Optimal Targeting in Chaos Control

A Discrete Hamiltonian Approach

Ulrich Vogl
Faculty of Electrical Engineering and Information Technology
HAW Amberg-Weiden, University of Applied Sciences
92224 Amberg, Germany
E-mail: U.Vogl@ieee.org

***Abstract*—One of the most interesting paradigms of chaos control is the possibility of switching a system between different unstable periodic orbits (UPOs) with effectively zero control energy. We give a robust method to find *finite-time optimal* transient trajectories, and show how to stabilize both, UPOs and transients, within the same LQ-framework. The method is quite general, and can also be used to drive a system from static stationary points to an UPO. To illustrate our approach we apply it to the controlled logistic map, and also to an experimental driven-pendulum setup.**

***Index Terms*—Chaos Control, nonlinear optimal control, targeting, time-dependent LQ, discrete Hamiltonian, driven pendulum.**

I. Introduction

In recent years much progress has been made in controlling nonlinear and especially chaotic systems, and a variety of algorithms have been developed to find and stabilize periodic solutions in chaotic systems [1]. Indeed, two of the most important paradigms in chaos control are stabilizing unstable periodic orbits (UPOs) embedded in an attractor and steering the system between different UPOs. The latter is of special interest, since it is well known [2], that any strange attractor has countable infinite UPOs embedded. If the UPOs are contained in the same attractor (i.e. the same invariant set of the dynamic flow), the energy needed to switch the system between these UPOs vanishes exponentially with the time N allowed for the transient trajectory.

In this paper we give a time-discrete algorithm, which yields for any given finite number of time steps N an optimal control signal $u(k)$ which switches the system between different UPOs. The method is universal and can e.g. also be used to find minimal-energy trajectories connecting static states outside an attractor with an UPO embedded in. This kind of transients are obviously a necessity in any practical or experimental setup. We can also connect UPOs embedded in different attractors, and find our way back to static stationary points. Since the method does not relay on chaos, also optimal trajectories for non-chaotic systems can be found.

Once a transient to a desired UPO has been found, the latter, and in most cases also the transient itself, need to be stabilized. To this end we apply a time-dependent LQ- (linear-quadratic) approach which will briefly be outlined in Section III.

Finally, we apply our method to a periodically driven nonlinear pendulum experiment. Short video sequences, demonstrating the practicability of the method in an experimental setup, can be watched at [3].

II. Optimal Trajectories

A. Problem Definition

In most modern applications control algorithms will be implemented in a digital, time-discrete fashion. Thus we assume that we can describe our physical system with a set of (nonlinear) difference equations. The goal is to find control signals and transient trajectories between two states in a finite time N, which leads to a system of difference equations with boundary conditions. The problem can thus be stated as:

$$\begin{aligned}\mathbf{x}(k+1) &= \mathbf{f}(\mathbf{x}(k),\mathbf{u}(k),k), \quad k=0,1,\dots N-1 \quad (1)\\ \mathbf{x}(0) &= \mathbf{a},\\ \mathbf{x}(N) &= \mathbf{b}.\end{aligned}$$

Here $\mathbf{x}(k) \in \mathbb{R}^n$ is a $n-$dimensional state vector, $\mathbf{u}(k) \in \mathbb{R}^m$ is the external control vector, and k is the (discrete) time index. The vector function $\mathbf{f}$ is a mapping of $\mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} \to \mathbb{R}^n$. As stated, we seek an optimal control sequence $\mathbf{u}(k), \quad k=0,1,\dots N-1$, in the sense that the following weighted energy functional

$$\mathcal{J} = \frac{1}{2}\sum_{k=0}^{N-1}\left(\mathbf{u}(k)^T,\mathbf{x}(k)^T\right)\begin{pmatrix}\mathbf{S} & \mathbf{R}\\ \mathbf{R}^T & \mathbf{T}\end{pmatrix}\begin{pmatrix}\mathbf{u}(k)\\ \mathbf{x}(k)\end{pmatrix} \quad (2)$$

is minimized. Here $\mathbf{S}$, $\mathbf{T}$ and $\mathbf{R}$ are weighting matrices, $\mathbf{T}$ and $\mathbf{S}$ are supposed to be symmetric. Furthermore, $\mathbf{S}$ must be positive definite (thus non-singular) and $\mathbf{T}-\mathbf{R}^T\mathbf{S}^{-1}\mathbf{R}$ positive semi-definite. In a first approach, one might choose $\mathbf{S}=\mathbf{1}_{m\times m}$, and $\mathbf{R}=\mathbf{T}=0$. In this case, only the external control signal $\mathbf{u}(k)$ is used for the optimality criterion (minimal total drive energy). Note, that in special cases it could be even useful to allow for time- dependence in the above weighting matrices.

B. Hamiltonian Approach

As in the time-continuous case (see e.g. [4]) we start defining the Lagrangian functional

$$\mathcal{L} = \mathcal{J} + \sum_{k=0}^{N-1}\mathbf{p}^T(k)\cdot\{\mathbf{f}(\mathbf{x}(k),\mathbf{u}(k),k)-\mathbf{x}(k+1)\}, \quad (3)$$

where the components of $\mathbf{p}(k)$ are Lagrange-multipliers. The next step is to perform a Legendre-transform by defining the Hamiltonian function. In this context $\mathbf{p}(k)$ is considered as new, *independent* system state vector, and is referred to as conjugate variable or (generalized) impulse.

