

PROCESS CONTROL USING CONTROLLED FINITE MARKOV CHAINS WITH AN APPLICATION TO A MULTIVARIABLE HYBRID PLANT

Enso Ikonen

*University of Oulu, Department of Process and Environmental Engineering, Systems Engineering Laboratory
4PYOSYS, FIN-90014 Oulun yliopisto, Finland
Enso.Ikonen@oulu.fi*

Keywords: Markov decision process, generalized cell-to-cell mapping, qualitative modelling.

Abstract: Predictive and optimal process control using finite Markov chains is considered. A basic procedure is outlined, consisting of discretization of plant input and state spaces; conversion of a (a priori) plant model into a set of finite state probability transition maps; specification of immediate costs for state-action pairs; computation of an optimal or a predictive control policy; and, analysis of the closed-loop system behavior. An application, using a MATLAB toolbox developed for MDP-based process control design, illustrates the approach in the control of a multivariable plant with both discrete and continuous action variables. For problems of size of practical significance (thousands of states), computations can be performed on a standard office PC. The aim of the work is to provide a basic framework for examination of nonlinear control, emphasizing in on-line learning from uncertain data.

1 INTRODUCTION

For identification and control of stochastic nonlinear dynamic systems, no general method exists. Development of physical models is typically far too time consuming and knowledge intensive, and results in models that are not well suited for process control design. Instead, in industrial practice, linear approximations have turned out to be most useful, commonly extended by considering local linear approximations (cf. gain scheduling, indirect adaptive control, piecewise (multi)linear multimodel systems, etc.), where linear descriptions vary with state and/or time.

For nonlinear plant identification, a multitude of efficient methods exists (Ikonen and Najim, 2002). These include polynomial functions, neural nets, etc. Identification of nonlinear dynamical relations is more difficult. Time series approaches (NARMAX, etc.) are a common and straightforward framework for extending to dynamic nonlinear systems. They rely in that mapping past (delayed) signals through a nonlinear static function will enable capturing the system dynamics. The properties of these models are, in general, difficult to analyze, however; complicating significantly the control design. A common simplifi-

cation is that of Wiener and Hammerstein systems: to consider static process nonlinearities only and equip the static nonlinear model with separate linear dynamics. This is a powerful approach in that the control design can be largely based on linear analysis, and in that knowledge on plant static nonlinearities can be exploited in the development of plant control. However, only linear dynamics can be dealt with.

This paper focuses on an approach that can cope with a large class of nonlinear systems: the finite Markov chains (Puterman, 1994) (Hägström, 2002) (Poznyak et al., 2000). The basic idea is simple. The system state space is quantized (discretized, partitioned, granulated) into a finite set of states (cells), and the evolution of system state in time is mapped in a probabilistic (frequentist) manner. With controlled Markov chains, the mappings are constructed from each state-action pair. Once equipped with such a model, a control action for each state can be deduced by minimizing a cost function defined in a future horizon, based on specification of immediate costs for each state-action pair. Specification of immediate costs allows versatile means for characterising the desired control behavior. Dynamic programming, studied in the field of Markov decision processes (MDP),

offers a way to solve various types of expected costs. Since a process model is available, the paradigm of model predictive control can also be used to derive the desired controls.

As the basic ideas are old, well-known, and widespread, relevant literature can be found from many fields and under different keywords: generalized cell-to-cell mapping (Hsu, 1987), qualitative modelling (Lunze et al., 2001), and reinforcement learning (Kaelbling et al., 1996), for example. Much of the terminology in the sections that follow originate from (Hsu, 1987): we refer to mappings between cells as (simple or) generalized cell maps, we use a sink cell, etc.

As pointed out in (Lee and Lee, 2004) (see also (Ikonen, 2004) (Negenborn et al., 2005)), applications of MDP in process control have been few; instead, the model predictive control paradigm is very popular in the process control community. Whereas not-so-many years ago the computations associated with finite Markov chains were prohibitive, the computing power available using cheap office-pc's enables the re-exploration of these techniques.

