Paper

On the connections between optimal control, regulation and dynamic network routing

Andrzej Karbowski

Abstract — The paper is devoted to studying general features of dynamic network routing problems. It is shown that these problems may be interpreted as receding horizon optimal control problems or simply regulation problems. In the basic formulation it is assumed, that the nodes have no dynamics and the only goal of the optimization mechanism is to find the shortest paths from the source to the destination nodes. In this problem the optimization mechanism (i.e. the Bellman-Ford algorithm) may be interpreted as a receding horizon optimal control routine. Moreover, there is one-to-one correspondence between the Bellman optimal cost-to-go function in the shortest path problem and the Lyapunov function in the regulation problem. At the end some results of the application of the routing optimization algorithm to an inverted pendulum regulation problem are presented.

Keywords — stabilization, nonlinear control, optimal control, dynamic programming, data networks, routing algorithms.

1. General optimal control problem formulation

We consider a deterministic stationary discrete-time, dynamic system described by the state equation:

$$x_{k+1} = f(x_k, u_k), \;\; k = 0,1,2,...,\tau \tag{1}$$

where x_k, u_k, such that

$$x_k \in S \tag{2}$$

$$u_k \in U \tag{3}$$

are, respectively, the state and control vectors, and

$$f : S \times U \to S\,. \tag{4}$$

By S, U we denoted the subsets of some vector spaces of dimensions n and m, respectively.
For this system we would like to find a closed-loop control strategy

$$\pi = \{\mu_0, \mu_1, \ldots, \mu_\tau\}\,, \tag{5}$$

where $\mu_k(.), k = 0,1,\ldots,\tau$, is the kth stage control rule, admissible in the sense of state and control constraints, that is

$$u_k = \mu_k(x_k) \in U, \;\; \forall x_k \in S, \tag{6}$$

that minimizes the cost functional:

$$J(x_0) = \sum_{k=0}^{\tau} g\left(x_k, u_k\right) \tag{7}$$

with respect to both the policy π and the terminal time τ (i.e., the control horizon is free).
Let us select a point $\bar{x}$ from the state space S. We will assume, that for all $x \neq \bar{x}$ and any $u \in U$

$$g(x,u) > 0 \tag{8}$$

and there exists $\bar{u} \in U$ such that:

$$f(\bar{x}, \bar{u}) = \bar{x} \tag{9}$$

with

$$g(\bar{x}, \bar{u}) = 0\,. \tag{10}$$

For instance g may be a quadratic function:

$$g(x,u) = (x-\bar{x})'Q(x-\bar{x}) + \\ +(u-\bar{u})'R(u-\bar{u})\,, \tag{11}$$

where the matrix Q is positive semidefinite and the matrix R is positive definite.
Summing up, we consider an optimal control problem with a fixed terminal state, but free terminal time, defined by

$$\min_{\pi} \left\{ J(x_0) = \sum_{k=0}^{\tau} g\left(x_k, u_k\right) \right\} \tag{12}$$

$$x_{k+1} = f\left(x_k, u_k\right) \tag{13}$$

$$u_k = \mu_k(x_k) \in U \tag{14}$$

$$x_0 = \mathrm{x} \tag{15}$$

$$x_\tau = \bar{x} \tag{16}$$

where $\forall k \;\; x_k \in S$.
We assume, that the system (13)–(15) is controllable to the point $\bar{x}$ from every point of the state space.

2. Analysis

We will apply an analysis method inspired by Luenberger [6].

First, let us notice, that in our problem all functions are time-invariant (stationary). It means, that the solution will not depend on time, either. More precisely, the optimal trajectory from a given state $\mathtt{x}$ to the endpoint $\bar{x}$ is independent of the time k_0 at which $x_{k_0} = \mathtt{x}$. That is, if $x_0 = \mathtt{x}$ leads to the optimal trajectory $\{\tilde{x}_k\}$ for $k > 0$ with final time $\tau(\mathtt{x})$, then the condition $x_{k_0} = \mathtt{x}$ must lead to the trajectory $\{\tilde{x}_{k+k_0}\}$ with final time $\tau(\mathtt{x}) + k_0$. The delay of the initial time causes only delaying of the whole solution and the terminal time (i.e., the time of reaching the state $\bar{x}$) is simply an unknown function of the initial state only.

