge 0, \\ \qquad\qquad \text{THEN} \quad u_k = -1; \\ \text{R4: IF } x_{1,k} - x_{2,k}^2 \le 0 \ \wedge\ x_{1,k} \ne 0 \ \wedge\ x_{2,k} \le 0, \\ \qquad\qquad \text{THEN} \quad u_k = 1; \\ \text{R5: IF } x_{1,k} - x_{2,k}^2 > 0 \wedge x_{2,k} < 0 \wedge x_{1,k+1} - x_{2,k+1}^2 \ge 0 \\ \qquad\qquad \text{THEN} \quad u_k = -1; \\ \text{R6: IF } x_{1,k} + x_{2,k}^2 < 0 \wedge x_{2,k} > 0 \wedge x_{1,k+1} + x_{2,k+1}^2 \le 0, \\ \qquad\qquad \text{THEN} \quad u_k = 1; \\ \text{R7: IF } x_{1,k} - x_{2,k}^2 > 0 \ \wedge\ x_{2,k} < 0 \ \wedge\ x_{1,k+1} - x_{2,k+1}^2 < 0, \\ \qquad \text{THEN} \quad u_k = 0.5 - x_{2,k} - \sqrt{\frac{1}{4} + x_{1.k} + x_{2,k}}\,; \\ \text{R8: IF } x_{1,k} + x_{2,k}^2 < 0 \ \wedge\ x_{2,k} > 0 \ \wedge\ x_{1,k+1} + x_{2,k+1}^2 > 0, \\ \qquad \text{THEN} \quad u_k = -0.5 - x_{2,k} + \sqrt{\frac{1}{4} - (x_{1.k} + x_{2,k})}\,. \end{cases} \tag{25}$$

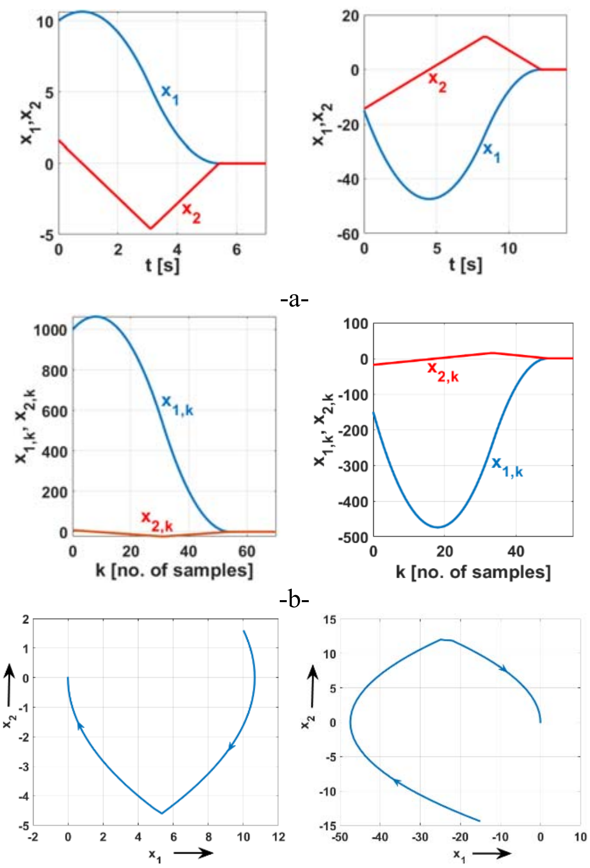
The rule R3 is applied for the region in Fig. 3 where P_1 is placed, while rule R4 for the region where P_3 is placed. The rule R5 is used for the region where P_4 is placed, if the applied control signal does not take the system in a position of the P_3 type (inside the striped area). Alike, the rule R6 refers to the region where P_2 is placed, if the applied control signal does not take the system in a position of the P_1 type (inside the striped area). The rule R7 corresponds to the situation when the system is in a position of P_4 type and it can move on a straight path (17), and the rule R8 when the state of the system corresponds to a point of P_2 type and it can move on a straight path.

Fig. 8 illustrates, as examples, the results obtained with the proposed algorithm in two numerical applications:

- Case 1 (left images): $x_{1,0} = 1000$, $x_{2,0} = 8$, $h = 0.1$ s, $r = 2$, $t \in [0, 7]$ s; ($x_1(0) = 10$, $x_2(0) = 1.6$).

- Case 2 (right images): $x_{1,0} = -150$, $x_{2,0} = -18$, $h = 0.25$ s, $r = 3.2$, $t \in [0, 14]$ s; ($x_1(0) = -15$, $x_2(0) = -14.4$).

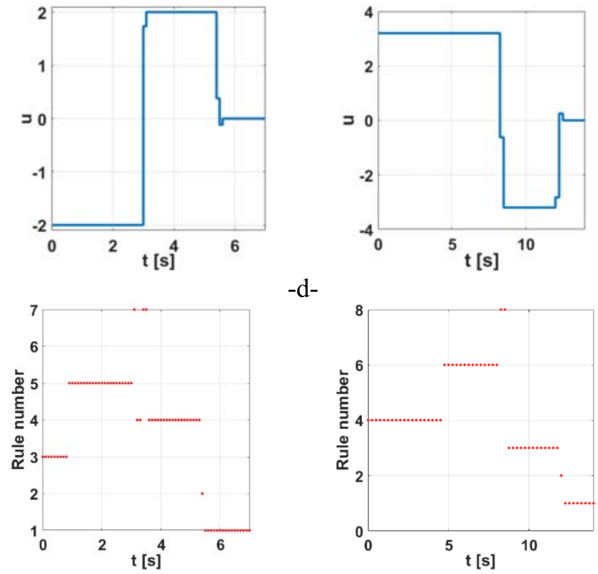
Fig. 8.a contains variations of $x_1(t)$ and $x_2(t)$ in absolute values, while Fig. 8.b in scaled values ($x_{1,k}$, $x_{2,k}$). In Fig. 8.c are presented the state trajectories with x_1 and x_2 in absolute values, in Fig. 8.d the control $u(t)$ in absolute values, and in Fig. 8.e the sequences of rules activated in time. Thereby, in the case 1 the sequence of rules activated is R3, R5, R7, R4, R7, R4, R2, R1, and in the case 2, the sequence is R4, R6, R8, R3, R2, R1.