The work described in this paper aims at building a proper basic framework for examining the possibilities of controlled finite Markov chains in nonlinear process control. A majority of current literature on MDP examines means to overcome the problem of curse-of-dimensionality, e.g., by means of function approximation (neuro-dynamic programming, Q-learning, etc.). The main obstacle in such approaches is in that the unknown properties introduced by the mechanisms of function approximation make void the fundamental benefit of applying finite Markov chains: a straightforward and elegant means to describe and analyse the dynamic characteristics of a stochastic nonlinear system. Recall how linear systems are limited by the extremely severe assumption of linearity (affinity), yet they have turned out to be outmost useful for control design purposes. In a similar way, the finite Markov chains are fundamentally limited by the resolution of the problem presentation (discretization of state-action spaces). The hypothesis of this work is that keeping in mind this restriction (just as we keep in mind the assumption of linearity) the obtained results can be most useful. The practical validity of this statement is in the focus of the research.

In particular, the field of process engineering is in our main concern, with applications characterized by: availability of rough process models, slow sampling rates, nonlinearities that are either smooth or appear as discontinuities, expensive experimentation (large-scale systems running in production), and substantial on-site tuning due to uniqueness of products. Clearly,

this type of requirements differ from those encountered, e.g., in the field of economics (lack of reliable models), robotics (very precise models are available), consumer electronics (mass production of low cost products), telecommunication (extensive use of test signals, fast sampling), or academic toy problems (ridiculously complex multimodal test functions).

Due to systematical errors, noise, and lack of accuracy in measurements of process state variables, among many other reasons, there is a urgent need for extended means of learning and handling of uncertainties. The finite Markov chains provide straightforward means for dealing with both of these issues.

This paper is organized as follows: The process models are considered in section 2, control design in section 3, open and closed loop system analysis in section 4. The MATLAB toolbox, and an illustrative example is provided in section 5. Discussion on aspects relevant to learning under uncertainties, and conclusions, are given in the final section.

2 GENERALIZED CELL MAPPING

Let the process under study be described by the following discrete-time dynamic system and measurement equations

$$\begin{aligned} \mathbf{x}(k) &= f(\mathbf{x}(k-1), \mathbf{u}(k-1), \mathbf{w}(k-1)) \quad (1) \\ \mathbf{y}(k) &= h(\mathbf{x}(k), \mathbf{v}(k)) \quad (2) \end{aligned}$$

where $f: \mathcal{R}^{n_x} \times \mathcal{R}^{n_u} \times \mathcal{R}^{n_w} \rightarrow \mathcal{R}^{n_x}$ and $h: \mathcal{R}^{n_x} \times \mathcal{R}^{n_v} \rightarrow \mathcal{R}^{n_y}$ are nonlinear functions, $w_k \in \mathcal{R}^{n_w}$ and $v_k \in \mathcal{R}^{n_v}$ are i.i.d. white noise with probability density functions (pdf's) p_w and p_v . The initial condition is known via $p_X(0)$.

Let the state space be partitioned into a finite number of sets called state cells, indexed by $s \in \mathcal{S} = \{1, 2, \dots, S\}$. The index s is determined from

$$s = \arg \min_{s \in \mathcal{S}} \|\mathbf{x} - \mathbf{x}_s^{\text{ref}}\|$$

where $\mathbf{x}_s^{\text{ref}}$ are reference points (e.g., cell centers). In addition, let us define a 'sink cell', s_{sink} ; a state is categorized into a sink cell if $\min_{s \in \mathcal{S}} \|\mathbf{x} - \mathbf{x}_s^{\text{ref}}\| > \mathbf{x}^{\text{lim}}$. Similarly, let the control action and measurement spaces be partitioned into cells indexed by $a \in \mathcal{A} = \{1, 2, \dots, A\}$ and $m \in \mathcal{M} = \{1, 2, \dots, M\}$, respectively, and determined using reference vectors $\mathbf{u}_a^{\text{ref}}$ and $\mathbf{y}_m^{\text{ref}}$. The partitioning results in $\mathcal{X} = \cup_{s=1}^S \mathcal{X}_s$, $\mathcal{U} = \cup_{a=1}^A \mathcal{U}_a$ and $\mathcal{Y} = \cup_{m=1}^M \mathcal{Y}_m$.

