Proceedings of Inverse Problems in Engineering: Theory and Practise
3rd Int. Conference on Inverse Problems in Engineering
June 13-18, 1999, Port Ludlow, Washington, USA

APP03

ILL-POSED PROBLEMS IN CLOSE-LOOP PREDICTIVE CONTROL

Tomas Hruz[1]
Department for Informatics at Mathematical Institute
Slovak Academy of Sciences, P.O.Box 56,
Dubravska cesta 9, 840 00 Bratislava 4, Slovak Republic
and
Department of Adaptive Systems
Institute of Information Theory and Automation
Academy of Sciences of the Czech Republic
E-mail: hruz@ifi.savba.sk

ABSTRACT

The main goal of this paper is to study the regularization problems which occur in predictive control of systems with severely ill-posed inverse model. We propose a new formulation of predictive control where the future controls are obtained as a solution of certain ill-posed inverse problem. This formulation leads to a possibility of closed loop on-line control of complex systems with ill-posed inverse. The approach is illustrated on heat conduction equation. We study the ill-posed problem arising from the above formulation by augmented Tikhonov method (Tikhonov, 1995). The computer simulation results are also provided.

NOMENCLATURE

PDE	Partial Differential Equation
MBPC	Model Based Predictive Control
GPC	Generalized Predictive Control
RBPC	Regularization Based Predictive Control

1 INTRODUCTION

Modern technological production contains lot of complex processes with distributed parameters like temperature, concentration, current or/and electro-magnetic field etc. Most of these technological processes are modeled by partial differential equations (PDE) which describe the time development of distributed quantities.

An important problem is a design or *inverse* problem where a given goal quantities are given and we are looking for process parameters which when applied to the process will lead to the a priori given goal. The quantity sought at the end of the process can be temperature, shape, specific micro-structure or/and other technologically useful distributed quantities. Many of these problems solved with standard methods exhibit a high numerical instability. They are so called "ill-posed" because they incorporate an inversion which lacks good properties of the original model. Among the most critical problems are the following: (a) there is no solution to the inverse problem, (b) the solution is not stable (inverse operator is not continuous). Generally, the methods developed for this class of problems consist in using certain "a priori" information to make the problem solvable or to "regularize" the problem.

On the other hand there are many other situations where ill-posed inverse problems occur like measurement, parameter identification, parameter tracking, failure detection, star spectroscopy etc. Not to mention that some theoreticians like Hadamard have considered the possibility to classify such problems early in this century.

There is a large body of literature (see for example (Tikhonov, 1995; Beck, 1985; Zabaras, 1993; Bui, 1993)) which describes how to solve the inverse problems which

[1] Research project VEGA, Algorithmic Design for Solving Some Problems of Parallel Numerics on Supercomputers, Scientific Grant Agency of Slovak republic and UTIA research grants GA CR 102/99/1564, GA AV CR A 2147701

are formulated for the above mentioned technologies. But most of the work focus on a specific situation which can be described with term *off-line control* because the final goal is given statically at the beginning and the solution is computed for the whole process from the start to the end. This approach is sufficient (or even necessary) for many processes, however there are situations when a persistent stabilization or adaptation of technology is needed. This can be called *on-line control* because the algorithm must persistently read the process data and set the parameters so that the distributed quantities are maintained in certain (could be also time varying) bounds.

Let us mention a recent technological example. Galvannealing is a new technology for zinc coated steel sheets, where the galvanized steel is thermally treated to convert the zinc coating to a coating of iron-zinc intermetallics through diffusional reactions (Gouel, 1995). Steel sheets appropriately processed through galvannealing have much better features for further processing as well as much higher resistance to corrosion. In the technology, the steel sheets are guided through the liquid zinc pot and then the zinc layer is solidified. The galvannealing, which comes after, consists in controlled re-heating and cooling of steel sheets so that a certain intermetallic controllably grows on the steel surface. The growth of the intermetallics on the steel surface is modeled by a system of diffusion-reaction PDE equations (Chakkingal, 1995). The control task is to on-line manipulate heating and spray cooling so that the specific intermetallic layer will prevail on the surface. Moreover, the process is very fast and the feedback data comes after 50-60 seconds. Considering the speed (3-4 m/s), any mistake leads to a destruction of hundreds of meters of the material.

This is a typical example where a recursive on-line algorithm is needed and the process model is known to have ill-posed inverse.

Our goal is to propose an approach for on-line control of processes for which the inverse problem (or model) is ill-posed. We inspire our development with a method from control theory called MBPC - Model Based Predictive Control. However, MBPC was originally developed for control of SISO (Simple Input - Simple Output) systems described by ordinary differential resp. difference equations and therefore it needs a synthesis with regularization methods to be usable also for distributed parameter systems control.

The paper is organized as follows: in the preliminary section we briefly recall the MBPC basic ideas. Then we describe our motivation and the model system on which the approach is illustrated. The main section describes our contribution which is the recursive on-line control algorithm containing Tikhonov regularization as an internal step. We conclude with simulation experiments and some notes to further research.

2 PRELIMINARIES

2.1 Model Based Predictive Control

In this section we briefly mention basic ideas and building blocks of model based predictive control. There are more approaches which belong to this rather general class of predictive control. We focus on Generalized Predictive Control (GPC) (Clarke, 1987) as one of the latest developments in this field.

The model based predictive control is based on the following set of ideas.

- The *process model* is used to derive the future process behavior as a function of past inputs/outputs and as a function of hypothetical future *controls*. There is (usually finite) time domain window over which the predictions are made.
- A cost function is defined which measures the tracking error between the future system outputs as a function of future controls and the reference signal. This cost is measured over certain time domain which is a subdomain of the prediction window.
- The cost function is optimized with respect to hypothetical future control. The optimization provides a vector of optimal future controls leading to minimal tracking error.
- The control loop is closed using a so called *receding horizon* strategy where only the first element of the above vector of optimal controls is transmitted to the plant and the whole processing window is moved 1 step ahead.

GPC in particular, uses a Controlled Auto-Regressive and Integrated Moving-Average (CARIMA) process model in the following form:

$$A(q^{-1})y(t) = B(q^{-1})u(t-1) + \frac{C(q^{-1})}{\Delta}\epsilon(t). \qquad (1)$$

In this equation $y(t)$ is the system output, $u(t)$ is the system control input and $\epsilon(t)$ is an uncorrelated random sequence. Δ is the difference operator $(1-q^{-1})$ i.e. $\Delta z = z(t)-z(t-1)$ and A, B, C are polynomials in the backward shift operator q^{-1}. (A, B) represents the plant dynamics and (A, C) the disturbance.

The above model is used in GPC to provide a *long-range prediction* strategy where the future outputs of the model are predicted up to a *prediction horizon.* As inputs to the prediction operator the past and present outputs $(y(t-i), i \geq 0)$ and the past controls $(u(t-i), i < 0)$ are used. A distinctive feature of GPC is that the predictions are expressed as a function of future control increments $\Delta u(t+i), i \geq 0$.

According to (Clarke, 1987) the predicted output can be decomposed to two terms. First is so called free response of the system

$$y_1(t+i) = y(t) + F_i(q^{-1})\Delta y(t) + G_1(q^{-1})\Delta u(t-1). \quad (2)$$

where the future controls $u(i), i \geq t$ equal $u(t-1)$ i.e. the future control increments are zero. y_1 is only a function of current output $y(t)$ and past outputs and controls. F_i, G_1 are polynomials in q^{-1}.

The second element of the output decomposition

$$y_2(t+i) = G_2(q^{-1})\Delta u(t+i-1) \quad (3)$$

is dependent on future control increments $\Delta u(t+i), i \geq 0$. G_2 is a polynomial in q^{-1}.