The Hamiltonian function is then defined by

$$\mathcal{H}(\mathbf{x}(k),\mathbf{p}(k),\mathbf{u}(k),k) \equiv \mathcal{H}(k) = \mathcal{J} + \mathbf{p}^T(k)\cdot\mathbf{f}(\mathbf{x}(k),\mathbf{u}(k),k), \tag{4}$$

thus we have

$$\mathcal{L} = \sum_{k=0}^{N-1} \mathcal{H}(\mathbf{x}(k),\mathbf{p}(k),\mathbf{u}(k),k) - \mathbf{p}^T(k)\cdot\mathbf{x}(k+1). \tag{5}$$

Necessary conditions for an optimal trajectory are

$$\frac{\partial \mathcal{L}}{\partial \mathbf{x}(k)} = \frac{\partial \mathcal{H}(k)}{\partial \mathbf{x}(k)} - \mathbf{p}(k-1) = 0 \tag{6a}$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{p}(k)} = \frac{\partial \mathcal{H}(k)}{\partial \mathbf{p}(k)} - \mathbf{x}(k+1) = 0 \tag{6b}$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{u}(k)} = \frac{\partial \mathcal{H}(k)}{\partial \mathbf{u}(k)} = 0. \tag{6c}$$

Since $\mathbf{S}$ is invertible, $\mathbf{u}(k)$ can readily be calculated from (6c) to yield

$$\mathbf{u}(k) = -\mathbf{S}^{-1}\mathbf{p}^T(k)\cdot\frac{\partial \mathbf{f}(\mathbf{x}(k),\mathbf{u}(k),k)}{\partial \mathbf{u}(k)} - \mathbf{S}^{-1}\mathbf{R}\mathbf{x}(k). \tag{7}$$

If the system is linear in the external control $\mathbf{u}(k)$, the right hand side does not depend on $\mathbf{u}(k)$ any more. In all other cases (7) can either be solved analytically or numerically. For the relaxation method described below, implicit differentiation can be applied to yield the Jacobians $\mathbf{J}_f(k) = \frac{\partial \mathbf{f}(\mathbf{x}(k),\mathbf{u}(k),k)}{\partial \mathbf{u}(k)}$, thus avoiding the problem for explicitly solving (7) for $\mathbf{u}(k)$. The result can be used in (6a) and (6b) to form the canonical $2n$-dimensional system

$$\begin{aligned}
\mathbf{p}(k-1) &= \mathbf{f}_1(\mathbf{x}(k),\mathbf{p}(k),k), \quad k = 1,2,\dots N \\
\mathbf{x}(k+1) &= \mathbf{f}_2(\mathbf{x}(k),\mathbf{p}(k),k), \quad k = 0,1,\dots N-1 \\
\mathbf{x}(0) &= \mathbf{a}, \\
\mathbf{x}(N) &= \mathbf{b}.
\end{aligned} \tag{8}$$

Here the functions $\mathbf{f}_1$ and $\mathbf{f}_2$ are maps from $\mathbb{R}^{2n}\times\mathbb{R}\to\mathbb{R}^n$ and are given as

$$\begin{aligned}
\mathbf{f}_1(\mathbf{x}(k),\mathbf{p}(k),k) &= \frac{\partial \mathcal{H}}{\partial \mathbf{x}(k)} = \\
&\frac{\partial}{\partial \mathbf{x}(k)}\,\mathbf{p}^T(k)\cdot\mathbf{f}(x(k),\mathbf{u}(k),k)\big|_{\mathbf{u}=-\mathbf{S}^{-1}\mathbf{p}^T\cdot\frac{\partial\mathbf{f}}{\partial\mathbf{u}}-\mathbf{S}^{-1}\mathbf{R}\mathbf{x}}, \\
\mathbf{f}_2(\mathbf{x}(k),\mathbf{p}(k),k) &= \mathbf{f}(\mathbf{x}(k),\mathbf{u}(k),k)\big|_{\mathbf{u}=-\mathbf{S}^{-1}\mathbf{p}^T\cdot\frac{\partial\mathbf{f}}{\partial\mathbf{u}}-\mathbf{S}^{-1}\mathbf{R}\mathbf{x}}.
\end{aligned} \tag{9}$$

In (8) we have in total $2nN$ equations and also $2nN$ unknowns: $n\cdot(N-1)$ for the state trajectory $(\mathbf{x}(1),\mathbf{x}(2),\dots\mathbf{x}(N-1))$ and $n(N+1)$ unknown impulses $(\mathbf{p}(0),\mathbf{p}(1),\dots\mathbf{p}(N))$. Note that this formulation of the canonical Hamiltonian equations (8) comprise one iteration forward and one backward in time.

C. Solution methods

1) Shooting: If in (8) $\mathbf{f}_1$ is invertible in the variable $\mathbf{p}(k)$, we can define a totally forward directed difference equation system. To this end we define $\mathbf{p}'(k+1) = \mathbf{p}(k)$ and solve $\mathbf{f}_1$ for $\mathbf{p}'(k+1)$:

$$\mathbf{z}(k+1) \equiv \begin{pmatrix}\mathbf{p}'(k+1)\\ \mathbf{x}(k+1)\end{pmatrix} = \tag{10}$$

$$\begin{pmatrix}\mathbf{F}_1(\mathbf{x}(k),\mathbf{p}'(k),k)\\ \mathbf{F}_2(\mathbf{x}(k),\mathbf{p}'(k),k)\end{pmatrix} = \mathbf{F}(\mathbf{z}(k),k). \tag{11}$$