The evolution of the system can now be approximated as a finite state controlled Markov chain over

the cell space (however, see (Lunze, 1998)). In simple cell mapping (SCM), one trajectory is computed for each cell. Generalized cell mapping (GCM) considers multiple trajectories starting from within each cell, and can be interpreted in a probabilistic sense as a finite Markov chain.

2.1 Evolution of States

Let the state pdf be approximated as a $S \times 1$ cell probability vector $\mathbf{p}_X(k) = [p_{X,s}(k)]$ where $p_{X,s}(k)$ is the cell probability mass. The evolution of cell probability vectors is described by a Markov chain represented by a set of linear equations

$$\mathbf{p}_X(k+1) = \mathbf{P}^{a(k)} \mathbf{p}_X(k)$$

or, equivalently,

$$p_{X,s'}(k+1) = \sum_{s \in S} p_{s',s}^a p_{X,s}(k)$$

where \mathbf{P}^a is the transition probability matrix under action a , $\mathbf{P}^a = [p_{s',s}^a]$.

3 CONTROL DESIGN

Using a GCM model of the plant, an optimal control action for each state can be solved by minimizing a cost function. In both optimal (Kaelbling et al., 1996) and predictive control (Ikonen and Najim, 2002) the cost function is defined in a future horizon, based on specification of immediate costs for each state-action pair. Whereas optimal control considers (discounted) infinite horizons and solves the problem using dynamic programming, nonlinear predictive control approaches rely on computation of future trajectories (predictions) and exhaustive search.

3.1 Optimal Control

In optimal control, the control task is to find an appropriate mapping (optimal policy or control table) π from states (\mathbf{x}) to control actions (\mathbf{u}), given the immediate costs $r(\mathbf{x}(k), \mathbf{u}(k))$. The infinite-horizon discounted model attempts to minimize the geometrically discounted immediate costs

$$J(\mathbf{x}) = \sum_{k=0}^{\infty} \gamma^k r(\mathbf{x}(k), \pi(\mathbf{x}(k)))$$

under initial conditions $\mathbf{x}(0) = \mathbf{x}$. The optimal control policy π^* is the one that minimizes J . The optimal cost-to-go is given by $J^* = \min_{\pi} J$.

Bellman's principle of optimality states that

$$J^*(\mathbf{x}) = \min_{\mathbf{u}} [r(\mathbf{x}, \mathbf{u}) + \gamma J^*(f(\mathbf{x}, \mathbf{u}))]$$

i.e., the optimal solution (value) for state \mathbf{x} is the sum of immediate costs r and the optimal cost-to-go from the next state, $J^*(f(\mathbf{x}, \mathbf{u}))$. Application of the Bellman equation leads to methods of dynamic programming.

3.1.1 Value Iteration

In value iteration, the optimal value function is determined by a simple iterative algorithm derived directly from the Bellman equation. Let the immediate costs be given in matrix $\mathbf{R} = [r^a]$, with column vectors $\mathbf{r}^a = [r_s^a]$, and collect the values of the cost-to-go at iteration i into a vector $\mathbf{J}^*(i) = [J_s^*(i)]$. Given arbitrary initial values $J_s^*(0)$, the costs are updated for $i = 0, 1, 2, \dots$:

$$\begin{aligned} Q_s^a(i) &= r_s^a + \gamma \sum_{s' \in S} p_{s',s}^a J_{s'}^*(i) \\ J_s^*(i+1) &= \min_{a \in \mathcal{A}} Q_s^a(i) \end{aligned}$$

$\forall s, a$, until the values of $J_s^*(i)$ converge. Denote the converged values by J_s^* . The optimal policy is then obtained from

$$\pi_s^* = \arg \min_{a \in \mathcal{A}} \left[r_s^a + \gamma \sum_{s' \in S} p_{s',s}^a J_{s'}^* \right].$$

3.2 Predictive Control

Given a system model and the associated costs, we can easily set up a predictive control type of a problem. In predictive control, the costs are minimized in an open loop in a fixed horizon

$$\begin{aligned} J(\mathbf{x}(k), \dots, \mathbf{x}(k+H_p), \mathbf{u}(k), \dots, \mathbf{u}(k+H_p)) \\ = \sum_{h=0}^{H_p} r(\mathbf{x}(k+h), \mathbf{u}(k+h)) \end{aligned}$$

under initial conditions \mathbf{x} . In practice it is useful to introduce a control horizon, where it is assumed that the control action will remain fixed after a given number of steps, H_c . Often only one step is allowed and the optimization problem reduces to the minimization of $J(\mathbf{x}(k), \dots, \mathbf{x}(k+H_p), \mathbf{u}(k))$.