The future system response for the whole prediction interval $i = 1,..,N$ can be written in a matrix form as

$$\mathbf{y} = \mathbf{G}\mathbf{u} + \mathbf{p} + \epsilon \quad (4)$$

where $\mathbf{y}$ is the vector of future outputs $y(t+i), i > 0$, $\mathbf{u}$ is the vector of future control increments, $\mathbf{p}$ is the vector of predictions according to the free response $y_1(t+i)$ and ϵ is the vector of error due to future noise terms.

If we denote as $w(t+i), i > 0$ the reference signal then the control algorithm can be synthesized from the following quadratic cost function with constraints

$$J(N1, N2, NU, \lambda) = \sum_{i=N1}^{N2} e^2(t+i) + \lambda \sum_{i=1}^{NU} \Delta u^2(t+i-1) \quad (5)$$

$$\Delta u(t+i) = 0, i \geq NU \quad (6)$$

where $\mathbf{e} = (w(t+1) - y(t+1), ..., w(t+N) - y(t+N))$ is a vector of tracking errors, and $N1, N2, NU$ and λ are the tuning parameters of GPC, namely: N1 - the minimum costing horizon, N2 - the maximum costing horizon, NU - the control horizon, λ - the weighting factor.

For a deterministic plant with $\epsilon(t+i) = 0$ the prediction equation can be substituted into the cost function. We obtain the following normal equations for the minimizer of future controls:

$$\mathbf{u} = (\mathbf{G^T G} + \lambda\mathbf{I})^{-1}\mathbf{G^T}(\mathbf{w} - \mathbf{p}) \quad (7)$$

2.2 The basic model and motivation

To maintain a certain simplicity we describe our approach on 1-dimensional diffusion model. However, the approach presented here can be straightforwardly generalized to more spatial dimensions and different boundary conditions. With the word "straightforward" we mean that the basic conceptual framework would be the same. On the other hand this step is by no means trivial because it involves replacing all the basic blocks of our 1-dimensional procedure with the 2/3-dimensional blocks with the same functionality.

We develop a different point of view to predictive control where we interpret the minimization of (5) as a tool for solving an ill-posed operator equation

$$Au = w. \quad (8)$$

The different point of view is in the fact that we can consider the control algorithm synthesis as a solution of ill-posed problem (8). Then the *regularization* (ill-posed problem solving) methods give us an algorithm for solution of (8) based on the minimization of the following functional:

$$\alpha(u) = \| Au - w \|^2_{\|X\|, D_1} + \alpha \| u \|^2_{\|Y\|, D_2} \quad (9)$$

where $\| X \|, D_1$ and $\| Y \|, D_2$ denote different norms and discretizations. This looks to be the same as (7) but now the parameter α plays a crucial role and we have a deep theory (Tikhonov, 1990) how to choose and interpret this so called regularization parameter.

Moreover, we have an interpretation of tuning knobs as being in fact regularization parameters also for the classical predictive control case. Particularly the possibility to interpret the appropriate value of λ seems to be interesting as far as the MBPC framework do not contain a theory how to set up the value of this so called weighting factor without which the procedure is not usable in practise because of numerical instability.

As a model system we use the following basic diffusion equation

$$\frac{\partial}{\partial t}Y(x,t) - \frac{\partial}{\partial x}\left(a^2(Y)\frac{\partial}{\partial x}Y(x,t)\right) + b(Y)Y(x,t) = 0$$

$$Y(x,t_0) = Y_0(x), Y(0,t) = u(t), \frac{\partial Y(L,t)}{\partial x} = 0$$

$$0 \leq x \leq L, \quad t \geq t_0, \quad a \neq 0$$

$$a^2 = \frac{\lambda}{c.\rho}, \quad b = \frac{h}{c.\rho} \quad (10)$$

where L is the length of the bar in meters; λ is thermal conductivity coefficient; a^2 is thermal diffusivity coefficient in m^2/s; c is the specific heat in $\frac{J}{kg\ K}$; ρ is specific mass of the bar in kg/m^3 and h is the heat-transfer coefficient in

$\frac{W}{m^2\ K}$. In the remaining text $Y(x,t)$ denotes the deviation of temperature from the surrounding fluid in K.

As the notation suggests it is a heat conduction/heat transfer 1-Dimensional equation which describes heat conduction in a metal bar from the heated end at the spatial coordinate 0 to the isolated end at the spatial coordinate L with a thermal diffusivity coefficient a^2. The equation describes also a heat transfer to the surrounding air expressed by term $bY(x,t)$ where b contains the heat transfer coefficient. The equation is nonlinear and we suppose to handle the nonlinearity by "freezing" the values of coefficients a, b during the prediction horizon.

The control task is to drive the temperature along the spatial coordinate to the prescribed reference sequence of temperature profiles with a boundary heating represented by the function $Y(0,t) = u(t)$.

The physical model described by the above equation consists in a metal bar with a boundary heater on one side and an isolation on the other side.

3 REGULARIZATION BASED PREDICTIVE CONTROL

To build a predictive control we must derive a predictor which describes the system state in future times as a function of hypothetical future control. For the derivation of predictor we use finite difference Crank-Nicolson scheme which for the case of system (10) reads

$$p(x,t) = a(Y(x,t)),\ q(x,t) = b(Y(x,t)),\ \omega = \frac{\tau}{2h^2} \tag{11}$$

$$\begin{aligned} \mathbf{A}_U^{(l)}\mathbf{Y}^{(l)} &= \mathbf{A}_L^{(l)}\mathbf{Y}^{(l-1)} + \omega(\mathbf{f}^{(l)} + \mathbf{f}^{(l-1)}) \\ \mathbf{A}_U^{(l)} &= \mathbf{C} + \omega\mathbf{M}^{(l)}, \\ \mathbf{A}_L^{(l)} &= \mathbf{C} - \omega\mathbf{M}^{(l-1)}, \end{aligned} \tag{12}$$

$$\begin{aligned} k = 1,...,n-1,\ \ C_{kk} &= 1 \\ C_{nn} &= \frac{1}{2} \end{aligned} \tag{13}$$

$$\begin{aligned} k = 1,...,n-1,\ \ M_{kk}^l &= p(x_k - \frac{h}{2}, t_l) + p(x_k + \frac{h}{2}, t_l) + h^2 q(x_k, t_l) \\ M_{k,k+1}^l &= M_{k+1,k}^l = -p(x_k + \frac{h}{2}, t_l) \\ M_{nn} &= p(x_k - \frac{h}{2}, t_l) + \frac{1}{2}h^2 q(x_k, t_l) \end{aligned} \tag{14}$$

$$\begin{aligned} \mathbf{E}^{(l)} &= (\mathbf{A}_U^{(l)})^{-1}\mathbf{A}_L^{(l)}, \\ {}^{\frac{1}{2}}\mathbf{A}^{(l)} &= (\mathbf{A}_U^{(l)})^{-1} \end{aligned}$$

$$f_1^l = p(x_0 + \frac{h}{2}, t_l)u(t_l) \tag{15}$$

$$k = 2,...,n,\ \ f_k^l = 0$$

$$\mathbf{v}^l = \omega\mathbf{f}^{(l)}$$

where τ, h are the discretization steps in time and space.

In our case of boundary control only the first element of the vector $\mathbf{f}^{(l)}$ is not zero.

3.1 Predictor

We start this section with discussion of what we call a *local predictor*. It is summarized in Theorem 1 below as a predictive scheme which predicts the state k-steps ahead for the whole spatial domain. Then, at the end of the section, the *predictor* is defined as a set of local predictors with an option not to consider all spatial points of certain local predictors.