-a-

-b-

-c-



-d-

-e-

Fig. 8. Case 1 (left), Case 2 (right).

Note: Regarding the interpretation of the curves like the ones from the figures 8, one must ignore the linear interpolation performed by the graphical representation environment, between the sampling points.

5. COMPARISON BETWEEN THE CONTROL METHODS

In this chapter, the PM control is compared with the CM (Dragomir and Cîmpeanu, 2018) control and it is shown that the former performs better in simulations. The PM is also compared with Peng's method (Peng, 2015), and it is shown that even if the PM implies a simpler mathematical framework, the simulation results are very similar.

5.1 Comparison between the effects of the PM control and CM control

In (Dragomir and Cîmpeanu, 2018) a digital time-suboptimal control algorithm is presented, referred further on as „corrected fhan control law". This algorithm noted as CM (corrected method), corrects the control algorithm in (Gao, 2004), called „fhan control law", within the regions where R1 and R2 rules of (24) are activatable. The CM algorithm contains the three rules that in scaled values become:

$$\begin{cases} R'1: \text{IF } x_{1,k} + x_{2,k} = 0 \wedge \left|x_{2,k}\right| \le 1, \text{ THEN } u_k = -x_{2,k} \\ R'2: \text{IF } \left|x_{1,k} + x_{2,k}\right| - 1 \le 0 \wedge \left|\frac{1}{3}x_{1,k} + x_{2,k}\right| - \frac{2}{3} \cdot \le 0, \\ \qquad \text{THEN } u_k = -\frac{1}{2}x_{1,k} - \frac{3}{2}x_{2,k} \\ R'3: \text{IF } \left|x_{1,k} + x_{2,k}\right| - 1 > 0 \vee \left|\frac{1}{3}x_{1,k} + x_{2,k}\right| - \frac{2}{3} \cdot > 0, \\ \qquad \text{THEN } u_k = -\begin{cases} sign(a_k), & \left|a_k\right| - 1 > 0 \\ a_k, & \left|a_k\right| - 1 \le 0 \end{cases}, \end{cases} \tag{26}$$

$$a_k = \begin{cases} x_{2,k} + \left(\sqrt{\frac{1}{4} + |b_k|} - \frac{1}{2}\right) \cdot sign(b_k), \text{for } |b_k| > 2 \\ \frac{1}{2} \cdot x_{1,k} + 2 \cdot x_{2,k}, \text{ for } |b_k| \le 2 \end{cases} \quad (27)$$

$$b_k = x_{1,k} + 2 \cdot x_{2,k}$$

Further on, the behavior of the system using PM and CM controllers is illustrated through numerical examples.

- Case 3 (Fig. 9): $x_{1,0}$=-1, $x_{2,0}$=-1, h=0.25 s, r=3.2, $t \in [0, 1]$ s; ($x_1(0)$ = -0.1, $x_2(0)$ = -0.8).

The results obtained with PM and CM are identical as the transitory regime takes place within the areas of R1 and R2. In the left figure, the variations of states $x_{1,k}$, $k \in \{0, 1, 2, 3, 4\}$ and $x_{2,k}$, $k \in \{0, 1, 2, 3, 4\}$ appear in scaled values, in the right figure, the state trajectories $x_{2,k}$ ($x_{1,k}$) appear also in scaled values, and in the lower figure the control signal $u(k \cdot h)$, $k \in \{0, 1, 2, 3, 4\}$ appears in absolute values.

- Case 4 (Fig. 10): $x_{1,0}$ = -3, $x_{2,0}$ = -2, h = 0.25 s, r = 3.2, $t \in [0,2]$ s; ($x_1(0)$ = -0.3, $x_2(0)$ = -1.6).

In this case there are differences between the two systems. All figures are represented in scaled values. In the central diagram, the differences between the values of x_1, x_2 and u (PM-CM) appear as functions of scaled time (number of steps). They were noted as *delta*-x_1, *delta*-x_2 and *delta*-u.

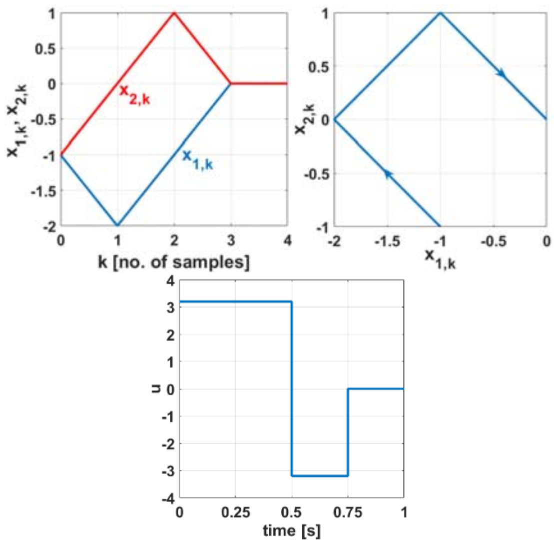


Fig. 9. In case 3 the control laws PM and CM lead to identical results.

- In Fig. 11 case 2 is reloaded: $x_{1.0}$ = -150, $x_{2,0}$ = -18, h = 0.25 s, r = 3.2, $t \in [0, 14]$ s; ($x_1(0)$ = -15, $x_2(0)$ = -14.4).