The optimal control rule is also a stationary function, that is for every k

$$u_k = \mu^*\left(x_k\right). \tag{17}$$

It must be so, because the initial control, as we have just stated, depends only on the initial state, not on the initial time, and we can repeat this reasoning at each time instant. Because of the assumptions (8)–(10) there will be:

$$\mu^*(\bar{x}) = \bar{u}. \tag{18}$$

If $\mu^*(.)$ is the optimal control rule, then we will obtain the following closed-loop system equation:

$$x_{k+1} = f\left(x_k, \mu^*\left(x_k\right)\right). \tag{19}$$

Let us notice, that due to Eqs. (18) and (9) the point $\bar{x}$ is an equilibrium point of the system (19) and according to the construction of the rule $\mu^*(.)$ the system eventually reaches $\bar{x}$. Hence, the system is stable.

Now, let us analyze formally the stability of the system and consider the optimal value function (that is the Bellman function, sometimes called "the optimal cost-to-go") $V_k(x)$ for this problem, expressed as:

$$V_k\left(x_k\right) = \sum_{l=k}^{k+\tau(x_k)} g\left(x_l, \mu^*(x_l)\right), \tag{20}$$

where the function $g(.,.)$ is defined by Eq. (10). This is the optimal (minimal) cost of the passage to $\bar{x}$ at time $k + \tau(x_k)$ when the initial point is x_k with time k. This function satisfies the following conditions:

(i) $V_k(\bar{x}) = 0$

(ii) $V_k(x) > 0$ for $x \neq \bar{x}$

(iii) $V_{k+1}(x_{k+1}) - V_k(x_k) = -g\left(x_k, \mu^*(x_k)\right) < 0$ for $x_k \neq \bar{x}$

Thus V – the Bellman function is a Lyapunov function and we proved the stability of the system.

3. Discrete-state version

In this section we will assume, that the sets S and U are finite and have, respectively, $T+1$ and $V+1$ elements. For the sake of simplicity we denote them by subsequent integers, that is:

$$S = \{0, 1, 2, 3, \ldots, T\} \tag{21}$$

$$U = \{0, 1, 2, 3, \ldots, V\} \tag{22}$$

Consequently we will have:

$$x_k \in S \subset \mathbf{Z}^n \tag{23}$$

$$u_k \in U \subset \mathbf{Z}^m \tag{24}$$

In these circumstances, for any state $x_k = i \in S$, a control $u_k = u \in U$ can be associated with a transition from the state $x_k = i$ to the state $f(i,u) = j \in S$. This passage is characterized by a cost:

$$c_{ij} = \min_{\substack{u \in U \\ f(i,u)=j}} g(i,u) \tag{25}$$

We assumed, that in the case when there are several controls $u \in U$, such that:

$$f(i,u) = j \tag{26}$$

we choose as the passage cost (25) the minimal cost among all costs corresponding to this passage.

Let us define now as a destination state $T \in S$. We will assume that the system may remain in this state, that is

$$\exists u^T \in U \;\; f(T, u^T) = T \tag{27}$$

and that the cost of being in this state equals zero, that is:

$$g(T, u^T) = 0. \tag{28}$$

In these conditions the state T is absorbing, that is if the system (1) passes to it, it remains in it for ever.

With this notation, we can interpret our deterministic optimal synthesis problem as a shortest path problem from an initial state 0 to the terminal state T (Fig. 1).

Let us denote now by $N(i)$ the set of all direct neighbours of the node i. The optimized dynamic programming algorithm for this problem will take the form:

$$J(i) = \min_{j \in N(i)} \{c_{ij} + J(j)\} \tag{29}$$

with the terminal condition:

$$J(T) = 0. \tag{30}$$

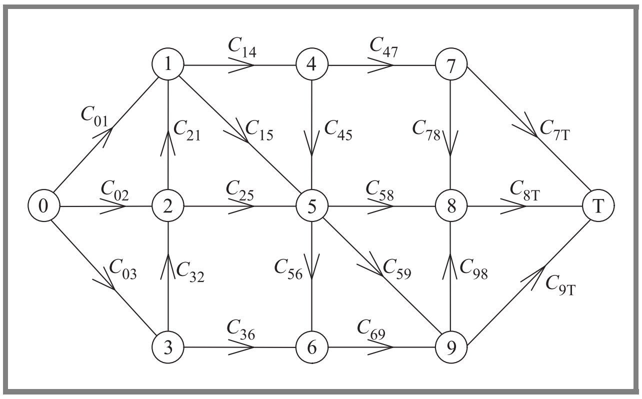


Fig. 1. Graph describing deterministic discrete optimal control problem with terminal state.