In the following, l denotes a discrete time starting from 0. Let k is a fixed number $k > l$. The matrices considered have dimension $n \times n$ or $n \times k$ (depending on a context) where n is the discretization density in space of the finite difference scheme and k determines the number of steps ahead to which the predictor computes the future state.

The state $\mathbf{y}^{l+k}$ can be expressed according to the following theorem

Theorem 1 (RBPC Local Predictor). *Let a finite difference scheme (11)-(15) with n discretization steps is given and let l denotes the discrete time starting at $l = 0$. Then a k-step ahead incremental predictor $\mathcal{P}^{k,l}$ assigned to this scheme equals*

$$\begin{aligned} \mathbf{y}^{l+k} = \mathcal{P}^{k,l}(\mathbf{y}^l, \Delta\mathbf{u}^{k,l}) = \mathbf{T}^{(k,l)}\mathbf{y}^l + 2\omega p_0\mathbf{F}^{(k,l)}\mathbf{u_1}^{k,l} + \\ +\omega p_0\mathbf{F}^{(k,l)}\mathbf{B}\Delta\mathbf{u}^{k,l}. \end{aligned} \tag{16}$$

$\mathbf{T}^{(k,l)}$ and $\mathbf{F}^{(k,l)}$ are $n \times n$ resp. $n \times k$ matrices defined by the following matrix recurrences

$$\begin{aligned} \mathbf{T}^{(k,l+1)} &= \mathbf{E}^{(l+k+1)}\mathbf{T}^{(k,l)}(\mathbf{E}^{(l+1)})^{-1} \\ \mathbf{T}^{(k,0)} &= \mathbf{E}^{(k)}\ldots\mathbf{E}^{(2)}\mathbf{E}^{(1)} \end{aligned} \tag{17}$$

$$\begin{aligned} \mathbf{F}^{(k,l+1)} &= \mathbf{S}_{-1}(\mathbf{E}^{(l+k+1)}\mathbf{F}^{(k,l)},\ {}^{\frac{1}{2}}\mathbf{A}_{\bullet 1}^{(l+k+1)}) \\ \mathbf{F}^{(k,0)} &= (\mathbf{E}^{(k)}\ldots\mathbf{E}^{(2)}\ {}^{\frac{1}{2}}\mathbf{A}^{(1)})_{\bullet 1} \mid \ldots \mid (\ {}^{\frac{1}{2}}\mathbf{A}^{(k)})_{\bullet 1}) \end{aligned} \tag{18}$$

where $\mathbf{S}_{-1}$ is a shift operator defined as

$$\mathbf{S}_{-1}(\mathbf{X}, \mathbf{z}) = (\mathbf{X}_{\bullet 2}, \mathbf{X}_{\bullet 3}, \ldots, \mathbf{X}_{\bullet k}, \mathbf{z}).$$

Vectors $\mathbf{u_1}^{k,l}$, $\Delta\mathbf{u}^{k,l}$ *and matrix* $\mathbf{B}$ *are defined as*

$$\mathbf{u_1}^{k,l} = (u^l, \ldots, u^l)^T,\ u^l = u(t_l)$$
$$\Delta\mathbf{u}^{k,l} = (\Delta u^1, \ldots, \Delta u^k)^T,\ \Delta u^i = u(t_{l+i}) - u(t_{l+i-1})$$
$$\mathbf{B} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & \ldots & 0 \\ & \ldots\ldots & & \\ 2 & \ldots & 1 & 0 \\ 2 & 2 & 2 & 1 \end{pmatrix}.$$

$\omega = \frac{\tau}{2h^2}, p_0 = p(x_0 + h/2, t_l)$ *and* τ, h *are the discretization steps in time resp. in space.*

Some of the complexities in the derivation of the predictor are related to the nonlinearity in the model (10). In the case when the thermal diffusivity coefficient a and coefficient b are constant the matrices $\mathbf{T}^{(k,l)}$ and $\mathbf{F}^{(k,l)}$ are constant which leads to a faster algorithm.

The pre-multiplication with ωp_0 in the theorem is important because it allows to use more local predictors with different space grids located at a different future times. The unknown variable of control increments is made independent of the particular grid density of a local predictor.

Another aspect of the theorem is that it serves well in organizing the structure of the predictor but it can not be directly used as an algorithm. The problem is in the matrix recursion (17). The characteristic high stability features of Crank-Nicolson scheme stems from the fact that the transition matrix $\mathbf{E}^{(k)}$ has all eigenvalues less than 1. That means that the matrix $(\mathbf{E}^{(l+1)})^{-1}$ which is on the right hand side of the recursion (17) has all eigenvalues larger than 1 which makes the matrix recursion totally unstable. However, we are not in fact interested in the recursion itself but in the vector $\mathbf{T}^{(k,l)}\mathbf{y}^l$ which can be computed by the simplified scheme $\mathbf{A}_U^{(l)}\mathbf{y}^{(l)} = \mathbf{A}_L^{(l)}\mathbf{y}^{(l-1)}$ because the vector $\mathbf{T}^{(k,l)}\mathbf{y}^l$ corresponds exactly to the solution of the system with zero boundary conditions. Moreover, this is in fact also much more computationally efficient as far as both matrices in the recursion are dense to the contrary to standard Crank-Nicolson step which involves only tridiagonal matrices. Therefore the computational complexity of computing the matrix recurrence would be $O(n^3)$ whereas the complexity of computing the vector $\mathbf{T}^{(k,l)}\mathbf{y}^l$ by k simplified steps as above is only $O(kn)$.

On the other hand the second recurrence (18) is stable as far as it involves only the stable matrix $\mathbf{E}^{(k)}$.

The local predictor from the above theorem can be used for constructing the generalized version of a predictor where more predictions to the various depth in time and space are made. This is necessary to gain a control over the spatial domain behavior of the system.

In general, the predictor is constructed taking a set of local predictors $\mathcal{P}^{k,l}(\mathbf{y}^l, \Delta\mathbf{u}^{k,l})$ all starting from the same state but having different time k of the prediction as well as different discretization grid. Moreover, from each predictor only a certain subset of points can be chosen.

When we assume linearity (the coefficients a, b thermally independent) the equation (10) is known to have a unique solution for the unknown control $u(t)$. From this point of view it might seem that the concept of constructing predictor from more local predictors is contradictory. But the regularization method handles also a case when the ill-posed problem has no solution. In that case we are looking for a nearest possible solution in some norm. So we can make a trade off compromising the uniqueness but solving another important problem which we have in predictive control. It is that if we do not incorporate more predictions in time and space then the algorithm leaves the current reference point earlier than needed as can be seen from comparison of figures 1 and 2. The reason is that when the predictor will see the new reference point on the horizon it will immediately start to leave the current reference point as the simulation experiments show.

3.2 Regularized solutions and control law

In order to formulate a control law for RBPC we recall some notions from Tikhonov theory of ill-posed problems. We follow (Tikhonov, 1990; Tikhonov, 1995) in defining the basic notions.

For operator equation $Az = u$ resp. for its perturbation $A_h z = u_\delta$ we can define a smoothing functional $M^\alpha(z) = \| A_h z - u_\delta \|^2 + \alpha \| z \|^2$ where α is a *regularization parameter.*

Definition 2 (Generalized residuum). *The real function* $\rho_{(h,\delta)}(\alpha) = \| A_h z^\alpha - u_\delta \|^2 - (\delta + h \| z^\alpha \|)^2 - \mu^2(u_\delta, A_h)$ *is called* generalized residuum *where* $\mu^2(u_\delta, A_h) = \inf_{u \in D} \| A_h z - u_\delta \|$ *is a* degree of inconsistency *and* z^α *is a minimizer of the smoothing functional* $M^\alpha(z)$.