Thus we end up with an iteration of a $2n$-dimensional system $\mathbf{z}(k+1) = \mathbf{F}(\mathbf{z}(k),k)$. To find a solution which fits the boundary conditions $\mathbf{x}(0) = \mathbf{a}$, and $\mathbf{x}(N) = \mathbf{b}$, we use the start vector $\mathbf{z}_0 = \mathbf{z}(0) = \binom{\mathbf{p}'_0}{\mathbf{a}}$, where $\mathbf{p}'_0$ is an unknown variable. After $N-$fold iteration $\mathbf{F}^{(N)}(\mathbf{z}_0) := \mathbf{F}(\mathbf{F}(\mathbf{F}(\cdot\dots\mathbf{z}(0),N-2),N-1),N) = \binom{\mathbf{p}'_N}{\mathbf{x}(N)}$ we can read off the final state $\mathbf{x}(N)$. Thus the remaining problem is finding roots of the nonlinear equation in the unknown $\mathbf{p}'_0$

$$\begin{pmatrix}\mathbf{0} & \mathbf{1}\end{pmatrix}\cdot\mathbf{F}^{(N)}\begin{pmatrix}\mathbf{p}'_0\\ \mathbf{a}\end{pmatrix} - \mathbf{b} = 0. \tag{12}$$

For higher dimensional *nonlinear* systems however, solving this equation can become numerically very sensitive. After several iterations (N large), the function $\mathbf{F}^{(N)}$ can become extremely complicated and we encounter a high sensitivity to initial conditions $\mathbf{p}'_0$. This effect is well known for chaotic systems indeed.

As an aside, it is worth noting that the flux of (10) is *symplectic*. This means, that the Jacobian $\mathbf{J}_F(k) = \begin{pmatrix}\frac{\partial\mathbf{F}_1}{\partial\mathbf{p}'(k)} & \frac{\partial\mathbf{F}_1}{\partial\mathbf{x}(k)}\\ \frac{\partial\mathbf{F}_2}{\partial\mathbf{p}'(k)} & \frac{\partial\mathbf{F}_2}{\partial\mathbf{x}(k)}\end{pmatrix}$ has for all k the property

$$\mathbf{J}_F^T\,\mathbf{I}\,\mathbf{J}_F = \mathbf{I}, \tag{13}$$

where $\mathbf{I}$ is the following $2n\times 2n$ matrix $\mathbf{I} = \begin{pmatrix}\mathbf{0} & \mathbf{1}\\ -\mathbf{1} & \mathbf{0}\end{pmatrix}$. Note that $\mathbf{I}^2 = -\mathbf{1}_{n\times n}$. Equation (13) implies for example, that $\det(\mathbf{J}_F) = 1$ for all k, thus it is always invertible. In fact, solving symplectic flux equations is well investigated [5].

2) Relaxation Method: A much more robust method for solving (8) is the following relaxation-based method. It is especially suitable for finding optimal transients in chaotic systems, where the above-mentioned sensitivity to initial conditions are encountered. In fact, we can extend the so called "targeting method" [6], allowing state transients with minimal energy (in fact, the energy will exponentially vanish with transient time) of the control signal within a given *finite* transient time N.

We start out with linearizing (8) around a supposedly known start-trajectory:

$$\begin{aligned}
\mathbf{p}(k-1)+\boldsymbol{\eta}(k-1) &= \mathbf{f}_1(k) + \mathbf{J}_{11}(k)\boldsymbol{\xi}(k) + \mathbf{J}_{12}(k)\boldsymbol{\eta}(k), \\
k &= 1,2,\dots N \text{ and}
\end{aligned} \tag{14}$$

$$\begin{aligned}
\mathbf{x}(k+1)+\boldsymbol{\xi}(k+1) &= \mathbf{f}_2(k) + \mathbf{J}_{21}(k)\boldsymbol{\xi}(k) + \mathbf{J}_{22}(k)\boldsymbol{\eta}(k), \\
k &= 0,1,2,\dots N-1,
\end{aligned} \tag{15}$$

where the Jacobians are given here as

$$\begin{aligned}\mathbf{J}_{11}(k) &= \frac{\partial \mathbf{f}_1}{\partial \mathbf{x}(k)}, \mathbf{J}_{12}(k) = \frac{\partial \mathbf{f}_1}{\partial \mathbf{p}(k)}, \\ \mathbf{J}_{21}(k) &= \frac{\partial \mathbf{f}_2}{\partial \mathbf{x}(k)}, \mathbf{J}_{22}(k) = \frac{\partial \mathbf{f}_2}{\partial \mathbf{p}(k)}.\end{aligned} \tag{16}$$

Note, that with the indices chosen here, we have the desirable feature that the Jacobian is symmetric, i.e. $\mathbf{J}^T(k) = \mathbf{J}(k)$. The set of equations for the unknowns $\boldsymbol{\eta}(k)$ and $\boldsymbol{\xi}(k)$ can be casted in matrix form as follows: For $k = 1, 2, 3, \ldots N-1$ we have

$$\begin{pmatrix} -\mathbf{1} & \mathbf{J}_{11}(k) & \mathbf{J}_{12}(k) & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_{12}^T(k) & \mathbf{J}_{22}(k) & -\mathbf{1} \end{pmatrix} \begin{pmatrix} \boldsymbol{\eta}(k-1) \\ \boldsymbol{\xi}(k) \\ \boldsymbol{\eta}(k) \\ \boldsymbol{\xi}(k+1) \end{pmatrix} = \begin{pmatrix} \mathbf{p}(k-1) - \mathbf{f}_1(k) \\ \mathbf{x}(k+1) - \mathbf{f}_2(k) \end{pmatrix}.$$