Under control action a , the costs are given by

$$\begin{aligned} J_a &= \sum_{h=0}^{H_p} [\mathbf{r}^a]^T \mathbf{p}_X(k+h) \\ &= \sum_{h=0}^{H_p} [\mathbf{r}^a]^T [\mathbf{P}^a]^h \mathbf{p}_X(k) \end{aligned}$$

where $\mathbf{r}^a = [r_s^a]$ is a column vector of immediate costs and $\mathbf{p}_X(k)$ is current state cell pdf. In order to solve the problem, it suffices to evaluate the costs for all $a \in \mathcal{A}$ and select the one minimizing the costs. The prediction horizon H_p is a useful tuning parameter; a

long prediction horizon leads to mean level type of control.

The control policy mapping π° can be obtained by solving the above problem in each state s and tabulating the results: $\pi_s^\circ = \arg \min_a J_a$.

For many practical cases, a good controller design can be obtained using either the optimal control approach, or the predictive control approach with $H_c = 1$. In some cases, however, an engineer may be interested in extending the controller design possibilities to larger control horizons. In principle, this is straightforward to realize in the GCM context: One simply creates A different sequences of control actions, simulates the system accordingly, and selects the sequence that minimizes the cost function.

4 SYSTEM ANALYSIS

The generalized cell-to-cell mapping is a powerful tool for analysis of nonlinear systems. In what follows, it is assumed that the system map (Markov chain) is described by transition probabilities \mathbf{P} . This may correspond to the process output under a fixed (open loop) control action a ($\mathbf{P} := \mathbf{P}^a$) or the systems closed loop behavior obtained from the construction of transition probabilities under $\mathbf{u} = \pi(\mathbf{x})$: $\mathbf{P} := \mathbf{P}^\pi = \left[P_{s',s}^\pi \right]$.

4.1 Characterization of Cells

In GCM, an useful characterization of cells is obtained by studying the long term behavior of the Markov chain. A state is said to be recurrent (Najim et al., 2004) iff starting at a given state there will be a return to the state with probability 1. Otherwise the state is said to be transient. If $p_{s,s} = 1$, a state is said to be absorbing.

Decomposing the probability vector into recurrent cells ($i_r \in I_r$) and transient cells ($i_t \in I_t$), the Markov chain can be written as follows:

$$\begin{bmatrix} \mathbf{p}_r(k+1) \\ \mathbf{p}_t(k+1) \end{bmatrix} = \begin{bmatrix} \mathbf{P}_{rr} & \mathbf{P}_{rt} \\ \mathbf{0} & \mathbf{P}_{tt} \end{bmatrix} \begin{bmatrix} \mathbf{p}_r(k) \\ \mathbf{p}_t(k) \end{bmatrix}$$

As $k \rightarrow \infty$, the recurrent cells are visited infinitely often, whereas the transient cells are visited only finitely often. Among the recurrent cells, we can further classify the absorbing cells, ($i_a \in I_a$): $\mathbf{P}_{aa} = \mathbf{I}$. The absorbing states are never left, once visited.

The recurrent cells form communicating classes (closed subsets), where the cells within each communicating class (inter)communicate with each other, i.e., the probability of transition from one state to the other is nonzero, and do not communicate with other

states. Each absorbing state only communicates with itself. A closed communicating class constitutes a sub-Markov chain, which can be studied separately.

A stationary probability distribution satisfies $\bar{\mathbf{p}}_X = \mathbf{P}\bar{\mathbf{p}}_X$ and, consequently, the distribution must be an eigenvector of \mathbf{P} ; for the distribution to be a probability distribution, the eigenvalue must be one. Therefore, the recurrent cells are found by searching for the unit amplitude eigenvalues of \mathbf{P} ; the nonzero elements of the associated eigenvectors $\bar{\mathbf{p}}_X$ point to the recurrent cells.