The left figure presents the variations of the state variables in time, while the right figure the state trajectories. The blue curves correspond to the PM while the red curves correspond to the CM. They indicate that the differences appear after 8 sec. and in the PM case, the system reaches the final state one second faster than in the CM case. Thus, the new control assures a faster evolution. The lower figure illustrates the differences *delta*-x_1=$x_{1,k\text{-}PM}$ - $x_{1,k\text{-}CM}$, *delta*-x_2=$x_{2,k\text{-}PM}$ − $x_{2,k\text{-}CM}$ and *delta*-u=$u_{k\text{-}PM}$ - $u_{k\text{-}CM}$ in scaled values.

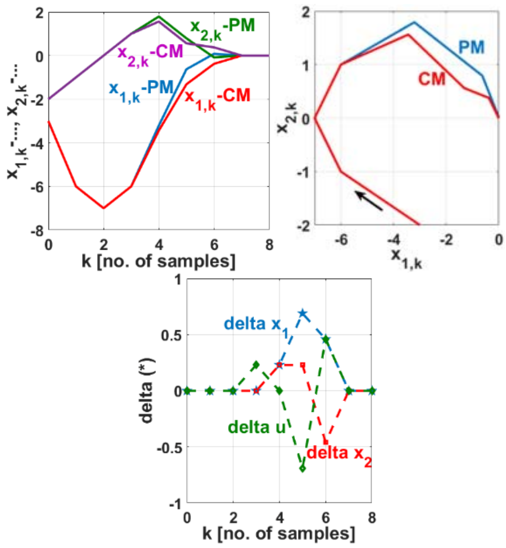


Fig. 10. In case 4 the control laws PM and CM lead to different results.

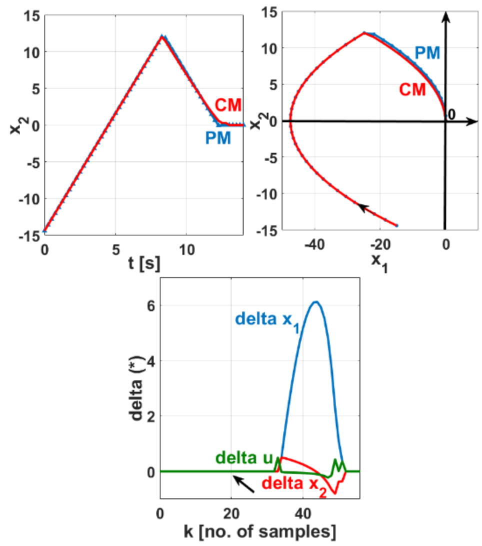


Fig. 11. The comparison of the effect of PM and CM algorithms in case 2 from Fig. 8.

5.2 Comparison between the PM and Peng's method

In scaled values, the control algorithm from (Peng, 2015) takes the three rules form from (28):

$$\begin{cases} \text{R1: IF } u_{c,k} < -1, \text{ THEN } u_k = 1, \\ \text{R2: IF } |u_{c,k}| \le 1, \text{ THEN } u_k = -u_{c,k}, \\ \text{R3: IF } u_{c,k} > 1, \text{ THEN } u_k = -1, \end{cases}$$

where

$$u_{c,k} = x_{2,k} + \frac{1}{1 + fix(\kappa_k)} \cdot x_{c,k} + \frac{1}{2} \cdot fix(\kappa_k) \cdot \text{sgn}(x_{c,k}), \quad (28)$$

$$\kappa_k = \frac{1}{2} \cdot \left(\sqrt{1 + 8 \cdot |x_{c,k}|} - 1\right),$$

$$x_{c,k} = \frac{1}{2} \cdot x_{1,k} + (1 - a) \cdot x_{2,k}$$

In (28), a is a positive parameter ($a \geq 0$) and $fix(x)$ rounds x to the nearest integer in the direction of zero. For $a = 0$, one can obtain the algorithm from (Gao, 2004) and for $a = 0.5$, the algorithm from (Sun, 2010).

For $a = 0.5$, the RBOC algorithm (28) obtains the corresponding form of the double integrator system (9).

Next, the steps through which in (Peng, 2015) the algorithm (28) is obtained for $a = 0.5$ are described as a synthesis. Unlike the form developed in (Peng, 2015), in which it is operated with absolute states and control signals values, below only scaled values are used.

i) The points set from which the origin ($[0, 0]^T$ state) can be reached by the double integrator system in k steps, by applying the control signals $u_i = 1$, $i = 0, \ldots, k\text{-}1$ or $u_i = -1$, $i = 0, \ldots, k\text{-}1$ contains the points $a_{+k}=[k^2, -k]^T$ and $a_{-k}=[-k^2, k]^T$.

ii) The segments set $\overline{a_{\pm j}a_{\pm(j-1)}}$, $j \in N$, (denoted G) defines a broken line that passes through the origin. If the current point $[x_{1,k}, x_{2,k}]^T$ is contained by the segment $\overline{a_{\pm j}a_{\pm(j-1)}}$ and $u = \pm 1$, then the next point, $[x_{1,k+1}, x_{2,k+1}]^T$ is contained by Q, that is the segment $\overline{a_{\pm(j-1)}a_{\pm(j-2)}}$.

iii) If the current point $[x_{1,k}, x_{2,k}]^T$ is found on the segment $\overline{a_{\pm 1}a_0}$, then by applying the control signal $u = -x_{2,k}$, the system state is brought to the origin.

iv) Starting from the fact that according to equations (13) between the current point $[x_{1,k}, x_{2,k}]^T$ and the next point $[x_{1,k+1}, x_{2,k+1}]^T$ there is the following relation $x_{1,k} + x_{2,k} = x_{1,k+1} - x_{2,k+1}$, it results that the double integrator system can transit from the current state on the segment $\overline{a_p a_{p-1}}$ or on the segment $\overline{a_{-p}a_{-(p-1)}}$, by applying the control signal

$$u = -x_{2,k} - \frac{1}{p+1}(x_{1,k} + x_{2,k}) - \frac{p}{2} \cdot sgn(x_{1,k} + x_{2,k})$$

where

$$p = fix\left(\frac{1}{2}\sqrt{1 + 4\left|x_{1,k} + x_{2,k}\right|} - 1\right) .$$

v) Because of the condition $u \in [-1, 1]$, the values of u are adjusted through saturation to the interval [-1, 1]. In the case when $p = 0$, the formula of u is reduced to the one from iii).