At the boundaries $k = 0$ and $k = N$ we have

$$\begin{pmatrix} -\mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{J}_{12}^T(0) & \mathbf{J}_{22}(0) & -\mathbf{1} \end{pmatrix} \begin{pmatrix} \boldsymbol{\xi}(0) \\ \boldsymbol{\eta}(0) \\ \boldsymbol{\xi}(1) \end{pmatrix} = \begin{pmatrix} \mathbf{x}(0) - \mathbf{a} \\ \mathbf{x}(1) - \mathbf{f}_2(0) \end{pmatrix} \text{ and} \tag{17}$$

$$\begin{pmatrix} -\mathbf{1} & \mathbf{J}_{11}(N) & \mathbf{J}_{12}(N) \\ \mathbf{0} & -\mathbf{1} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{\eta}(N-1) \\ \boldsymbol{\xi}(N) \\ \boldsymbol{\eta}(N) \end{pmatrix} = \begin{pmatrix} \mathbf{p}(N-1) - \mathbf{f}_1(N) \\ \mathbf{x}(N) - \mathbf{b} \end{pmatrix} \tag{18}$$

respectively. Stacking the matrices above for $k = 0, 1, \ldots N$ into an almost block-diagonal matrix $\mathbf{Z}$, the vector of unknowns to a vector $\boldsymbol{\zeta}$ and the right hand vector (error vector) to $\mathbf{e}$, we end up with a matrix equation of the form

$$\mathbf{Z}\,\boldsymbol{\zeta} = \mathbf{e}. \tag{19}$$

Equation (8) is then solved by the following *algorithm*:

1) "Guess" some initial trajectory $\mathbf{x}^{(0)}(k), \mathbf{p}^{(0)}(k)$. In most cases a very crude guess will be sufficient.
2) Calculate the Jacobians according to (16) and form the matrix $\mathbf{Z}^{(0)}$. Calculate the r.h.s error vector $\mathbf{e}^{(0)}$.
3) Solve the linear equation $\mathbf{Z}^{(0)}\boldsymbol{\zeta}^{(0)} = \mathbf{e}^{(0)}$. Due to symmetry and the diagonal-dominated structure of $\mathbf{Z}$ this is in general no problem and can be done using standard linear algebra packages.
4) Calculate the relative residual error

$$\varepsilon^{(0)} = \left\|\boldsymbol{\zeta}^{(0)}\right\|_2 \left(\left\|\mathbf{x}^{(0)}(k)\right\|_2 + \left\|\mathbf{p}^{(0)}(k)\right\|_2\right)^{-1}. \tag{20}$$

5) If the residual error is smaller than a given limit (e.g. 10 times machine precision), end the algorithm. Else, calculate the update parameter γ according to

$$\gamma = \begin{cases} \alpha & \varepsilon^{(0)} > \beta \\ 1 & \varepsilon^{(0)} \le \beta \end{cases}. \tag{21}$$

For practical implementation the values $\alpha = 0.4$ and $\beta = 0.05$ have proved good performance.
6) Update the state- and impulse vectors according to

$$\begin{aligned}\mathbf{x}^{(1)}(k) &= \mathbf{x}^{(0)}(k) + \gamma\boldsymbol{\xi}^{(0)}, \\ \mathbf{p}^{(1)}(k) &= \mathbf{p}^{(0)}(k) + \gamma\boldsymbol{\eta}^{(0)}.\end{aligned} \tag{22}$$

Then continue the iteration with step 2), using these updated vectors.

This algorithm shows very robust behavior, even in complicated situations and for large N. Reducing the update gradient to a factor of α in (21, 22) if the error is large, avoids updating into the wrong direction, if the trajectory is still far away from a converged solution. Of course, the algorithm may be refined further, e.g. using a continuous behavior of $\gamma(\varepsilon)$. However, numerical experiments have shown, that this does not improve convergence dramatically. Similar methods are used for solving differential boundary value equations, cf. e.g. [7].

3) Optimal targeting: The true advantage of the relaxation algorithm emerges in chaos control applications. Here the initial trajectory can be used from a targeting procedure ($\mathbf{x}^{(0)}(k) = \mathbf{x}_{\text{Target}}(k)$ and $\mathbf{u}^{(0)}(k) \equiv 0$). We can choose $\mathbf{p}^{(0)}(k) \equiv 0$, letting the system do most of the control work by itself. The latter choice is motivated by (7), since we are interested in solutions with energy $\mathcal{J}$ almost zero (if the mixing weight $\mathbf{R}$ in (2) is small or zero). Small corrections necessary are then achieved by only very few iterations of the relaxation algorithm. To find $\mathbf{x}_{\text{Target}}(k)$ we iterate (1) with $\mathbf{u}^{(0)}(k) \equiv 0$ $N-$ times, starting at a slightly displaced initial state $\mathbf{x}^{(0)}(k) = \mathbf{a} + \boldsymbol{\varepsilon}_a$. The norm function

$$\varepsilon_b(\boldsymbol{\varepsilon}_a) = \left\|\mathbf{b} - \mathbf{f}^{(N)}(\mathbf{a} + \boldsymbol{\varepsilon}_a, 0, k)\right\|_2 \tag{23}$$

measures the difference of $\mathbf{x}^{(0)}(N)$ to the desired state $\mathbf{b}$, and will typically show very sharp notches as function of $\boldsymbol{\varepsilon}_a$, see also Fig. 23. Other methods for finding targeting solutions use random-seed initial states, which are spread in the vicinity of $\mathbf{a}$ [9]. Iterating (1) with such a "notch-solution" as initial state $\mathbf{x}(0) = \mathbf{a} + \boldsymbol{\varepsilon}_{a,\text{notch}}$ gives us a good candidate for $\mathbf{x}_{\text{Target}}(k)$. In fact, as shown in the application section, this extended targeting method yields trajectories with the property of exponentially decaying energy of the control signal as function of length N:

$$\sum_{k=0}^{N} \|\mathbf{u}(k)\|_2^2 \sim e^{-N}. \tag{24}$$

III. Stabilizing Trajectories

Since most trajectories we are interested in are non-stable, implying (at least one) positive Lyapunov exponent, we need to stabilize them. This is especially important in the case of chaos control and targeting, where we want to connect UPOs embedded in the same attractor. Of course the optimal transients discussed above are also very close to that strange attractor, and in fact $\mathbf{x}_{\text{Transient}}(k)$ converges to the attractor-set in the limit $N \to \infty$.