4. Routing problem and the Bellman-Ford asynchronous algorithm

The shortest path problem presented above, with a special structure resulting from the original optimal control formulation, may be immersed into a broader class of problems, namely into the class of routing problems. It consists in finding for every node from the set $S = \{0, 1, 2, \dots, I\}$ a table defining direct neighbours to which a load or message addressed to some remote node r should be transmitted. A destination may be every node from the set S.

Usually this problem is solved with the help of the asynchronous Bellman-Ford algorithm. This algorithm may be shortly described in the following way [1, 2, 4].

Let us denote the set of all arcs (i, j) between elements of the set S by A. Every arc from A can be characterized by the weight representing its length c_{ij}. The problem is to compute for every node $i \in S$ vectors x_{ir} of shortest distances from this node to the node r. We assume, that every arc in the directed graph $G = (S,A)$ has positive length and that there exists at least one path from every node to others. Then the shortest distances correspond to the unique fixed point of the monotone mapping $F : \mathbf{R}^{I+1} \times \mathbf{R}^{I+1} \to \mathbf{R}^{I+1} \times \mathbf{R}^{I+1}$ defined by $F_{rr}(x) = 0, r \in S$ and

$$F_{ir}(x) = \min_{\{j|(i,j)\in A\}} (c_{ij} + x_{jr}), \; i \in S. \tag{31}$$

The Bellman-Ford algorithm consists in the iteration

$$x_{ir} := F_{ir}(x) = \min_{\{j|(i,j)\in A\}} (c_{ij} + x_{jr}), \; \forall i, r \in S \tag{32}$$

or in the vector notation:

$$x := F(x) \tag{33}$$

and can be shown to converge to a fixed point

$$x^* = F(x^*) \tag{34}$$

when initialized with $x_{ij} = \infty \; \forall i \neq j$.

The convergence takes place also in the case of an asynchronous implementation [1, 4].

5. Integration

Taking into account conclusions drawn from the previous sections, we can write the following:

1. The optimal control policy in the receding horizon control problem for stationary systems with a Lagrange-type performance index is stationary.

2. When the terminal time is free, the optimal closed-loop control problem consists in finding the minimal cost trajectory from any point of the state space to a given point $\bar{x}$.

3. The deterministic closed-loop discrete optimal control problem with a fixed terminal state but with free terminal time (i.e. horizon) can be represented as a shortest path problem.

4. The shortest path may be solved with the help of the Bellman-Ford algorithm designed for routing problems, that might be implemented asynchronously (as in the Internet protocols RIP, IGP or Hello [2]).

Thus, having discretized the problem (12)–(16), connecting all resulting nodes according to the state equation (13) and solving the shortest path problem from all nodes to the node representing the point $\bar{x}$, we can transform the receding horizon optimal control problem into the routing problem and vice-versa.

6. Application of the routing algorithm to the stabilization of an inverted pendulum

To confirm experimentally the equivalence between routing algorithms and the feedback regulation the presented approach was tested on an example taken from [5].

A control law synthesis problem for a simple inverted pendulum was considered. The state variables of this system are the angle ξ and the angular velocity $\dot{\xi}$. The input u is a torque in the shaft, which is bounded to such an amount, that the pendulum cannot directly be turned from the hanging into the upright position. Instead, it is first necessary to "gain enough momentum", which requires a complex trajectory planning, even for this simple system. This nonlinearity posses the main difficulty for the feedback design in this example.