The results from above can be obtained by substituting $a = 0$ in (28).

As it was noted in the section 1, the solution from (Peng, 2015), represented by D-RBOC (28) and the solution (25), described above, conduct to very similar results. This is underlined by Fig. 12, referring to case 1 and 2 from Fig. 8. Figures a, b and c show the differences between the values of x_1, x_2 and u obtained using D-RBOC (25) and D-RBOC (28). All the values are given in absolute form. In both cases, the difference *delta-u* takes values of under 1% from r, with the exception of two moments when the difference is approx. 6-8%. Figure d presents the successions in which the three rules R1, R2 and R3 of algorithm (28) are used.

Fig. 13 reproduces the differences *delta-*x_1, *delta-*x_2 and *delta-u*, in the same manner as Fig. 12. The reduced number of sampling steps from the case 3 permits a clearer underlining of the differences that appear when using algorithms (25) and (28).

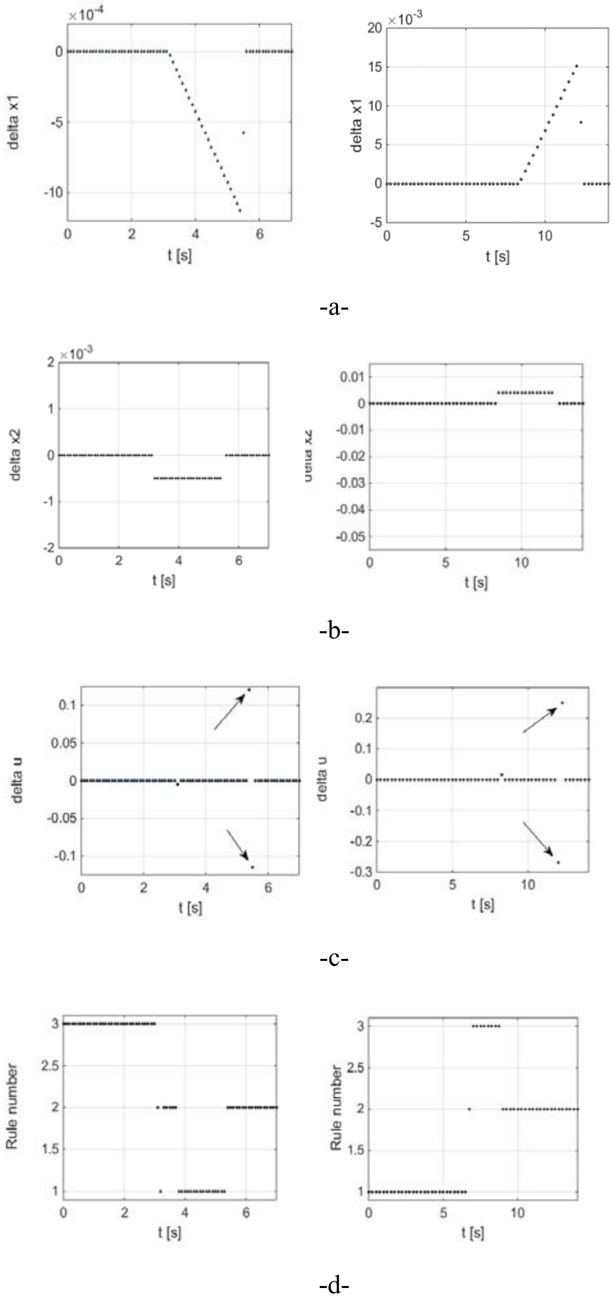


Fig. 12. Comparisons in Case 1 (left) and Case 2 (right) between the D-RBOC (25) and D-RBOC (28) with a=0.5.

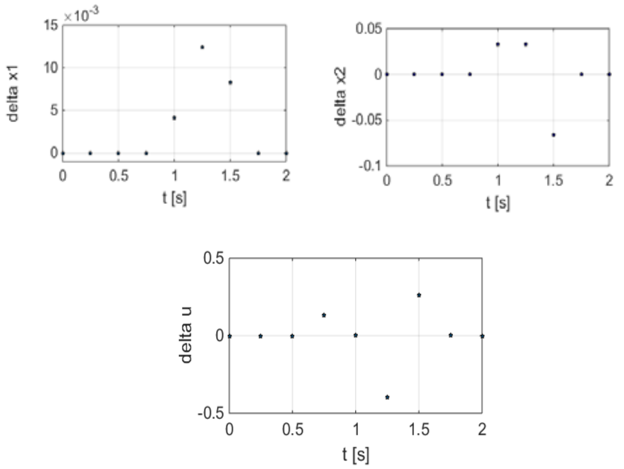


Fig. 13. Comparisons in Case 3 between the D-RBOC (25) and D-RBOC (28) with a=0.5.

6. THE EXTENSION OF D-RBOC ALGORITHMS FOR DOUBLE INTEGRATOR WITH TIME DELAY

The D-RBOC algorithms presented in previous chapters were designed under the assumption of an ideal digital implementation. This means that at the sampling moments three operations are performed consecutively: i) acquire the values of the states x_1 and x_2, ii) calculate the value of the control u_k from these values, iii) apply at the plant input a control signal equal to $u(t)=u_k$. Fig. 14 suggests this operating mode. The vertical lines represent the moments the operations are performed: blue for operation i), red for operation ii) and green for operation iii).