The method described below relies on the standard LQ (linear-quadratic) approach [8] and can be seen as a generalization of the OGY method [10] well known to the chaos-control community. The only difference to the standard textbook result is that we have to deal with a time-variant system here. We briefly outline this method here for completeness. The first step is linearizing (1) around the desired full trajectories (UPOs and transients) $\mathbf{x}_0(k)$ and $\mathbf{u}_0(k)$, resulting in a *time-dependent* linear difference equation:

$$\boldsymbol{\xi}(k+1) = \mathbf{A}(k)\boldsymbol{\xi}(k) + \mathbf{B}(k)\boldsymbol{\eta}(k). \tag{25}$$

Here $\boldsymbol{\xi}(k) = \mathbf{x}(k) - \mathbf{x}_0(k) \in \mathbb{R}^n$is the n-dimensional state vector deviation, $\boldsymbol{\eta}(k) = \mathbf{u}(k) - \mathbf{u}_0(k) \in \mathbb{R}^m$ is the external control vector deviation. Note that for UPOs we have $\mathbf{u}_0(k) \equiv 0$. The time-dependent matrices $\mathbf{A}(k)$ and $\mathbf{B}(k)$ are the Jacobians of $\mathbf{f}$ with respect to $\mathbf{x}$ and $\mathbf{u}$.

We define a linear (L) control law

$$\boldsymbol{\eta}(k) = -\mathbf{K}(k)\boldsymbol{\xi}(k), \tag{26}$$

where the control coefficients $\mathbf{K}(k)$ to stabilize the system are found by minimizing the quadratic (Q) functional

$$\mathcal{I} = \frac{1}{2}\sum_{k=0}^{\infty} \boldsymbol{\xi}^T(k)\mathbf{U}\boldsymbol{\xi}(k) + \boldsymbol{\eta}(k)\mathbf{V}\boldsymbol{\eta}(k), \tag{27}$$

where $\mathbf{U}$ and $\mathbf{V}$ are positive definite weighting-matrices of states and control signal, respectively. A solution to this problem is provided by the Riccati *backward* iteration

$$\begin{aligned}
\mathbf{P}(k) &= \mathbf{U} + \mathbf{A}^T(k)\mathbf{P}(k+1)\mathbf{A}(k) - & (28)\\
&\mathbf{A}^T(k)\mathbf{P}(k+1)\mathbf{B}(k)\left(\mathbf{V} + \mathbf{B}^T(k)\mathbf{P}(k+1)\mathbf{B}(k)\right)^{-1} \cdot &\\
&\mathbf{B}^T(k)\mathbf{P}(k+1)\mathbf{A}(k) & (29)\\
\mathbf{K}(k) &= \left(\mathbf{U} + \mathbf{B}^T(k)\mathbf{P}(k+1)\mathbf{B}(k)\right)^{-1} \cdot &\\
&\mathbf{B}^T(k)\mathbf{P}(k+1)\mathbf{A}(k). &
\end{aligned}$$

To compute e.g. the control coefficients for a UPO$_1$-transient-UPO$_2$ scenario, we create the total trajectory by concatenating the parts from UPOs and transients. In addition, UPO$_2$ will be appended several times, to ensure the iteration converges to a periodic solution. The *backward* iteration is then started with $\mathbf{P}(M) = \mathbf{U}$, where M is the last index in the full trajectory described above.

IV. Applications

A. Logistic Map

In the following we illustrate our method for a very simple yet nontrivial example, the controlled logistic map [11]:

$$x(k+1) = r\ x(k)(1 - x(k)) + u(k). \tag{30}$$

States x and control u are scalar signals and $1 < r \leq 4$ is a model parameter. Generally we have $|u(k)| \ll 1$ and care must be taken that $0 < x(k) < 1$ for all times. The logistic map is well known to exhibit chaotic behavior for $r > r_\infty \approx 3.57$. The functional (2) is chosen as

$$\mathcal{J} := J_u(N) = \frac{1}{2}\sum_{k=0}^{N} u(k)^2. \tag{31}$$

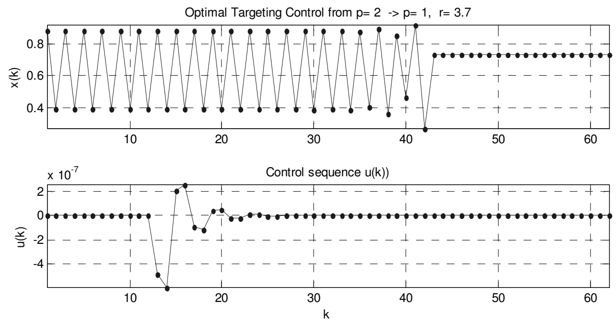

Fig. 1. Transition from a period $p = 2$ orbit to an (instable) stationary point with $r = 3.7$ (chaotic regime). State $x(k)$ (upper) and control signal $u(k)$ (lower graph). Note that the control signal is to be multiplied by10^{-7}.