The system is described by the state equations:

$$\dot{x_1}(t) = x_2(t) \tag{35}$$

$$\dot{x_2}(t) = \sin x_1(t) + h(u(t)), \tag{36}$$

where $x_1 = \xi,\ x_2 = \dot{\xi}$ and $h(.)$ is the linear function with saturation, when the module of its argument exceeds 0.7, that is

$$h(u) = \begin{cases} -0.7 & u \le -0.7 \\ u & -0.7 < u < 0.7 \\ 0.7 & u \ge 0.7 \end{cases} \tag{37}$$

An interesting feature of the above system is that a continuous state feedback, which asymptotically stabilizes the system for all initial conditions, does not exist! The reason is, that for any continuous feedback there is a different than origin equilibrium point. More precisely, this point has a nonzero first coordinate. It must be so, because the function

$$f(x_1) = \sin x_1 + h(\mu(x_1, 0)) \tag{38}$$

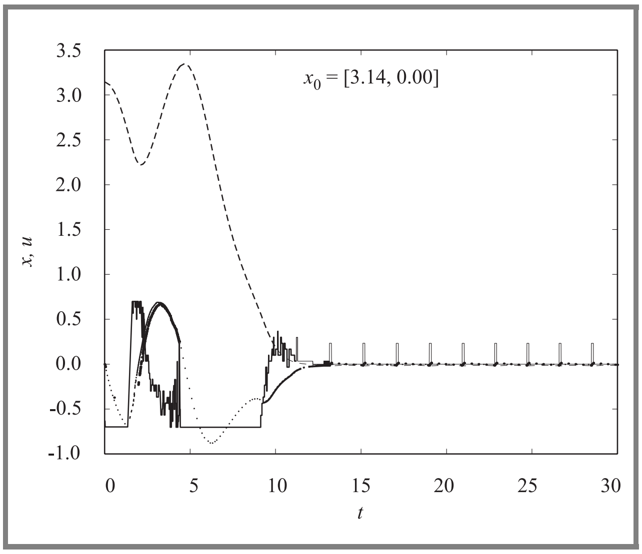


Fig. 2. Trajectories x_1 (---), x_2 ($\cdots$), u (—) for initial condition $[\pi, 0]$ and RB controller.

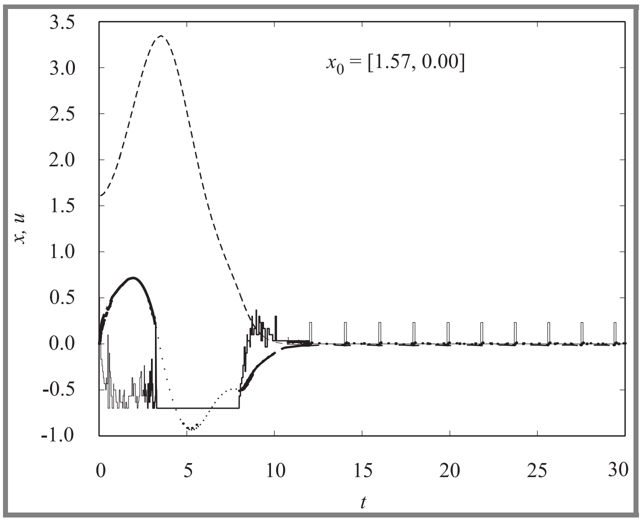


Fig. 3. Trajectories x_1 (---), x_2 ($\cdots$), u (—) for initial condition $[\frac{\pi}{2}, 0]$ and RB controller.

has the positive sign for $x_1 = \pi - arc\sin 0.8$ and the negative sign for $x_1 = \pi + arc\sin 0.8$. It means (from the Darboux theorem) that this function has a root in the interval $[\pi - arc\sin 0.8,\ \pi + arc\sin 0.8]$. In other words, the dynamic system (35)–(36) has an equilibrium point with a zero second and a nonzero first coordinate.

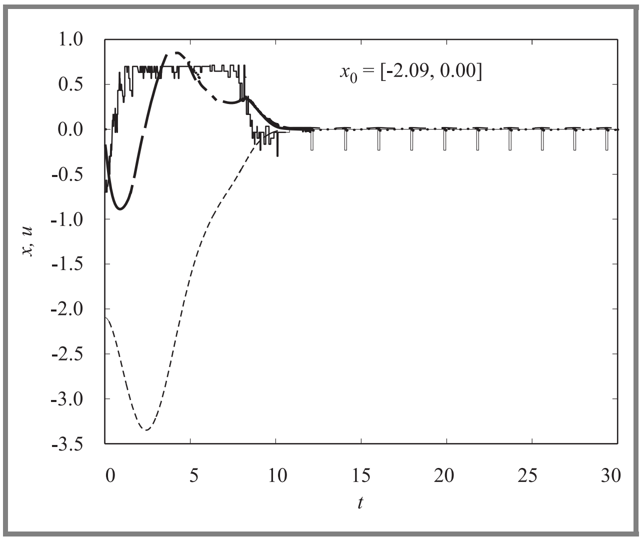


Fig. 4. Trajectories x_1 (---), x_2 ($\cdots$), u (—) for initial condition $[-\frac{2}{3}\pi, 0]$ and RB controller.