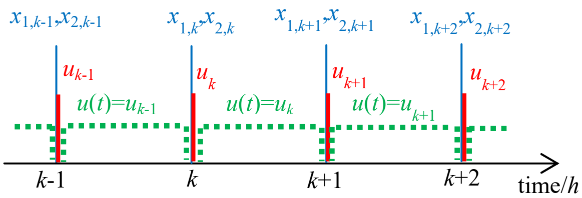


Fig. 14. Control signal generation for an ideal digital system.

Real digital implementation of these algorithms differs from the ideal one. The most important difference is due to the fact that between operation i) and iii) comes up a time delay τ:

$$\tau < h. \tag{29}$$

The operating mode is illustrated in Fig. 15. Unlike the one in Fig. 14, the control value, computed from the values of the state variables $x_{1,k}$ and $x_{2,k}$ taken at $k\cdot h$ moments, is applied at $k\cdot h+\tau$ moments, when x_1 and x_2 have the values $x_{1,k+\tau/h}$, respectively $x_{2,k+\tau/h}$. The quantity k+(τ/h) is the scaled value of $k\cdot h+\tau$ divided to h. The k+(τ/h) moments are marked in the figure with brown lines. Consider the notation:

$$\alpha = \frac{\tau}{h} \tag{30}$$

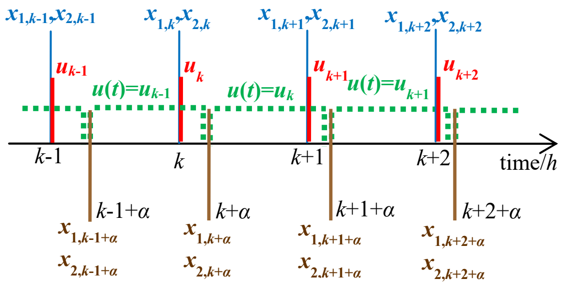


Fig. 15. Approximation model of the control signal generation in case of a real digital system.

Applying the D-RBOC algorithms in the context of a real digital system can lead to the loss of optimality or suboptimality characteristics, the system becoming oscillating or even unstable. Thus, the D-RBOC algorithms have to be extended in order to fulfill the practical implementation conditions.

From the point of view of continuous time systems, the new situation imposes to use, instead of (2), the system:

$$\begin{cases} \dot{x}_1(t) = x_2(t), \quad x_1(0) \quad fixed \\ \dot{x}_2(t) = u(t-\tau), \quad x_2(0) \quad fixed \end{cases} \tag{31}$$

The discrete time model associated to (31) as invariant realization to step signal at $k\cdot h$ moments, based on (Ackerman, 1983) or (Astrom & Wittenmark, 1997) is:

$$\begin{cases} x_1((k+1)\cdot h) = x_1(k\cdot h) + h\cdot x_2(k\cdot h) + \\ \quad +(h\cdot\tau - \dfrac{\tau^2}{2})\cdot x_3(k\cdot h) + \dfrac{1}{2}(h-\tau)^2\cdot u(k\cdot h), \ x_1(0)\ fixed \\ x_2((k+1)\cdot h) = x_2(k\cdot h) + \tau\cdot x_3(k\cdot h) + \\ \quad +(h-\tau)\cdot u(k\cdot h), \quad x_2(0) \ \ fixed \\ x_3((k+1)\cdot h) = u(k\cdot h), \quad x_3(0) \ \ fixed \end{cases} \tag{32}$$

In (32), x_3 is an auxiliary state variable due to the time delay. According to the last equation, for x_3 the same base value as for u will be chosen. The list (10) of base values becomes:

$$u_b = r, \quad x_{1b} = 0.5\cdot r\cdot h^2, \quad x_{2b} = r\cdot h, \quad x_{3b} = r \tag{33}$$

In scaled values, the system (32) becomes:

$$\begin{cases} x_{1,k+1} = x_{1,k} + 2\cdot x_{2,k} + \alpha\cdot(2-\alpha)\cdot x_{3,k} + (1-\alpha)^2\cdot u_k, \\ \qquad\qquad\qquad\qquad x_{1,0} \quad fixed \\ x_{2,k+1} = x_{2,k} + \alpha\cdot x_{3,k} + (1-\alpha)\cdot u_k, \quad x_{2,0} \quad fixed \\ x_{3,k+1} = u_k, \quad x_{3,0} \quad fixed \end{cases} \tag{34}$$

For [$k\cdot h$, $k\cdot h+\tau$] time interval, when $u(t)=u_{k-1}$, as it appears in Fig. 15, computing as in equation (9), from the first two equations (34) one can obtain the equations (35). In scaled values, they become (36).

$$\begin{cases} x_1(kh+\tau) = x_1(kh) + \tau\cdot x_2(kh) + 0.5\cdot\tau^2\cdot u((k-1)h) \\ x_2(kh+\tau) = x_2(k\cdot h) + \tau\cdot u((k-1)\cdot h) \end{cases} \tag{35}$$

$$\begin{cases} x_{1,k+\alpha} = x_{1,k} + 2\cdot\alpha\cdot x_{2,k} + \alpha^2\cdot u_{k-1} \\ x_{2,k+\alpha} = x_{2,k} + \alpha\cdot u_{k-1} \end{cases} \tag{36}$$

Fig. 16 describes the implementation technique of this result. DI_τ is the double integrator with the time delay block that runs after equations (31). C represents any of D-RBOC controllers calculated in the previous sections. It operates with $x_{1,k+\alpha}$ and $x_{2,k+\alpha}$ calculated with formulas (36). C* is the digital compensator, that compensates the effect of the time delay. It contains, apart from the controller C, the blocks that perform the dividing operations ($1/x_{1b}$ and $1/x_{2b}$), the block that performs the reverse of dividing operation (u_b) and the block that model the equations (36). The values calculated with (36) are written in red. The DAC and ADC converters are not shown here, considering that the conversion parameters are equal to 1.