The *regularized solution* is given by the algorithmic *Principle of generalized residuum.*

Definition 3 (Principle of generalized residuum). *If* $\| u_\delta \|^2 \leq \delta^2 + \mu^2(u_\delta, A_h)$ *the regularized solution* z^α *is defined to be identically zero. Otherwise the regularized solution is given as* $z^{\alpha^\star}$ *where* $\alpha^\star$ *is the root of generalized residuum* $\rho(\alpha^\star) = 0$.

An important design aspect of particular regularization method which follows the general theory mentioned above is the selection of norms (resp. scalar products) in the functional spaces Z, U to which z, u belong. Because an inversion of the parabolic equation (10) is severely ill-posed problem, we choose a strongly dumping norm in the space of controls, namely the Sobolev W_2^1 norm given as $\int_0^T z^2(\tau) + \frac{dz}{d\tau}^2(\tau)\, d\tau$. The distance on the right hand side (deviation from the reference signal) is expressed with L_2 norm.

The authors in (Tikhonov, 1990) show that under rather general assumptions the discretization of the above method can be equivalently obtained through different procedures: (i) either first the Euler (normal) equation of smoothing functional is obtained in functional space and then discretized or (ii) the smoothing functional is first discretized and then Euler equations are computed in discrete space. As far as we derived the predictor in discrete sense we use the letter method.

Using the above defined notions we can state the control law for Regularization Based Predictive Control as:

Control Law 4 (Regularization Based Predictive Control). *Let the system is described by parabolic heat equation as in (10). A is an evolution operator given by the equation (10) and $u(t)$ is a boundary control signal. Let the perturbed operator A_h is given by the predictor $\mathbf{P}^l = (\mathcal{P}^{k_1,l}_{g_1,d_1}, \ldots, \mathcal{P}^{k_i,l}_{g_i,d_i}, \ldots, \mathcal{P}^{k_m,l}_{g_m,d_m})$ consisting in a set of local predictors as defined in section 3.1. Then control is given as the first $n \leq k_h$ elements of the* regularized solution *of an operator equation $\mathbf{P}^l u = w$ over a finite discretization horizon k_h where w is a reference signal known over the prediction horizon.*

The interpretation of the perturbation parameters (h, δ) must be discussed at this point. The parameter h is clearly interpreted as a discretization error with which the discrete version of the predictor approximates the evolution operator given by (10). In different words, it is the discretization error of the finite difference Crank-Nicolson method. This equals to $O(\tau^2 + s^2)$ where τ, s are discretization steps in time resp. in space. In practice the constant in the $O(\tau^2 + s^2)$ can be easily established by few experiments.

The parameter δ defines in fact a neighborhood around the right hand side of the governing equation. Concerning this, it is well reasonable to set its value to the standard deviation of data noise. In fact the Tikhonov regularization method trades smoothness against the right hand side fit.

Now we can define the control algorithm as follows:

Control Algorithm 5 (Regularization Based Predictive Control). *Applying the Control law 4 we have the following control algorithm.*

1. *(*Prediction*) Construct the predictor from the local predictors as described in section 3.1.*
2. *(*Inversion*) Find the regularized solution of the equation*

$$\mathbf{P}^l(\Delta u) = w$$

.

3. *(*Control*) Apply the first $n \leq k_h$ controls to the plant.*
4. *(*Receding horizon*) $l = l + n$*
5. *(*Close loop*) Repeat from the Point 1.*

Because the Point 1. of the above algorithm has been discussed in section 3.1 we focus now on the Point 2. i.e. how to efficiently find the regularized solution.

To simplify the writing of formulas we denote Δu as q and we drop the superscript on the predictor $\mathbf{P}$. The Euler equations for discrete form of smoothing functional can be written taking into the account the definition of the L_2 and W_2^1. The discrete form of W_2^1 can be obtained by an approximation of the integral $\int_0^T z^2(\tau) + \frac{dz}{d\tau}^2(\tau)\, d\tau$ by $\sum_{i=1}^{k_h} q_i^2 \tau + \sum_{i=2}^{k_h} \frac{(q_i - q_{i-1})^2}{\tau}$. The discrete version of the normal equation reads

$$\mathbf{P}^T \mathbf{P} q + \alpha \mathbf{G} q = \mathbf{P}^T w.$$

The tridiagonal matrix $\mathbf{G}$ has a form

$$\mathbf{G} = \begin{pmatrix} 1 + \frac{1}{\tau^2} & -\frac{1}{\tau^2} & 0 & 0 \\ -\frac{1}{\tau^2} & 1 + \frac{2}{\tau^2} & \ldots & 0 \\ & \ldots & \ldots & \\ & \ldots & 1 + \frac{2}{\tau^2} & -\frac{1}{\tau^2} \\ & & -\frac{1}{\tau^2} & 1 + \frac{1}{\tau^2} \end{pmatrix} \qquad (19)$$

Now we follow (Tikhonov, 1990) in Choleski decomposition of $\mathbf{G}$ to $\mathbf{G} = \mathbf{H}^T \mathbf{H}$. This gives

$$\mathbf{P}^T \mathbf{P} q + \alpha \mathbf{H}^T \mathbf{H} q = \mathbf{P}^T w.$$

Multiplying by $\mathbf{H^{-1}}$ we obtain

$$(\mathbf{P}\mathbf{H^{-1}})^T \mathbf{P}\mathbf{H^{-1}} (\mathbf{H} q) + \alpha (\mathbf{H} q) = (\mathbf{P}\mathbf{H^{-1}})^T w.$$

Introducing a change of variables $y = \mathbf{H} q$ we have the following basic form of the equation

$$\mathbf{J}^T \mathbf{J} y + \alpha y = \mathbf{J}^T w$$

where $\mathbf{J} = \mathbf{PH^{-1}}$. Now in (Tikhonov, 1990) a Householder transform is used to produce a tridiagonal matrices. We proceed differently; the matrix $\mathbf{J}$ is decomposed using singular value decomposition (SVD) to the form $\mathbf{J} = \mathbf{LDR}$ where $\mathbf{L}$ resp. $\mathbf{R}$ are orthogonal transforms ($\mathbf{R^{-1}} = \mathbf{R}^T$) and $\mathbf{D}$ is a diagonal matrix of singular values (in fact the eigenvalues of $\sqrt{\mathbf{J}^T\mathbf{J}}$).

We can write

$$\mathbf{R}^T\mathbf{D}^T\mathbf{D}\mathbf{R}y + \mathbf{R}^T(\alpha\mathbf{R}y) = \mathbf{R}^T\mathbf{D}^T\mathbf{L}^T w.$$

Multiplying with $\mathbf{R}$ from left and introducing a new variables $\mathbf{R}y = x$ we obtain

$$(\mathbf{D}^T\mathbf{D} + \alpha\mathbf{I})x = \mathbf{D}^T\mathbf{L}^T w.$$

Now the solution can be explicitly written as

$$x = \frac{\mathbf{D}^T}{\mathbf{D}^T\mathbf{D} + \alpha\mathbf{I}}\mathbf{L}^T w \qquad (20)$$

where $\frac{\mathbf{D}^T}{\mathbf{D}^T\mathbf{D}+\alpha\mathbf{I}}$ is a symbolic denotation for a diagonal matrix with $\frac{D_{ii}}{D_{ii}^2+\alpha}$ on diagonal for $i \leq min(d_r, d_c)$; the rest of the matrix is zero. d_r is the dimension of reference signal w and d_c is the dimension of control increments. The dimension of the matrix is $d_c \times d_r$. As far as $\mathbf{L}^T$ is always a regular matrix (a rotator) the reference signal is rotated and multiplied by the matrix $\frac{\mathbf{D}^T}{\mathbf{D}^T\mathbf{D}+\alpha\mathbf{I}}$. Therefore we can consider two cases (except the regular case $d_r = d_c$): (i) $d_r < d_c$ and (ii) $d_r > d_c$. The former case means introducing redundancy in control whereas the latter case means introducing a rank deficiency (uncontrollability). This leads to the conclusion that the default setting for the control algorithm should have the same dimension of controls as of the reference signal.