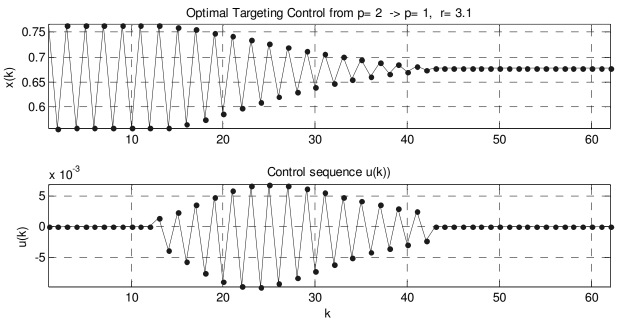

Fig. 2. Transition from a period $p = 2$ orbit to an (instable) stationary point in a *non*-chaotic scenario ($r = 3.1$). State $x(k)$ (upper) and control signal $u(k)$ (lower graph). The the control signal is to be multiplied by 10^{-3} here.

Equation (7) reduces to $u(k) = -p(k)$ and the canonical equations (8) read here

$$\begin{aligned}
p(k-1) &= r\ p(k)(1 - 2x(k)) \text{ and} & (32)\\
x(k+1) &= r\ x(k)(1 - x(k)) - p(k). &
\end{aligned}$$

The symmetric Jacobian (16) is

$$\mathbf{J}(k) = \begin{pmatrix} -2rp(k) & r(1-2x(k)) \\ r(1-2x(k)) & -1 \end{pmatrix}. \tag{33}$$

Periodic orbits (UPOs) in the chaotic regime and optimal transients are easily found using the relaxation method. In Fig.1 we display a rather long transient ($N = 30$) from a period $p = 2$ UPO to an (instable) stationary state. Since we are in the chaotic regime here ($r = 3.7$) the control signal has a very low amplitude ($u \ll x$) (scale $\sim 10^{-7}$). It is also remarkable, that the action of the control signal $u(k)$ is not immediately visible in the state signal $x(k)$. It takes a delay of about 25 time steps, until a significant impact becomes evident in the graph.

The same setup has been repeated for the *non*-chaotic case with $r = 3.1$, cf. Fig. 2. The behavior of the control signal is quite different, and the scale of the control signal $u(k)$ is much larger ($\sim 5 \cdot 10^{-3}$) here. Yet the desired target state $x(N) = b \approx 0.67$ is exactly reached within finite time.

In Fig. 3 the r.m.s of the energy $J_u(N)^{1/2}$ needed for the transient control signal in dependence of the transition length N is displayed on a logarithmic scale. Exponential decay of control energy as in (24) is clearly seen for the chaotic scenario $r = 3.7$. In the *non*-chaotic case $r = 3.1$ the control energy flattens out and increasing N does not yield any energy savings.

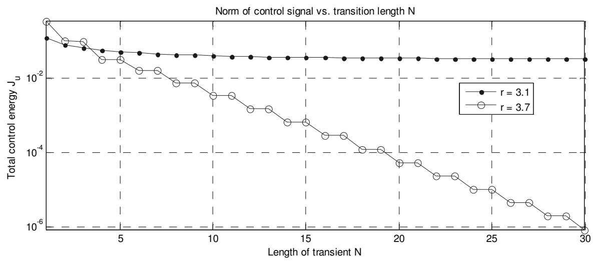

Fig. 3. Norm of control signal vs. transition length N of transient control signal for the chaotic ($r = 3.7$) and the non-chaotic regime ($r = 3.1$).

B. Driven Nonlinear Pendulum

1) System description: Next we consider a rigid pendulum hanging (with low friction) on a sled, which is driven by a stepper motor, cf. Fig. 4. Length of the pendulum is $l \approx 1.2$m and sampling period is $T_0 = 1/(20\text{Hz})$. Of course, the steps for the motor are interpolated (factor 2^{13}), which is achieved by an interpolation filter realized with a fix-point signal processor. The angle ϕ of the pendulum is measured by an high-resolution optical sensor, the corresponding signal $\phi(kT_0)$ is returned to the signal processor, which in turn can drive the sled position $s(kT_0)$ via the motor.

The time-continuous equation of motion for the system reads

$$\ddot{\phi} + \rho(\dot{\phi}) + \omega_p^2 \sin(\phi) = -\omega_p^2 \cos(\phi)\frac{\ddot{s}}{g}, \tag{34}$$

where s is the position of the sled (the acceleration $\ddot{s}$ will be used as the proper control signal u) and ω_p is the eigenfrequency of the pendulum for small angles ϕ. The function $\rho(\dot{\phi})$ resembles a nonlinear (yet continuous) friction function, which combines a Stribeck type friction [12] with a quadratic term in $\dot{\phi}$. The latter simulates air friction and becomes increasingly important for orbits with fast rotating pendulum and high-speed transients. For a more complete description also the air friction induced by the speed of the sled can be considered, but this is neglected here for simplicity. Using standard techniques, this continuous equation of motion can be transformed into a system of difference equations compatible to (1):

$$\mathbf{x}(k+1) \quad = \quad \mathbf{f}(x_1(k), x_2(k), x_3(k), x_4(k), u(k), k) =$$
$$\begin{bmatrix} c_1 \sin(\pi x_1) + c_2 \cos(\pi x_1)(u+d) + x_1 + x_2 - \rho(x_1) \\ c_1 \sin(\pi x_1) + c_2 \cos(\pi x_1)(u+d) + x_2 - \rho(x_1) \\ u + x_3 \\ x_3 + x_4 \end{bmatrix}. \tag{35}$$