4.2 Stability and Size of Basin-of-attraction

Examination of the behavior of transient cells as they enter the recurrent cells reveals the dynamics of the nonlinear system. We have that

$$\begin{aligned} \mathbf{p}_r(k+1) &= \mathbf{P}_{rr}\mathbf{p}_r(k) + \mathbf{P}_{rt}\mathbf{p}_t(k) \\ &= \mathbf{P}_{rr}\mathbf{p}_r(k) + \mathbf{P}_{rt}\mathbf{P}_{tt}^k\mathbf{p}_t(0) \end{aligned}$$

where $\mathbf{P}_{rt}\mathbf{P}_{tt}^k$ represents the conditional probability that a solution starting from a transient cell will pass into an recurrent cell at time $k+1$. The probability that this will eventually happen, \mathbf{P}_{t2r} , is given by

$$\mathbf{P}_{t2r} = \sum_{k=0}^{\infty} \mathbf{P}_{rt}\mathbf{P}_{tt}^k = \mathbf{P}_{rt}(1 - \mathbf{P}_{tt})^{-1}$$

Each recurrent cell belongs to a communicating class, for absorbing cells this class consists of a single cell. The probability of transition into a particular communicating class is obtained by summing (column-wise) the entries in \mathbf{P}_{t2r} .

The sink cell (Hsu, 1987) is an absorbing cell that represents the entire region outside the domain of interest. A nonzero probability to enter the sink cell indicates instability of the system (given the resolution of the model). In the experimental section, the stationary probabilities of entering the sink cell are examined.

High probability cells determine the basin-of-attraction. The 'size of the basin-of-attraction' for each state was characterized by taking the sum of probabilities for entering a recurrent cell (from any transient cell, or from any recurrent cell) and weighting it with the probability of occurrence within a communicating class (i.e., multiplying this with the stationary mapping \mathbf{P}_∞):

$$\mathbf{B} = \mathbf{P}_\infty \left[\sum_{i \in I_t} [\mathbf{P}_{t2r}]_{j,i} + \sum_{i \in I_r} [\mathbf{P}_{rr}]_{j,i} \right]$$

where \mathbf{P}_∞ is a mapping to stationary distribution: $\mathbf{P}_\infty = \lim_{n \rightarrow \infty} \frac{1}{n} \sum \mathbf{P}_{rr}^n$, and $[\mathbf{x}]_{a,b}$ denotes an element of \mathbf{x} in a 'th row and b 'th column. Elements of \mathbf{B} take

values in the interval $[0, S]$, $\sum_{i \in I_r} [\mathbf{B}]_i = S$. A large value in \mathbf{B} indicates the presence of the following ingredients: the communicating class to which a recurrent state belongs to can be accessed from a large number of states; the probability of entering the class from these states is high; the stationary probability for the occurrence of a recurrent state (within a communicating class) is high. Recall that when working with large state-action spaces, separate examination of all states is hopeless. In the experimental section we hope to bring some light to some control relevant properties of the (open or closed-loop) system by projecting \mathbf{B} to dimensions of \mathbf{x} .

Based on the above analysis, some communicating classes can then be taken under closer examination: simulation of (expected or random) trajectories, examination of cells in basins of attraction, etc. In assessing control performance, examination of the speed of the system (lengths of trajectories converging to communicating classes) is of great interest. The absorption time from the i 'th state to the j 'th state ($i \in I_t, j \in I_r$), $E\{\mathbf{k}\}$, is obtained from:

$$E\{\mathbf{k}\} = \mathbf{P}_{rt} \sum_{k=0}^{\infty} k \mathbf{P}_{tt}^k = \mathbf{P}_{rt} (1 - \mathbf{P}_{tt})^{-2}.$$

5 SIMULATION STUDY

In this section some numerical results based on simulations are given. The following control design problem set-up was envisioned: A nonlinear state-space model of the plant is available (a set of ordinary differential equations, for example), and a decision on input, state, and output variables has been made. A controller is now sought for, such that desired transitions between plant output set points would be optimal. A typical GCM control design procedure would involve the following (iterative) steps:

- Set model resolution by specifying discretization of plant inputs, states, outputs and output set points; and sampling time.
- Set control targets by specifying immediate costs.
- Build a GCM plant model (by successive evaluations of the original model, and counting the occurred state transitions) and analyze its behavior.
- Design a controller (e.g., optimal or predictive), based on the GCM plant model.
- Build a GCM closed-loop model and analyze its behavior.