The regularization itself is achieved by the function

$$f(x, \alpha) = \frac{x}{x^2 + \alpha} = \frac{1}{x + \frac{\alpha}{x}}$$

which acts on the diagonal of the matrix $\mathbf{D}$. It is a bounded approximation of the function $\frac{1}{x}$.

The next step is the computation of the generalized residuum because α is supposed to be its root. It would be advantageous if we would not need to go back to the original coordinates while computing the function $\rho_{(h,\delta)}(\alpha) = \| \mathbf{P}q^\alpha - w_\delta \|^2 - (\delta + h \| q^\alpha \|)^2 - \mu^2(w_\delta, \mathbf{P})$. Indeed, it is possible to use the system in the canonic form. To see this, we write

$$\| \mathbf{J}y^\alpha - w_\delta \| = \| \mathbf{PH^{-1}H}u - w \| = \| \mathbf{P}u - w \| .$$

On the other hand

$$\begin{aligned} \| \mathbf{J}y^\alpha - w_\delta \| &= \| \mathbf{LDR}y^\alpha - w \| = \| \mathbf{LD}x - \mathbf{LL}^T w \| = \\ &= \| \mathbf{L}(\mathbf{D}x - \mathbf{L}^T w) \| = \| \mathbf{D}x - \mathbf{L}^T w \| \end{aligned} \qquad (21)$$

because $\mathbf{L}$ is an isometric operator (rotator).

But the solution is expressed explicitly in the canonic coordinates, therefore the final expression for residuum reads

$$\| \mathbf{P}q^\alpha - w_\delta \| = \| \frac{\alpha}{\mathbf{D}^T\mathbf{D} + \alpha\mathbf{I}}\mathbf{L}^T w \| .$$

The degree of inconsistency $\mu^2(w_\delta, \mathbf{P})$ is not computed at all. An inspection of the formula (20) for the solution of the normal equation as well as the formula for residuum shows that the degree of inconsistency being nonzero can be caused by two reasons. Either the rank deficiency occurs because there are not enough controls from the beginning or the dimension of the control is larger or equal than the dimension of reference signal but in this case the system itself is not controllable. In all cases the rank deficiency means zeros on the diagonal of matrix $\mathbf{D}$ therefore these elements of vector $\mathbf{L}^T w$ are not present in the residuum value. On the other hand they constitute exactly the value of the degree of inconsistency $\mu^2(w_\delta, \mathbf{P})$. Therefore instead computing residuum minus the degree of inconsistency, a quantity which we call *consistent residuum* is computed. The computation of consistent residuum is simple because it means only to add the values of residuum where the diagonal of $\mathbf{D}$ is nonzero.

Finally, the quantity $(\delta + h \| q^\alpha \|)$ is computed according to the W_2^1 approximation mentioned above. Here note that

$$\begin{aligned} \| q \| &= \| \mathbf{H^{-1}}y \| = \| \mathbf{H^{-1}R}^T x \| \leq \| \mathbf{H^{-1}} \| \, \| \mathbf{R}^T x \| = \\ &= \| \mathbf{H^{-1}} \| \, \| x \| \end{aligned} \qquad (22)$$

because matrix $\mathbf{R}^T$ is an isometric operator. The matrix $\mathbf{H^{-1}}$ is upper triangular therefore its operator norm can be easily computed by computing the W_2^1 norm of its last column.

The discussion above leads to the conclusion that for the solving of the normal equations as well as for the evaluation of generalized residuum we need only the diagonal matrix $\mathbf{D}$ and the vector $r = \mathbf{L}^T w$. If $\mathbf{D}$ is stored as a vector, then the algorithm which computes the root of generalized residuum, as well as the final solution, can work using only these two vectors (and scalars like error parameters h, δ) which leads to a very efficient implementation.

Even if the method is derived for the case of thermally dependent material properties only the numerical examples with constant material properties are considered in the simulations section.

3.3 Simulations and conclusions

The control algorithm 5 is illustrated on a process with load disturbance randomly changing the set point. The control goal is to reject the load disturbance whereas the set point is changed four times. The states of the system corresponding to four set points have the boundary temperature 100, 50, 200 and 70 degrees Celsius as it is shown in the figure.

A conclusion from the simulations is that the algorithm designer can choose between different responses considering different predictor structures. The predictor structure developed in the previous sections seems to be enough flexible to capture different needs with respect to the spatial and time distribution of tracking error.

In Figure 3 the time development of the regularization parameter α is shown for the same simulation as is in Figure 1. α shows a characteristic pattern where during the transient states when the system is enough excited the value of α is very small which means that only a very small level of smoothing (regularization) is needed. When the system reaches the stable state the value of α grows significantly because the information coming from the right hand side of the equation approaches zero and the inversion needs much higher stabilizing factor. Finally, if there is no further disturbance the condition $\| u_\delta \|^2 \leq \delta^2 + \mu^2(u_\delta, A_h)$ from the Principle of generalized residuum is fulfilled leading to the zero value of control increments until a new load disturbance or noise will not provide enough large (in L_2 norm) right hand side for the inversion task.

In the same figure, bottom picture, the number of generalized residuum evaluations needed for computation of the parameter α is shown as a function of time during the same simulation. As was mentioned in the previous section as far as the searching for root of generalized residuum can be completely done in canonic coordinates we prefer the simplest (but very robust) root searching method - namely, dividing the interval to halves. When we experimented with regula falsi and variations of Quasi-Newton method less evaluations have been typically needed but during the transient states they were divergent or too many iterations occurred.

Another aspect is related to the fact that incorporating the regularization into the heard of predictive algorithm we have a strong stabilizing factor. Therefore, we can try to stabilize the system in the *neighborhood* of unstabilizable state.

Let us consider a constant value reference signal along the spatial domain as an example for the system with a nonzero film coefficient $b \neq 0$. Denote this state $\mathbf{C}$. An inspection of the modal system of ordinary differential equations shows that there is no constant value of a boundary condition which can stabilize the system at this state. However, we can experiment with RBPC algorithm to find a *stable periodic orbit* containing the desired state in its interior. Interior is meant in a special sense that it is a domain defined as a union of intervals obtained taking a maximal and a minimal temperature at certain spatial point during the whole orbit period of the system.

We suggest that such orbit must also contain a stable state of the system which has the same minimal temperature at the point $x = L$ as is the constant temperature of the desired state $\mathbf{C}$. It intuitively follows from the maximality principle for parabolic equation.

If we take for example the state $\mathbf{C}$ with constant temperature 100 degrees, then for the values $a = 0.0095, b = 0.0020$ we obtain that the above orbit must also contain a stable state of the system with temperature 100 degrees at the point $x = L$. Denote this state $\mathbf{S}$. It has a temperature 230 degrees at the boundary $x = 0$. Therefore we have that any stable orbit containing the state $\mathbf{C}$ must also contain the state $\mathbf{S}$.

The situation is illustrated in Figure 4 where the orbit computed by the RBPC algorithm is shown in L_2 norm together with the corresponding controls. The algorithm provides almost the best possible orbit as far as the minimal temperature and maximal temperature on the boundary are few percent below and above the best possible values according the considerations in the preceding paragraph.