To emphasize the effect of the time delay compensation, the structure in Fig.16 was implemented on two dSPACE modules like in Fig.17. On dSPACE 2 the plant was implemented through equations (2), and on dSPACE 1 D-RBOC (24)-(25) was implemented. All the models work with scaled values. In order to obtain real physical signals between boards as in (10) and (9), both modules have blocks that convert „scaled value → absolute value" and „absolute value → scaled value", according to equations (10), (11) and (33). Lack of synchronization between the internal clocks of the modules, the algorithm computing time and the time needed for conversions introduce an equivalent time delay whose exact value is unknown (Ştefan et al., 2010). It must be measured and approximated. The compensation solution is valid for a constant value of the time delay τ. In case of a time varying delay τ, an adaptive compensation is necessary.

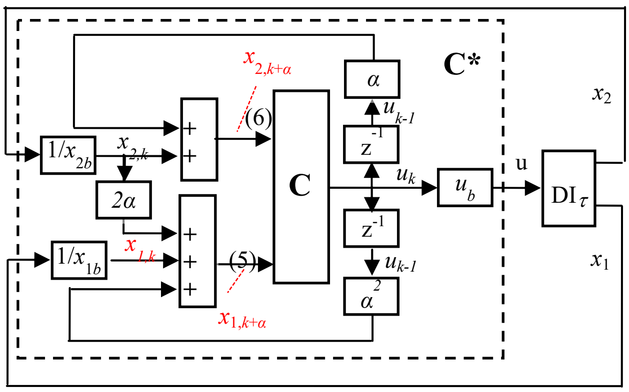


Fig. 16. Block diagram of the implementation of D-RBOC with time delay compensation.

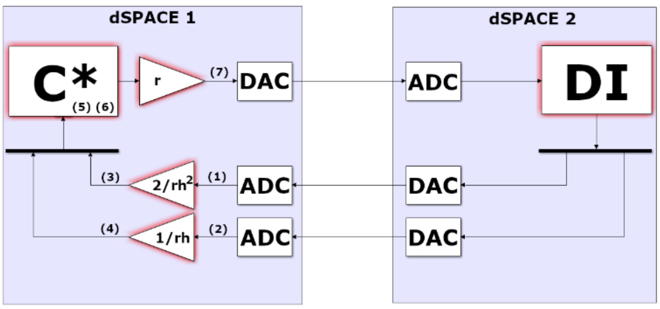


Fig. 17. Block diagram of the implementation structure in Fig. 16 with two dSPACE modules.

In Fig. 17 and Fig. 16 were noted with (1), (2), (3), (4), (5), (6) and (7) the points where the signals were recorded. The experiments were affected by perturbations due to electromagnetic interference.

Fig.18 and Fig.19 present experiments performed with the following setup: $r = 2$ V; $h = 0.5$ s, i.e. $x_{1b}=0.5 \cdot r \cdot h^2 = 0.25$, $x_{2b} = r \cdot h = 1$, $u_b = r = 2$ with initial conditions $x_{1,0} = 15$; $x_{2,0} = -3$. Signals in Fig. 18 shows the situation when the time delay is not compensated (α=0). The effect is oscillations of variables in the system. The control signal switches between the upper and lower limits $r = -2$ V and $r = 2$ V.

As the value of the time delay τ is not known, experiments were made for many values of $\alpha \in [0, 1)$. In Fig. 19, the results of two of these experiments for $\alpha = 0.4$ and $\alpha = 0.9$ are presented. During the process, the time delay compensation is different. For $\alpha = 0.4$ the compensation is weak. The behavior is still a strongly oscillating one, and the control signal switches between limits. However, the oscillation amplitudes of x_1 and x_2 are reduced. For $\alpha = 0.9$, the compensation is more effective. The oscillations are very reduced. The notations x_{1r} and x_{2r} indicate, by index r, that the values of x_1 and x_2 are scaled, measured after a double conversion D/A – A/D within the dSPACE modules.

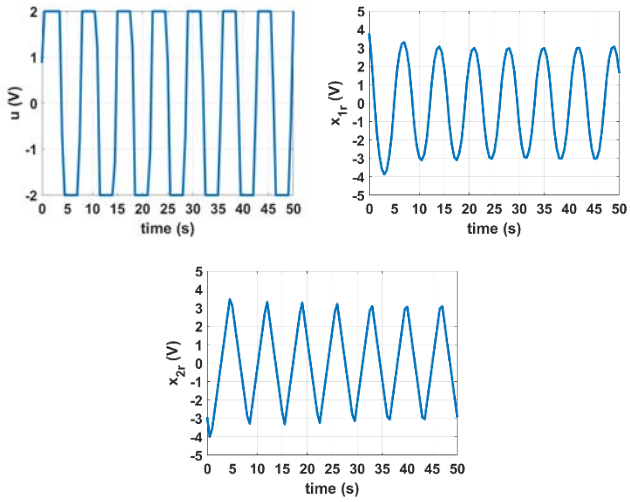
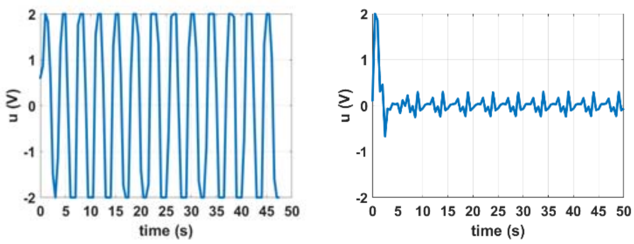


Fig. 18. The behavior of the system in Fig.17 in case α = 0 (time delay is not compensated). Signals u, x_1 and x_2 are taken from points (7), (1), respectively (2).