The system (35)–(36) was discretized under the following conditions:

- the conversion to the discrete-time representation was obtained via the Euler scheme for a sampling interval $T_s = 0.5$;
- as the state coordinate x_1 space, the interval $[-4, 4]$ was taken; it was discretized into 221 levels;
- as the state coordinate x_2 space, the interval $[-1.6, 1.6]$ was taken; it was discretized into 121 levels;
- the control space (the interval $[-0.7, 0.7]$) was divided into 20 equal subintervals;
- the cost function $g(x(t),\ u(t))$ was assumed to be quadratic, that is

  $$g(x, u) = x'Qx + u'Ru \tag{39}$$

  with

  $$Q = \begin{bmatrix} 5 & 0 \\ 0 & 2 \end{bmatrix} \tag{40}$$

  and $R = 2$.

It is worth noting, that according to the state equations (35)–(36), for $u = 0$, except of the origin, there are many other equilibrium points, those of coordinates: $[k\pi,\ 0]$, $k = 0, 1, 2, \ldots$. For instance, in the domain, there are two other (actually it is the same point, where the pendulum is hanging freely) such points.

Several experiments for different initial points were performed. All of them finished in the origin.

The resulting trajectories of the state and control variables are presented in Figs. 2–4. The abbreviation RB means routing based (controller).

For comparison, next figures (Figs. 5–7) present the same trajectories, obtained with the help of LQ methodology, without saturation of the function $h(.)$ (that is, it was replaced by identity). In those experiments, the system (35)–(36) was linearized in the origin, then the optimal static feedback matrix K (that is $u = K \cdot x$) was calculated, with the help of the Matlab Control Toolbox (procedure 'lqr').

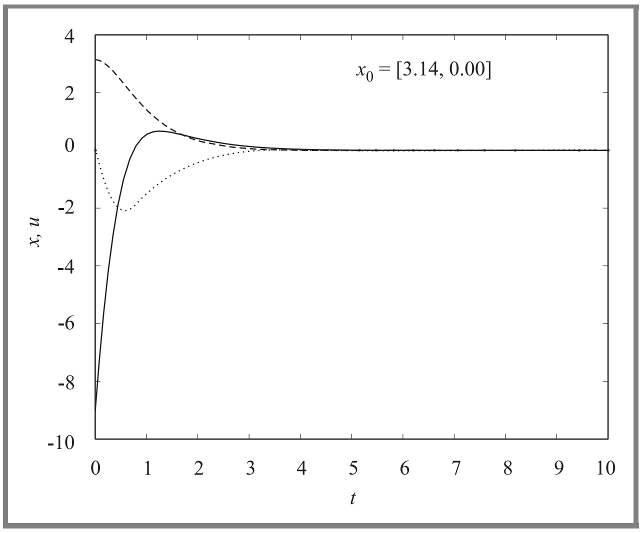


Fig. 5. Trajectories x_1 (---), x_2 $(\cdots)$, u (—) for initial condition $[\pi, 0]$ and LQ controller.