It must be stressed that to achieve the behavior in Figure 4 we must apply all controls computed for the prediction horizon k_h. That means in fact that the advantage of the receding horizon is lost and the algorithm is "blind" to disturbances during the whole prediction horizon time until the new sequence is computed at the beginning of a new horizon (this gives also a period of the orbit).

It seems that a sensitivity to disturbances can be gained back by a different approach based on the fact that a system on a stable orbit must see its *past states* on the horizon. Therefore, we can speculate about an algorithm which (i) uses a predictor to compute the sequence of controls which is then (ii) successively applied but (iii) at each step the sequence itself is *verified* by including past states as reference signals in the future and recomputing the sequence. If there is no disturbance and the system is on the stable orbit we must obtain the same sequence. Otherwise the current sequence is invalidated and a new sequence computed. However, to put this note on a solid ground a further research would be necessary.

GPC was proved to be one of the best control methods discovered so far. We think that the major reason is that GPC in contrary to the classical methods is working with *long-range prediction*. But for a system with reaction/diffusion PDE as a model, long-range prediction is inevitably leading to severely ill-posed problems.

There are different approaches for regularization which might be also used in our framework. One which we consider in particular is iterative regularization with adjoint operator approach (Jarny, 1991). This method is rather computationally intensive but the iterative character of the algorithm can be efficiently used in on-line control. The reason is that the current controls can serve as a good approximation for future solution. However, as far as the time window is moving it will need further research to apply this idea.

The presented approach could be also combined with thermally dependent parameter identification methods to handle high temperature processes as is for example described in (Artyukhin, 1993).

REFERENCES

Artyukhin, E. A., Iterative Algorithms for Estimating Temperature-Dependent Thermophysical Characteristics, *Proc. First International Conference on Inverse Problems in Engineering*, June 1993, in "Inverse Problems in Engineering: Theory and Practice" Zabaras, Woodbury, Raynaud, editors, ASME Book No. I00357, pp. 101-108.

Bui, H. D., Tanaka, M. et al., 1994, *Inverse Problems in Engineering Mechanics*, A.A.Balkema, Rotterdam.

Beck, J. V., Blackwell, B. and St. Clair, C. R., Jr., 1985, *Inverse Heat Conduction: Ill-Posed Problems*,Wiley-Interscience, NY

Clarke, D. W., Mohtadi, C., Tuffs, P. S., 1987, *Generalised Predictive Control*, Automatica, Vol. 23, pp. 137-160.

Chakkingal, U., Wright, R. N., *Mathematical Modelling of Microstructural Evolution in Galvanized Coatings During Galvannealing*, in *GALVATECH'95 Conference Proceedings*, Stahleisen, Duesseldorf, pp. 87-105

Gouel, R. and Hugh, J., *On-line Measurement of Galvanneal Coatings*, in *GALVATECH'95 Conference Proceedings*, Stahleisen, Duesseldorf, pp. 87-105

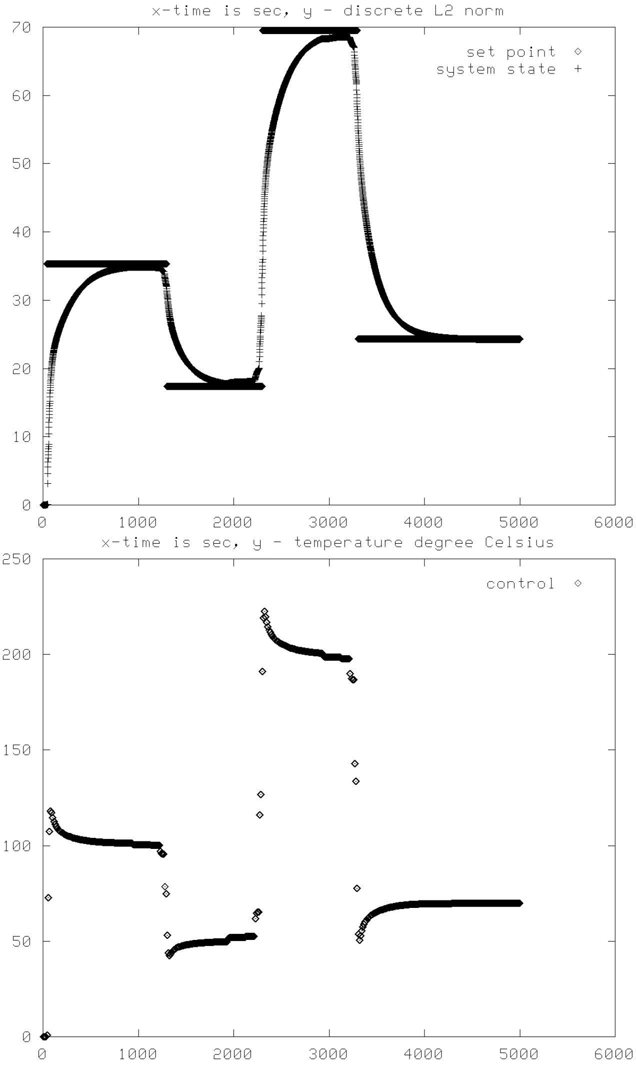

Figure 1. The control algorithm simulation with predictor structure (31.(1.16.1).(3.16.2).(7.16.4).(15.16.8).(31.16.16)). 31 is the total prediction horizon, 1,3,7,15,31 are the local predictor horizons, 16 mesh density and 1,2,4,8,16 are the depths considered for respective local predictors. The values of perturbation parameters are $h = 0.01, \delta = 0.1$. The set point and the system state are expressed in discrete L_2 norm. x-axis represents time in seconds and the y-axis value of L_2 norm. The lower picture shows the controls during the simulation.

Jarny, Y., Ozisik, M. N. and Bardon, J. P., *A General Optimization Method using Adjoint Equation for Solving Multidimensional Inverse Heat Conduction*, Int. J. Heat Mass Transfer, Vol. 34, pp. 2911-2929.

Johnson, C., 1987, *Numerical solutions of partial differ-*

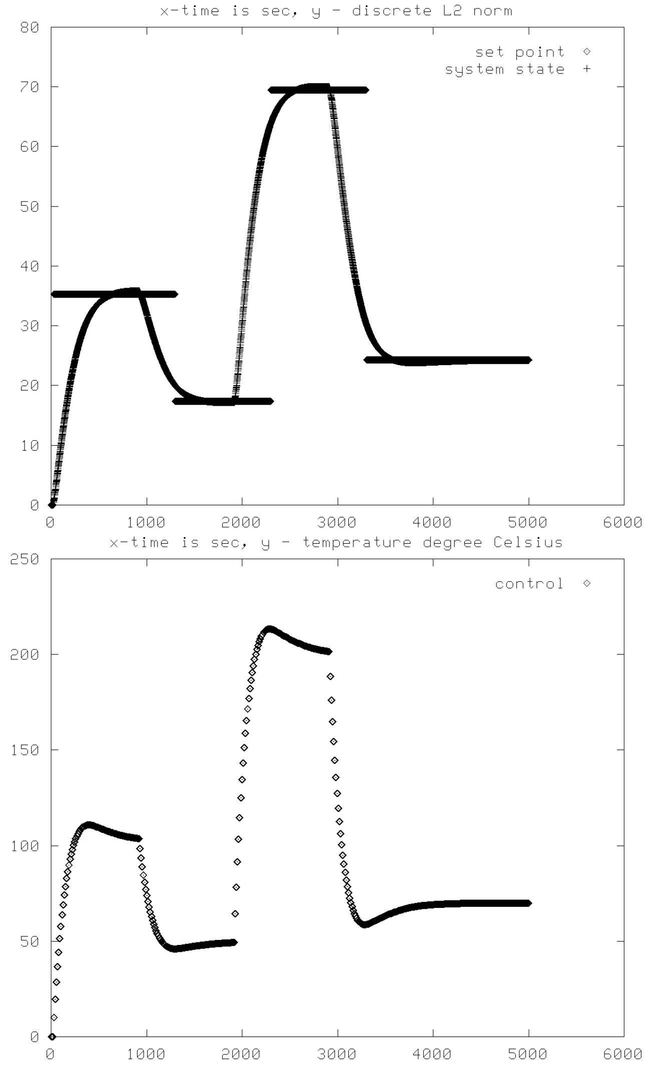

Figure 2. The control algorithm simulation with predictor structure (32.(32.32.32)). The set point and the system state are expressed in discrete L_2 norm. x-axis represents time in seconds and the y-axis value of L_2 norm. The lower picture shows the controls during the simulation.