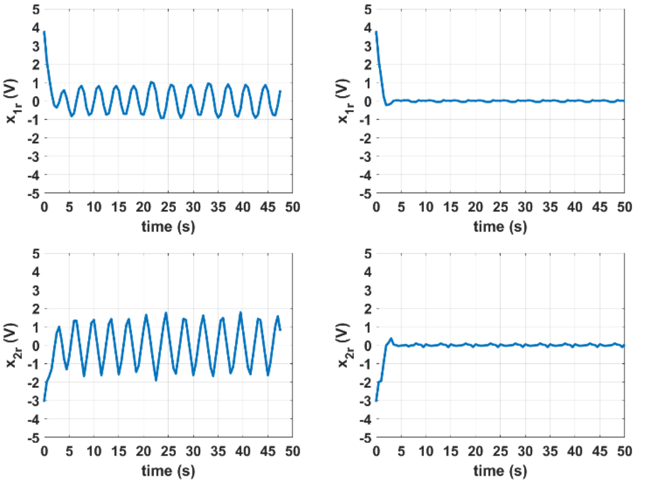


Fig. 19. The behavior of the system in Fig.17 in cases α = 0.4 (left) and α = 0.9 (right). The signals u, x_{1r} and x_{2r} are obtained from the points (7), (1), respectively (2).

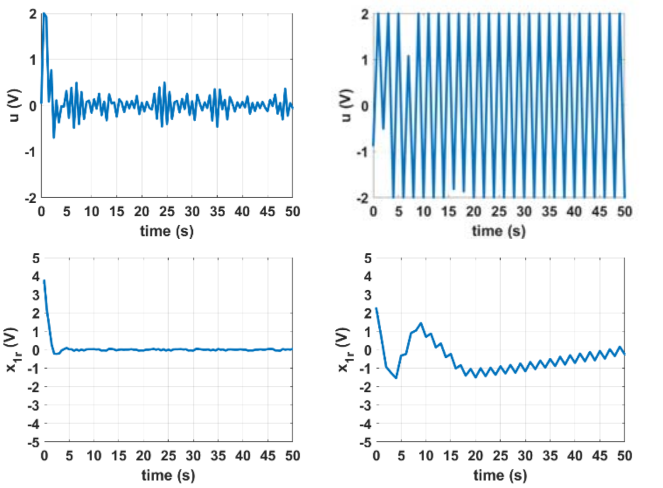
This conclusion encourages the increment of α to 0.95. The results are given in the left column of Fig. 20. Comparing the results for α=0.95 with the results for α=0.9, one can observe a better compensation of the time delay.

Stopping the experiments at this point, it can be concluded that for the implementation from Fig. 17, when h = 0.5 s, the time delay is:

$$\tau = \alpha \cdot h = 0.95 \cdot 0.5 = 0.475\ s. \tag{37}$$

In the right column of the figure, the results are depicted comparatively in case: r = 2 V; h = 1 s, i.e. x_{1b}=$0.5 \cdot r \cdot h^2$ = 1, x_{2b} = $r \cdot h$ = 2, u_b = r = 2, α=0.9 and the initial conditions $x_{1,0}$=15; $x_{2,0}$=-3. Because for the chosen initial conditions the state variable x_1 can determine values for the physical electric signals that exceed the voltage domain of the converters [-10 V, 10V], the moment 0 of these time diagrams was chosen after all the interface signals returned into this interval and no limitations occurred.

On the last two diagrams on every column of Fig. 20, the pairs of signals $\{x_{1,k}, x_{1,k+\alpha}\}$ and $\{x_{2,k}, x_{2,k+\alpha}\}$ are overlapped. They were noted $\{x_1, x_{1c}\}$, respectively $\{x_2, x_{2c}\}$ and were expressed as scaled values. The diagrams highlight the evolution of the time delay compensation.



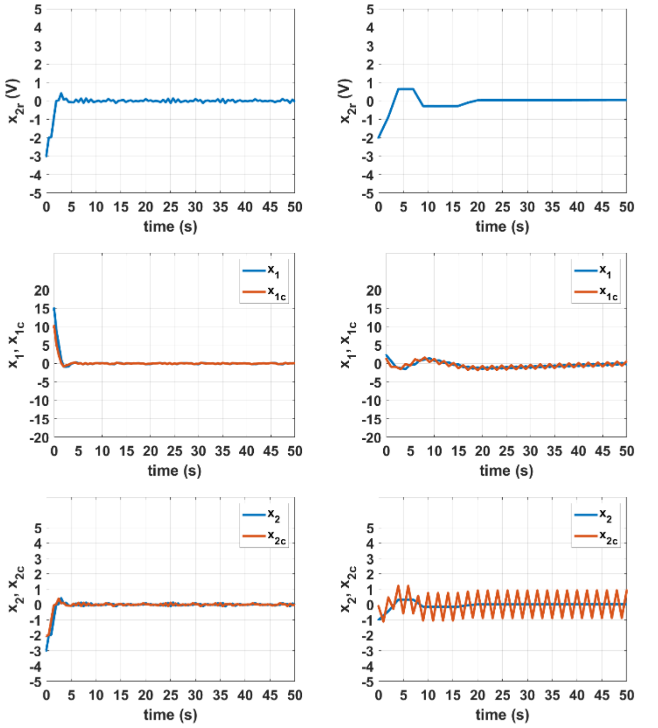
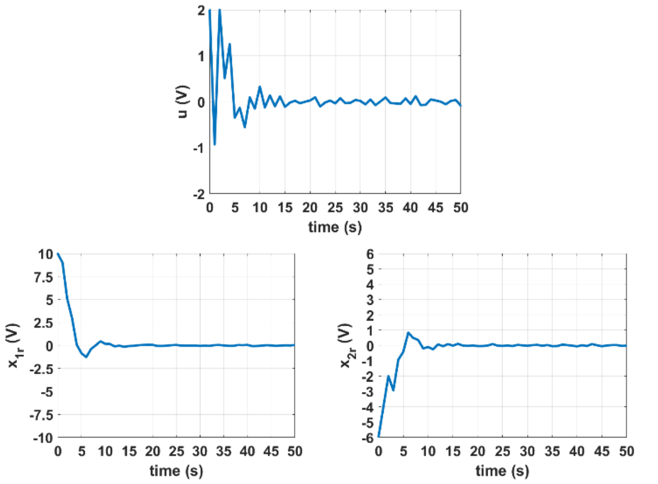


Fig. 20. The behavior of the system in Fig. 17 in case α = 0.95, h=0.5 (left diagrams) and α = 0.9, h = 1s (right diagrams). The signals u, x_{1r}, x_{2r}, x_1, x_2, x_{1c}, x_{2c} are obtained from the points (7), (1), (2), (3), (4), (5), (6).