The state vector is $n = 4$-dimensional and the control signal u is scalar ($m = 1$). The state variables $\mathbf{x(k)}$ are connected to $\phi(k)$ and $s(k)$ by $x_1(k) = \phi(k)$, $x_2(k) = \phi(k-1)$, $x_3(k) = v(k) = v(k-1)+u(k-1)$ where $v(k) = s(k)-s(k-1)$ is the speed of the sled and $x_4(k) = s(k) = s(k-1) + v(k-1)$ is its position (we let $T_0 \equiv 1$ to simplify notation). Basically, the third and fourth component of $\mathbf{f}$ doubly integrate the sled acceleration $u(k)$ to the sled position $s(k)$, which is the variable being directly accessible by the control output of the signal processor. With $d(k) = d_0 \sin(2\pi\alpha k)$ we realize a periodically driven system with frequency α/T_0 and amplitude d_0, rendering the difference equations non-autonomous. The nonlinear pendulum has only 2 degrees of freedom (ϕ and $\dot{\phi}$), which is not sufficient to create chaotic motion. With the external (periodic) force, however, we obtain a third degree of freedom. It is well known [13] that 3 degrees of freedom in non-linear systems suffice to produce chaotic motion. The parameters c_1, c_2 and also the parameters describing the friction function ρ are fitted by experimental measurements.

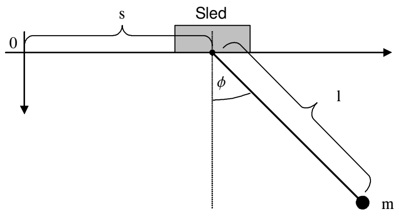

Fig. 4. Experimental driven pendulum setup.

2) Periodic orbits: Calculation of (8) from (35) and applying the relaxation method for state transitions is straightforward. We can create all kinds of optimal transients, e.g. to drive the pendulum into a static *inverse* state ($d_0 \equiv 0, \phi = \pi, \dot{\phi} = 0$) starting from the stable state ($\phi = 0, \dot{\phi} = 0$). It is more interesting, however, to drive the system into the chaotic regime, which becomes manifest for a wide range of frequencies α and amplitudes d_0. In this case the flux of the motion is part of a strange attractor, which hosts an abundant number of (unstable) periodic orbits. These orbits, having the property $\mathbf{x}(N+k) = \mathbf{x}(k)$, can be found by fixing $\alpha = 1/N$ and searching for the roots of $\mathbf{\Phi}$ defined by:

$$\mathbf{\Phi}(\mathbf{x}) = \mathbf{f}^{(N)}(\mathbf{x}, 0, k) - \mathbf{x} = 0 \text{ where } \mathbf{u}(k) \equiv 0 \tag{36}$$

and $\mathbf{f}^{(N)}(\mathbf{x}) = \mathbf{f}(\mathbf{f}(\mathbf{f}(\ldots\mathbf{f}(\mathbf{x}, 0, k)\ldots)$. Since $\mathbf{x}$ is in general higher dimensional (4-dimensional in this example) an analytical treatment of (36) is not possible. A practical method for finding UPOs is based on a monte-carlo method: randomly select a (physically reasonable) state $\mathbf{x}_0$, which is then to be used as initial trial solution for a standard Gauss-Newton algorithm. If no convergence appears, randomly select another $\mathbf{x}_0$. To drive the system from the stable state ($\phi = 0, \dot{\phi} = 0$) into an UPO, simply select $\mathbf{a} = (0, 0, 0, 0)^T$ as initial state and use any state *on* the UPO for $\mathbf{b}$. The transient can then be found easily using the relaxation method.

3) Optimal Targeting: To connect two UPOs with (almost) no expense of energy we can proceed as follows: select $\mathbf{a}$ as state on UPO$_1$, $\mathbf{b}$ as state on UPO$_2$. Then use (23) to find a prototype transient ($u \equiv 0$) between the orbits. Here we vary the initial state $\mathbf{a}$ in the x_1-direction only, i.e.

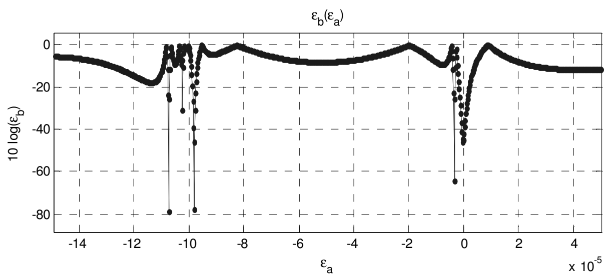

Fig. 5. Example of function (23) for the periodically driven pendulum. Note the logarithmic (dB) scale.