ential equations by the finite element method, (Cambridge University Press)

Tikhonov, A. N., Goncsarskij, A. V., Stepanov, V. V. and Jagola, A. G., 1995, *Numerical Methods for the Solution of Ill-Posed Problems*, (Kluwer Academic Publisher)

Tikhonov, A. N., Goncsarskij, A. V., Stepanov, V. V. and Jagola, A. G., 1990, *Chislennyje metody reshenija nekorrektnykh zadach, [Numerical Methods for Ill-Posed Problem Solving, in Russian]*, Nauka, Moscow

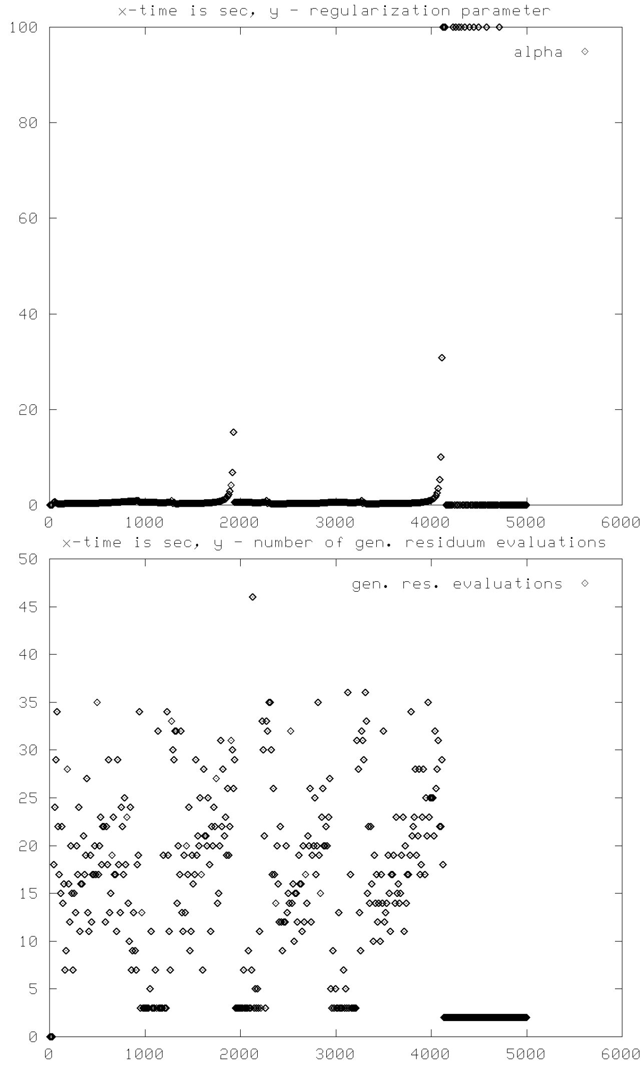

Figure 3. The time behavior of regularization parameter α during the simulation. In the lower picture, the number of generalized residuum evaluations is shown for the same simulation. For more details, see the text.

Zabaras, N., Woodbury, K. A. and Raynaud, M., 1993, *Inverse Problems in Engineering: Theory and Practice*, ASME Book No. I00357.

A APPENDIX: PROOF OF THEOREM 1

In the following, l denotes a discrete time starting from 0. Let k is a fixed number $k > l$. The matrices considered have dimension $n \times n$ or $n \times k$ (depending on a context) where n is the discretization density in space of the finite difference scheme and

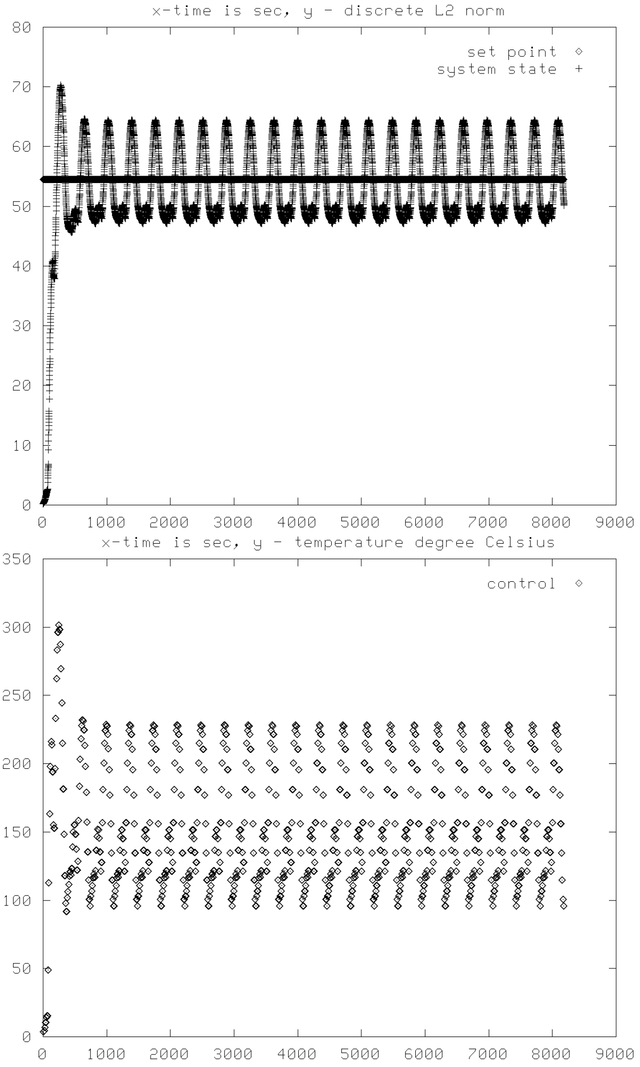

Figure 4. The stable orbit computed by RBPC algorithm.

k determines the number of steps ahead to which the predictor computes the future state.