5.1 Experiment Setup

Let us consider a simple example of a two-tank MIMO system (see (Åkesson et al., 2006) and ref-

erences there). In this hybrid system the action space (space of control inputs) consists of both real-valued and discrete-valued variables. The objective is to keep the temperature (T_2) in the second tank at its setpoint, while keeping the levels of both tanks (h_1, h_2) within preset limits. The system is controlled by a valve for the first tank input flow, a pump between the two tanks, a heater in the second tank, and a valve for the second tank output flow. The heater (u_1) is constrained to continuous values in the interval between 0 and 560 kW, the pump (u_2) has three operational levels {off, medium, high}, the valves (u_3, u_4) are binary {on/off}.

The system is described by the mass and energy balances

$$\begin{aligned} \frac{d}{dt} h_1 &= \frac{1}{A_1} (v_1 u_3 - \alpha u_2) \\ \frac{d}{dt} T_1 &= \frac{1}{A_1 h_1} (v_2 - T_1) v_1 u_3 \\ \frac{d}{dt} h_2 &= \frac{1}{A_2} (\alpha u_2 - v_3 u_4) \\ \frac{d}{dt} T_2 &= \frac{1}{A_2 h_2} (T_1 - T_2) \alpha u_2 + \frac{u_1}{c_1 \rho_1} \end{aligned}$$

where subscript 1 refers to the first tank (buffer), subscript 2 to the second tank (supply); v_1 is the inflow, v_2 the inflow temperature, and v_3 the outflow; A is the tank area ($A_1 = 3.5 \text{ m}^2, A_2 = 2 \text{ m}^2$), c_1 and ρ_1 are the liquid specific heat capacity and density ($c_1 = 4.2 \frac{\text{kJ}}{\text{kg K}}, \rho_1 = 1000 \frac{\text{kg}}{\text{m}^3}$); α is a pump capacity factor ($\alpha = 1 \frac{\text{m}^3}{\text{min}}$).

A discrete time Markov model (1)–(2) $\mathbf{x}(k) = f(\mathbf{x}(k), \mathbf{u}(k))$ for the system was constructed by forming the system state and controls as follows: $\mathbf{x}(k) = [h_1(k), T_1(k), h_2(k), T_2(k), v_1(k), v_2(k), v_3(k)]$ and $\mathbf{u}(k) = [u_1(k), u_2(k), u_3(k), u_4(k)]$, $y(k) = x_4(k)$. Since the model is based on another (deterministic) model we omit the disturbances here, $\mathbf{w} \equiv \mathbf{v} \equiv \mathbf{0}$ in (1)–(2). The state space was discretized by forming a grid, where tank levels and temperatures were quantized as follows:

$$\begin{aligned} x_1^{\text{ref}} &= x_3^{\text{ref}} = \{0, 1, 2, \dots, 9\} \text{ [m]} \\ x_2^{\text{ref}} &= \{17, 18, 19\}, x_4^{\text{ref}} = \{17, 18, \dots, 24\} \text{ [C]} \end{aligned}$$

disturbances into one and three values:

$$v_1 = v_3 = \{1\}, v_2 = \{17, 18, 19\},$$

and heating action in five values :

$$u = \{0, 140, \dots, 560\} \text{ [kW]}.$$

Roughly, the above states that deviations less than 0.5 °C in the supply tank temperature are out of our interest. Since we also want to place constraints on

high and low tank levels, with the above discretization we can set the switching point from allowed to non-desirable to 0.5 and 8.5 m. The quantizations in the disturbances allow to take known (step-wise) disturbances into account when deriving the optimal controls. The immediate costs were set based on the Euclidean norm between desired and reference temperatures, $\|w - y_s\|$ and deviation from nominal controls for u_2 , u_3 and u_4 at 1 with weights 0.1, 2 and 2 respectively (see (Åkesson et al., 2006)). For the states where reference points exceeded either upper or lower limits for the tank level, an additional large cost was added (ten times larger than largest cost so far). A 100 times larger cost was set for the sink cell.