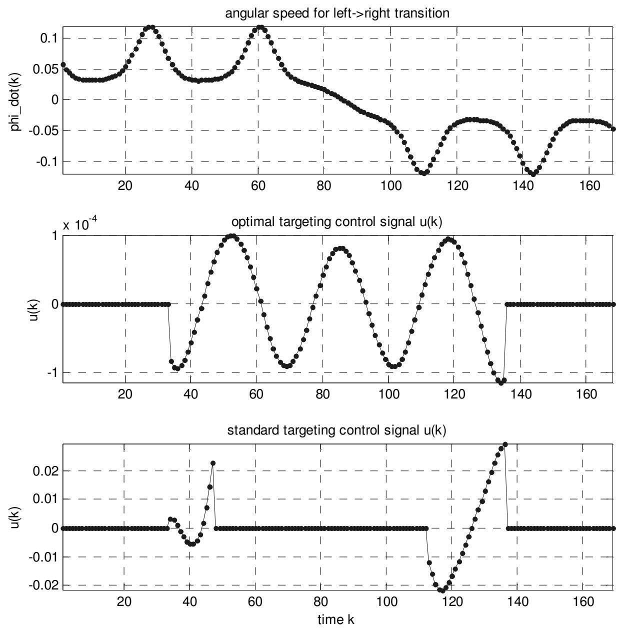

Fig. 6. Comparison between standard targeting method and optimal targeting for a left/right rotation transition of the pendulum. Angular speed of pendulum (upper), optimal targeting control acceleration $u(k)$ (middle), standard tageting of the same trajectory (lower). Note the different scales.

$\mathbf{x}(0) = \mathbf{a} + \varepsilon_a \mathbf{e}_1$ with $\mathbf{e}_1$ the unit vector in x_1-direction. If the UPOs are contained in the same attractor, ε_b will show distinct notches. An example of this function for a left/right alteration in rotation of the pendulum is shown in Fig. 5. As described before, $\min\{\varepsilon_b\}$ will yield a starting trajectory $\mathbf{x}^{(0)}(k) = \mathbf{x}_{\text{Target}}(k)$, $u^{(0)}(k) \equiv 0$ for the relaxation algorithm. Few iterations lead here to an optimized finite-time orbit-connecting control signal $u(k)$.

We give an example for an (almost) energy-free left/right rotation alteration. The additional acceleration $u(k)$, needed using the standard targeting approach, is compared to our optimal targeting method (Fig. 6). Clearly, standard targeting works with two "wing beats of the butterfly", the first is needed to leave UPO$_1$, the second to lock into the new UPO$_2$. In between u is exactly zero. In contrast, the energy of our optimal trajectory is distributed over the whole transient, which minimizes the overall energy needed. In this example we have an energy gain by a factor $J_u(u_{\text{Standard}})/J_u(u_{\text{Optimal}}) \approx 800$ for the same length of the transient $N = 70$.

This left/right rotation switch (among other UPO switches) of the pendulum can be watched on [3]. The additional acceleration $u(k)$ needed to perform the change in rotation is so small, that it can not be observed by eye. In the video initial and finite UPOs are illuminated by green or blue LEDs, whereas during the transient phase red LEDs are activated. The sled is continuously and periodically driven with fixed frequency α and amplitude $d(k)$.

V. Conclusion

Based on quadratic optimality criteria we discussed a method, to actuate any time-discrete system dynamics into a defined state within a finite time. When applied to nonlinear systems containing one ore more chaotic attractors, we can find optimal "almost zero" control signals, which connect different periodic orbits contained in the same attractor. In our application to the chaotic pendulum we could gain a factor of about 800 in energy saving compared to the standard targeting procedure given the same transient time. Moreover, optimal transients can be found also outside the attractor. This allows for connecting non-chaotic to chaotic regimes, and of course any other transition, e.g. between static points. A time dependent LQ-approach to stabilize UPOs and transients has also been discussed. Application to an experimental pendulum setup has been lined out, and many of the UPOs and trajectories can be seen as video clip on [3].

Using the Hamiltonian approach to linear systems yields very efficient algorithms, which can be implemented under hard real time conditions. Work to apply this idea to fast laser scan-heads is currently in progress.

Acknowledgment

The author would like to thank G. Mandel for fruitful discussions and all kinds of technical support.

References

[1] E. Schöll and H.G. Schuster, "Handbook of Chaos Control," 2nd ed., Wiley-VCH, 2008. (references)

[2] R. Badii, E. Brun, M. Finardi, L. Flepp, R. Holzner, J. Parisi,C. Reyl, and J. Simonet, Rev. Mod. Phys. 66, p. 1389, 1994.

[3] U. Vogl, (2011, Nov.). Videos of Chaotic Pendulum, HAW-AW, Amberg, Germany. [Online]. Available: http://www.haw-aw.de/vogl/alias/targeting

[4] V.I. Arnold, "Mathematical Methods of Classical Mechanics", Springer-Verlag, 1989.

[5] C.D. Ahlbrandt and A.C. Peterson, "Discrete Hamiltonian Systems", Kluwer Academic Publishers, 1996.

[6] T. Shinbrot et al., "Using the sensitive dependence of chaos (the "butterfly effect") to direct trajectories in an experimental chaotic system. Phys. Rev. Lett., 68: 2863-2866, 1992.

[7] P.P. Eggleton, Monthly Notices of the Royal Astronomical Society, vol. 151, pp. 351-364, 1971.

[8] P. Lancester, and L. Rodman, "Algebraic Riccati Equations", Oxford Science Publications, Clarendon Press, Oxford, 1995.

[9] E. E. N. Macau and C. Grebogi: "Driving trajectories in complex systems," Phys. Rev. E 57, 5337-5346, 1998.

[10] E. Ott, C. Grebogi and J.A. Yorke, "Controlling chaos". Phys. Rev. Lett. 64(11), 1196-1199, 1990.

[11] R.M. May, "Simple Mathematical Models with very Complicated Dynamics", Nature 261, 459, 1976

[12] R. Stribeck, "The basic properties of sliding and rolling bearings", (Die wesentlichen Eigenschaften der Gleit- und Rollenlager) Zeitschrift des Vereins Deutscher Ingenieure, Nr. 36, Band 46, p. 1341-1348, 2002.

[13] E. Ott, "Chaos in Dynamical Systems", Cambridge University Press, 1993.