The basic evolution equation can be written using the finite difference scheme (11)-(15) as

$$\mathbf{y}^{l} = \mathbf{E}^{(l)}\mathbf{y}^{l-1} + \tfrac{1}{2}\mathbf{A}^{(l)}(\mathbf{v}^{l} + \mathbf{v}^{l-1}). \qquad (23)$$

Then the state $\mathbf{y}^{l+k}$ can be expressed by a recursive application of equation (23) as

$$\begin{aligned} \mathbf{y}^{l+k} = \mathbf{E}^{(l+k)} \dots \mathbf{E}^{(l+2)}\mathbf{E}^{(l+1)}\mathbf{y}^{l} & + \\ +\mathbf{E}^{(l+k)} \dots \mathbf{E}^{(l+2)}\ \tfrac{1}{2}\mathbf{A}^{(l)}(\mathbf{v}^{l} + \mathbf{v}^{l+1}) & + \\ +\mathbf{E}^{(l+k)} \dots \mathbf{E}^{(l+3)}\ \tfrac{1}{2}\mathbf{A}^{(l+1)}(\mathbf{v}^{l+1} + \mathbf{v}^{l+2}) & + \\ + \dots & + \\ +\mathbf{E}^{(l+k)}\ \tfrac{1}{2}\mathbf{A}^{(l+k-1)}(\mathbf{v}^{l+k-2} + \mathbf{v}^{l+k-1}) & + \\ + \ \tfrac{1}{2}\mathbf{A}^{(l+k)}(\mathbf{v}^{l+k-1} + \mathbf{v}^{l+k}) & \end{aligned}$$

Now, we observe that the matrix $\mathbf{T}^{(k,l)} = \mathbf{E}^{(l+k)} \dots \mathbf{E}^{(l+1)}$ can be computed by the following matrix recursion:

$$\begin{aligned} \mathbf{T}^{(k,l+1)} &= \mathbf{E}^{(l+k+1)}\mathbf{T}^{(k,l)}(\mathbf{E}^{(l+1)})^{-1} \\ \mathbf{T}^{(k,0)} &= \mathbf{E}^{(k)} \dots \mathbf{E}^{(2)}\mathbf{E}^{(1)}. \end{aligned}$$

Moreover, because the vectors $\mathbf{v}^{i}$ have only the first coordinate nonzero, the terms $\mathbf{E}^{(i+k)} \dots \mathbf{E}^{(i+2)}\ \frac{1}{2}\mathbf{A}^{(i)}(\mathbf{v}^{i} + \mathbf{v}^{i+1})$ can be interpreted as the first columns of the respective matrices $(\mathbf{E}^{(i+k)} \dots \mathbf{E}^{(i+2)} \times$
$\frac{1}{2}\mathbf{A}^{(i)})_{\bullet 1}$ times a scalar $\mathbf{v}_1^{i} + \mathbf{v}_1^{i+1}$ ($X_{\bullet i}$ denotes the i-th column of matrix X).

Further we can write:

$$\begin{aligned} \mathbf{y}^{l+k} = \mathbf{T}^{(k,l)}\mathbf{y}^{l} & + \\ +(\mathbf{E}^{(l+k)} \dots \mathbf{E}^{(l+2)}\ \tfrac{1}{2}\mathbf{A}^{(l)})_{\bullet 1}(2\mathbf{v}_1^{l} + \Delta\mathbf{v}_1^{l}) & + \\ +(\mathbf{E}^{(l+k)} \dots \mathbf{E}^{(l+3)}\ \tfrac{1}{2}\mathbf{A}^{(l+1)})_{\bullet 1}(2\mathbf{v}_1^{l} + 2\Delta\mathbf{v}_1^{l} + \Delta\mathbf{v}_1^{l+1}) & + \\ + \dots & + \\ +(\ \tfrac{1}{2}\mathbf{A}^{(l+k)})_{\bullet 1}(2\mathbf{v}_1^{l} + 2\Delta\mathbf{v}_1^{l} + \dots + 2\Delta\mathbf{v}_1^{l+k-2} + \Delta\mathbf{v}_1^{l+k-1}) & \end{aligned}$$

where $\Delta\mathbf{v}^{l} = \mathbf{v}^{l+1} - \mathbf{v}^{l}$.

The above columns can be collected in the matrix $\mathbf{F}^{(k,l)}$ defined as

$$\mathbf{F}^{(k,l)} = (\mathbf{E}^{(l+k)} \dots \mathbf{E}^{(l+2)}\ \tfrac{1}{2}\mathbf{A}^{(l)})_{\bullet 1} \mid \dots \mid (\ \tfrac{1}{2}\mathbf{A}^{(l+k)})_{\bullet 1}).$$

Then, two new vectors can be defined as

$$\begin{aligned} \tilde{\mathbf{u}}_{\mathbf{1}}^{k,l} &= \underbrace{(\mathbf{v}_1^{l}, \dots, \mathbf{v}_1^{l})^{T}}_{\text{k times}} \\ \Delta\tilde{\mathbf{u}}^{k,l} &= (\Delta\mathbf{v}_1^{l}, \dots, \Delta\mathbf{v}_1^{l+k-1})^{T} \end{aligned}$$

Using the new notation the vector on the right hand side of the matrix $\mathbf{F}^{(k,l)}$ can be expressed as

$$\begin{pmatrix} 2\mathbf{v}_1^{l} + \Delta\mathbf{v}_1^{l} \\ 2\mathbf{v}_1^{l} + 2\Delta\mathbf{v}_1^{l} + \Delta\mathbf{v}_1^{l+1} \\ \dots \\ 2\mathbf{v}_1^{l} + 2\Delta\mathbf{v}_1^{l} + \dots + 2\Delta\mathbf{v}_1^{l+k-2} + \Delta\mathbf{v}_1^{l+k-1} \end{pmatrix} = 2\tilde{\mathbf{u}}_{\mathbf{1}}^{k,l} + \\ +\mathbf{B}\Delta\tilde{\mathbf{u}}^{k,l}$$

where the matrix $\mathbf{B}$ is a lower-triangular matrix with one's on the diagonal and two's under the diagonal

$$\mathbf{B} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & \ldots & 0 \\ & \ldots & \ldots & \\ 2 & \ldots & 1 & 0 \\ 2 & 2 & 2 & 1 \end{pmatrix} \tag{24}$$

The predicted state can be written using the above derivation as

$$\begin{aligned} \mathbf{y}^{l+k} &= \mathbf{T}^{(k,l)}\mathbf{y}^{l} + \mathbf{F}^{(k,l)}(2\tilde{\mathbf{u}}_{\mathbf{1}}^{k,l} + \mathbf{B}\Delta\tilde{\mathbf{u}}^{k,l}) \\ \mathbf{y}^{l+k} &= \mathbf{T}^{(k,l)}\mathbf{y}^{l} + 2\mathbf{F}^{(k,l)}\tilde{\mathbf{u}}_{\mathbf{1}}^{k,l} + \mathbf{F}^{(k,l)}\mathbf{B}\Delta\tilde{\mathbf{u}}^{k,l} \end{aligned}$$

The formula contains three parts, the first represents the evolution of the initial condition with a zero boundary condition, the second one is the impact of the current applied control and the last term is a contribution from future controls. Therefore, the future control part has been isolated from the rest of the evolution operator.

There is also a recursive formula for the matrix $\mathbf{F}^{(k,l)}$. Following the above derivation it can be seen that proceeding from the state $\mathbf{y}^{l}$ to the state $\mathbf{y}^{l+1}$ we must (roughly said) multiply by matrix $\mathbf{E}^{(l+k+1)}$ and add a term containing the new boundary condition. Because only the first columns of the respective matrices are present in matrix $\mathbf{F}^{(k,l)}$ a straightforward examination shows that the following recursion is valid:

$$\mathbf{F}^{(k,l+1)} = \mathbf{S}_{-1}(\mathbf{E}^{(l+k+1)}\mathbf{F}^{(k,l)},\ \tfrac{1}{2}\mathbf{A}_{\bullet 1}^{(l+k+1)})$$

where $\mathbf{S}_{-1}(\mathbf{X}, \mathbf{z})$ is a shift operator which shifts the columns of matrix $\mathbf{X}$ to the left, in a way that $\mathbf{X}_{\bullet 1}$ is forgotten, the second column is moved to the first position, the third to the second and so on. The last column is set to be equal to vector $\mathbf{z}$.

$$\mathbf{S}_{-1}(\mathbf{X}, \mathbf{z}) = (\mathbf{X}_{\bullet 2}, \mathbf{X}_{\bullet 3}, \ldots, \mathbf{X}_{\bullet k}, \mathbf{z})$$

This concludes the proof of the theorem.