5.2 Computer Simulations

The selected discretization results in a finite state-action space of 7201 states and 60 actions, including the sink cell. For the considered computational platform (a standard office PC: 3GHz Pentium 4 CPU, 1GB RAM, MATLAB R12) this posed no problem. In fact, up to ten times larger state-action spaces were experimented successfully.

A GCM model was built by evaluating the state transitions five hundred times for each possible state-action pair (s, a) . The starting state was generated from a random uniform distribution from within the state hypercube. This resulted in roughly 2×10^8 evaluations of the plant model; within each evaluation the ode were solved one sampling time (15 seconds) ahead using a standard ode-solver (MATLAB ode23). While most of the computing time was spent on solving the ode, the computations took a couple of hours. Clearly this presented a significant burden both in terms of computing power and memory, but not excessive at all. As long as the discretization of the state-action spaces is kept fixed during latter stages of control design, the model need not be re-evaluated, even if other parameters such as the immediate costs (\mathbf{R}) or controller design parameters (H_p or γ) would be modified.

Given the GCM model, a predictive controller was designed using $H_p = 5$. Using the obtained control policy π° , a closed loop GCM map was constructed.

The stability of the closed loop system is revealed by examination of the probabilities of entering the sink cell. The probabilities for entering the sink cell from any other cell were zero. Consequently, the closed loop system was stable for all initial states and set points.

Figure 1 illustrates the sizes of the basins-of-attraction, projected to three different dimensions of \mathbf{x} : level of tank 1 (x_1), level of tank 2 (x_3), and tem-

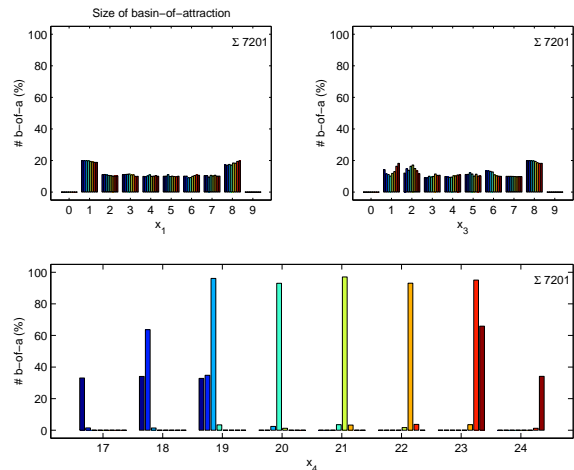


Figure 1: Size of basin-of-attraction. The probability mass for entering a particular state, projected to dimensions of \mathbf{x} . Top plots: tank levels, bottom plot: tank 2 temperature.

perature of tank 2 (x_4). The bars in the plots show the size of basin-of-attraction (as a percentage of whole state space), i.e., the cumulative number of states that are mapped to a recurrent state, recurrent states being sorted according to their projection to dimension of \mathbf{x} . From top plots in Fig 1, it can be immediately observed that the constraints on tank levels are fulfilled: the basin-of-attraction is empty for projections to levels 0 or 9, to both tanks.

The projections to tank 2 temperature, see bottom plot in Fig 1, show the success (controllability) of the plant to its set point (in steady state). For set points 19°C... 23°C, almost 100% success is obtained. For low temperatures, the smaller sizes of basins-of-attraction are explained by the lack of means to cool the incoming feed. Since one third of the states characterizes states with input feed equal to 19°C, one third in 18°C and one third in 17°C, it is easy to understand that the setpoint of 17 degrees is attained only when the input feed is 17°C, etc.

In few cases (three initial states), the model predicted transitions that could be judged as impossible using physical arguments. Closer examination of the plant model statistics revealed that these cases were due to the random sampling when building the GCM model, and the slow dynamics of the system. For example, an input feed in 19°C with no heating resulted the 18°C steady state temperature in both tanks. Examining the plant model, it was straightforward to attribute this to the fact that during the 500 simulations, none of the simulated trajectories had lead to another discrete state. Consequently, this state was categorized as absorbing. The remedy for this problem is to increase either the sampling rate, or the number of evaluations.