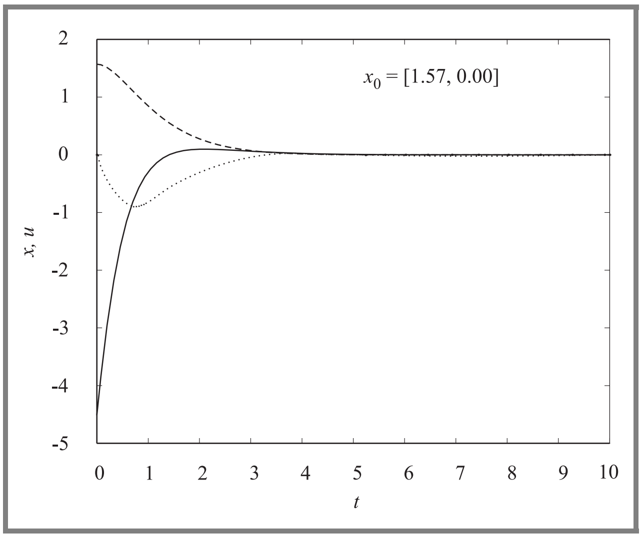


Fig. 6. Trajectories x_1 (---), x_2 $(\cdots)$, u (—) for initial condition $[\frac{\pi}{2}, 0]$ and LQ controller.

It is seen, that although in all cases the LQ controller was able to stabilize the pendulum, the control u was very big, out of the admissible interval $[-0.7, 0.7]$ of the previous (RB) case.

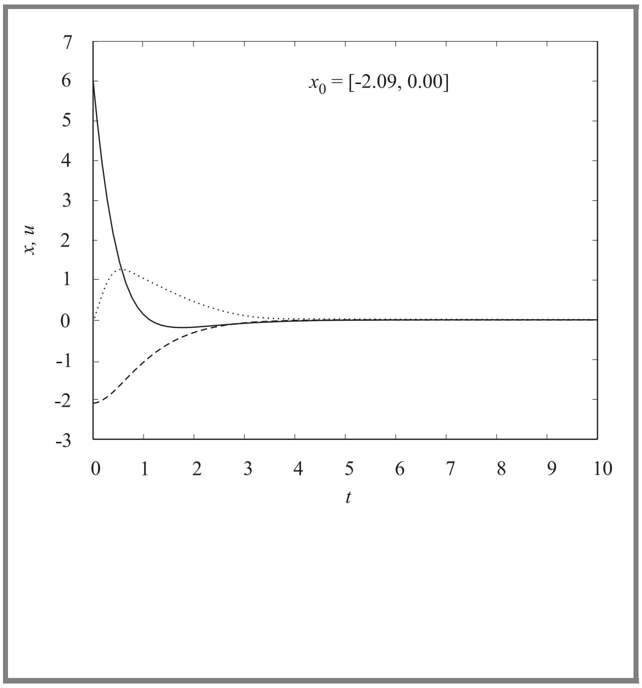


Fig. 7. Trajectories x_1 (---), x_2 $(\cdots)$, u (—) for initial condition $[-\frac{2}{3}\pi, 0]$ and LQ controller.

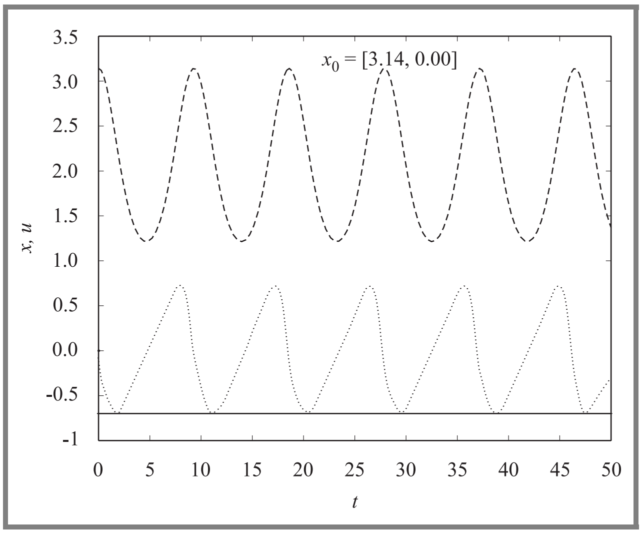


Fig. 8. Trajectories x_1 (---), x_2 $(\cdots)$, u (—) for moving pendulum and LQ controller with saturation for initial condition $[\pi, 0]$.