Comparing both columns of Fig. 20, obtained for different values for h and close to each other values of the parameter α, it can be seen that for h = 1s and α = 0.9 the results are getting worse. The explanation is that an overcompensation is obtained, D-RBOC acting as $\tau = \alpha \cdot h = 0.9 \cdot 1 = 0.9\ s$.

In order to confirm the value of τ, more experiments were performed for h = 1 s. Using the compensation coefficient α = 0.475, the results from Fig. 21 are obtained. They correct substantially the results in case h = 1 s and α = 0.9, bringing them closer to the results from the left side of Fig. 20. Consequently, it can be stated that the time delay is approximately $\tau = 0.475\ s$.



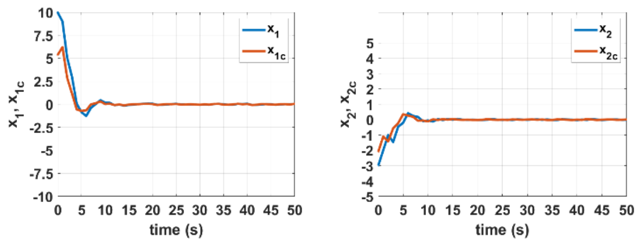


Fig. 21. The behavior of the system in Fig. 17 in case α = 0.475, h=1 s. Signals u, x_{1r}, x_{2r}, x_1, x_2, x_{1c} and x_{2c} are taken in points (7), (1), (2), (3), (4), (5), respectively (6).

Moreover, it can be observed that operating with $\alpha{\cdot}h$*=const. for* τ *= const.,* the value of *h* has an influence over the system behavior. In these conditions, it seems feasible adopting the value of $h > \tau$, so that $\alpha \in [0, 1)$ has a value as big as possible.

7. CONCLUSIONS

The discrete time optimal control of a double integrator plant, with limitation of the control signal domain, can be performed with the rule-based controller designed in this paper. It uses as a starting point the invariant realization to step signal of a double integrator plant.

The proposed algorithm conducts to very similar results as two other solutions available in the literature, making it a novel alternative. Like the case studies from this article illustrate, the obtained system's behavior is very close to the one of the control system that uses the general form from (Peng, 2015), form which through particularization provides the function from (Sun, 2010). As opposed to (Peng, 2015), our method implies a more intelligible and intuitive way of operating in the state space.

Just like in other time optimal control, the implementation of the solution in real situation is associated with the occurrence of time delay that alters the behavior of the control system. The effect of the time delay can be compensated by modifying the optimal control signal from the ideal case (without time delay). The feasibility of the proposed solution is proved both theoretically and experimentally on a system comprised of two dSPACE modules.

ACKNOWLEDGMENT

This work was supported by the Post-Doctoral Programme ''Entrepreneurial competences and excellence research in doctoral and postdoctoral programs - ANTREDOC'', project co-funded from the European Social Fund, contract no. 56437/24.07.2019.

REFERENCES

Ackermann, J., (1983). *Abtastregelung*, Bd.1, 2nd edition, Springer, Berlin, Heilderberg.

Åström, K.J. and Wittenmark, B., (2011). *Computer-Controlled Systems: Theory and Design*, Dover Publications, 3rd edition, Mineola, New York.

Athans, M. and Falb, P.L., (2007). *Optimal Control*, Dover Publications, Dover Publications, 3rd edition, Mineola, New York.

Dragomir, T.L. and Cîmpeanu, A., (2018). Improving the behaviour of a suboptimal control system, *Journal of Electrical Engineering Sciencies and Innovation*, vol. 3 (no. 4), pp. 351-362.

Gao, Z. (2004). On discrete time optimal control: a closed-form solution, *Proc. of the 2004 American Control Conference, Boston, June 30 – July 2, USA*, pp. 52-58.

Gao, Z. and Hu, S. (2004). On Properties and Applications of a New Form of Discrete Time Optimal Control Law, *Conference Record of the 2004 IEEE Industry Applications Conference, 39th IAS Annual Meeting*, pp. 1511-1518.

Han, J. (2009). From PID to Active Disturbance Rejection Control, *IEEE Trans. Industrial Electronics*, vol. 56 (no. 3), pp. 900-906.

Li, Y. (2010). A Complementary form of the Discrete Tracking-Differentiator, *2010 International Conference on Measuring Technology and Mechatronics Automation*, pp. 46-49.

Peng, J-l, Liu, X., Li, J, (2015). General form of Discrete Optimal Tracking Differentiator, *27th Chinese Control and Decision Conference*, pp. 629-633.

Stefan, O., Dragomir, T.L., Codrean and A., Silea, I. (2010). Issues of identifying, estimating and using delay times in telecontrol systems based on TCP/IP networks, *IFAC Proceedings Volumes 43 (23),* pp. 143-148.

Sun, B. and Sun, X. (2010). *Optimal control synthesis function of discrete-time systems*, [J], Control and Decision, 25 (3), 473-477.

Weihrich, G. (1973). Optimale Regelung linearer determi-nistischer Prozesse, R. Oldenbourg, München, Wien.