After the series of experiments it turned out, that in the case when the control constraints are taken into account while implementing the LQ control law, even for much greater values of the coefficient R, it is impossible to conduct the pendulum from the free ($[\pi, 0]$) to the upright position (Fig. 8). Let us recall, that it was not a prob-

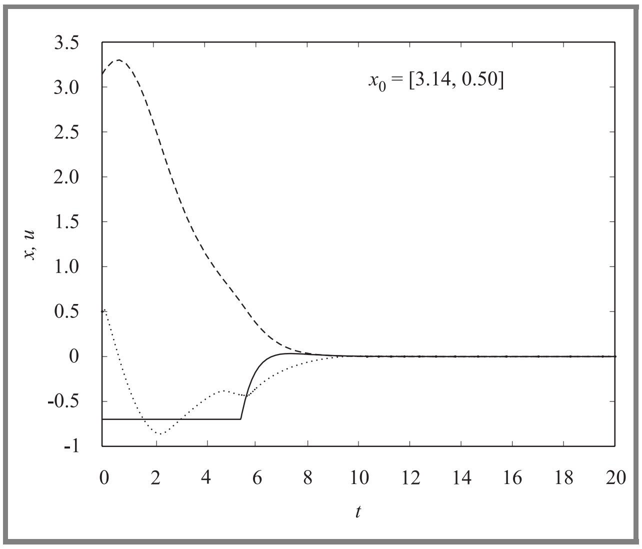


Fig. 9. Trajectories x_1 (---), x_2 $(\cdots)$, u (—) for moving pendulum and LQ controller with saturation for initial condition $[\pi, 0.5]$.

lem for RB controller (Fig. 2). However, after giving the pendulum some momentum, the LQ controller with saturation succeeded in regulating the pendulum to this position (Fig. 9).

7. Conclusions

The paper presented connections between a nonlinear stabilization problem and a network routing problem. The may idea lies in the formulation of the original regulation problem as a set of discrete-time receding horizon control problems, solved for all initial states. The optimal control rule may then be calculated (after state discretization) by the application of the Bellman-Ford algorithm, which is an elementary method for calculation of the shortest paths in networks.

An inverted pendulum case of study results showed, that the regulator obtained in this simple way has some advantages over classical LQ approach: it requires much smaller controls to move the state of the system to the equilibrium point neighbourhood, and it can successfully control the system even for initial conditions lying very far from the equilibrium point (that is, it is global). The drawbacks of this regulator are small oscillations around the terminal point, caused by discretization, and the longer time of regulation. Because of that, the best solution in the case of continuous nonlinear systems would be probably a hybrid regulator: discrete – routing based for points lying far from the terminal point and continuous – LQ methodology based, in its neighbourhood.

References

[1] D. Bertsekas and R. Gallager, *Data Networks*. Englewood-Cliffs: Prentice-Hall, 1992.

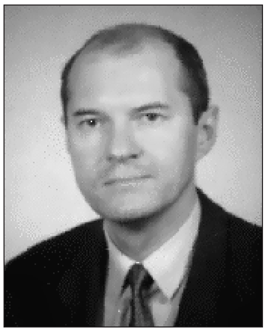
[2] D. E. Comer, *Internetworking with TCP/IP 4/e*. Vol. I : *Principles, Protocols, and Architecture*. Englewood Cliffs: Prentice-Hall, 2000.

[3] A. Isidori, *Nonlinear Control Systems*. London: Springer, 1995.

[4] A. Karbowski, E. Niewiadomska-Szynkiewicz, Red., *Obliczanie równoległe i rozproszone*. Warszawa: Oficyna Wydawnicza Politechniki Warszawskiej, 2001.

[5] G. Kreisselmeier and T. Birkhölzer, "Numerical nonlinear regulator design", *IEEE Trans. Autom. Contr.*, vol. 39, pp. 33–46, 1994.

[6] D. G. Luenberger, *Introduction to Dynamic Systems: Theory, Models and Applications*. New York: Wiley, 1979.



Andrzej Karbowski received his M.Sc. degree in electronic engineering (specialization automatic control) from Warsaw University of Technology (Faculty of Electronics) in 1983. He received the Ph.D. in 1990 in automatic control and robotics. He works as adjunct both at Research and Academic Computer Network (NASK) and at the Faculty of Electronics and Information Technology (at the Institute of Control and Computation Engineering) of Warsaw University of Technology. His research interests concentrate on data networks management, optimal control in risk conditions, decomposition and parallel implementation of numerical algorithms.

e-mail: A.Karbowski@ia.pw.edu.pl

Research and Academic Computer Network (NASK)

Wąwozowa st 18

02-796 Warsaw, Poland