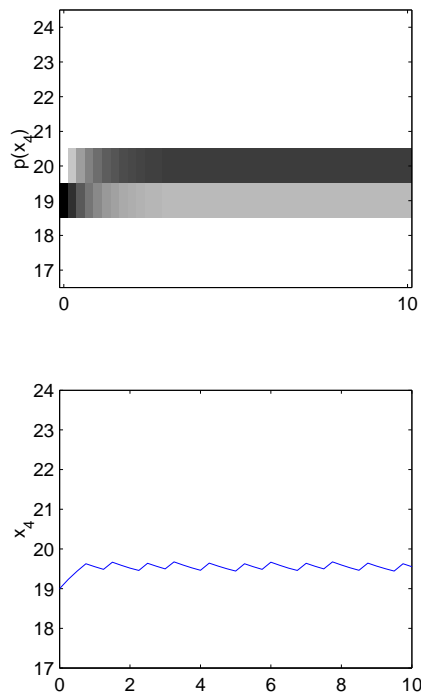


Figure 2: A communicating class consisting of two states. The closed loop system is 'ringing'.

For set points 19°C... 23°C, full 100% sizes of basin-of-attraction were not obtained. Instead, from bottom plot in Fig 1 it can be observed that a small percentage of the probability mass is distributed to the neighboring projections. A closer examination of these cases reveals a typical reason for this. Top plot in Fig. 2 shows the evolution of state in closed-loop from a particular initial state. The top plot shows the state probabilities projected towards the dimension x_4 (tank 2 temperature), for the case of setpoint in 20°C. As suggested by the plot, and as can be detected by examination of the communicating classes, the stationary distribution is a communicating class formed by two states. A sample trajectory is illustrated in the bottom plot, Fig. 2, showing that a phenomenon of 'ringing' clearly takes place. Most of the time is spent in the desired set point, but occasionally the system crosses the border and visits the state 19°C.

For a particular communicating class, or a state in it, very precise information can be obtained. For example, for the closed-loop system with set point in 20°C there were 182 communicating classes of which 67 were absorbing. For the 'ringing'-class example the basin-of-attraction contained 201 states (there was a nonzero probability of entering this class from 201 states). The expected time of absorption within these states ranged from 0 (from the recurrent states) to 8.85 minutes. Evolution of the transition probability dis-

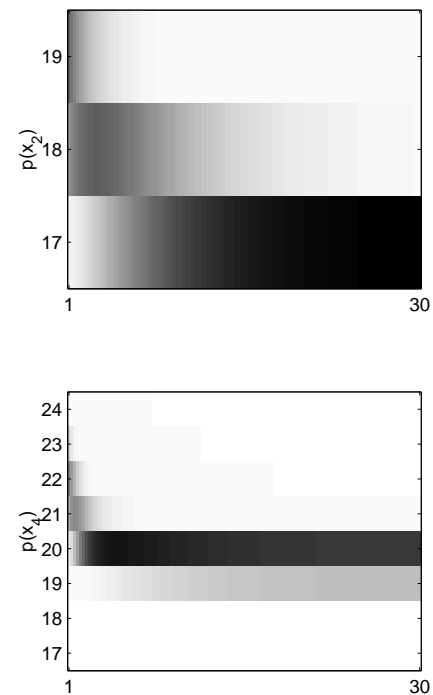


Figure 3: Evolution of probability distribution. The expected time of absorption is 9 minutes.

tribution (projected to two dimensions of \mathbf{x}) from the slowest initial state is shown in Fig. 3, confirming the exactness of this result. Unfortunately, it is not feasible to examine all states with this much care.

As a final example of system analysis, let us examine if there were any cases when a plant shutdown would happen (heating off, pump off, valves off). Examination of the policies (for all setpoints) revealed that only the sink cell resulted in plant shutdown. Another set of interesting control commands could be the one characterized by high pumping and closing of the output valve. It was observed that for each set point, the policy table contained roughly 800 occurrences of this control (i.e., in 800 out of 7201 states, this was the control to apply). These states were characterized by low levels in tank 2. Again, this appeared to make sense from an engineering point of view.

It can be concluded that convenient tools for analysis of the closed loop system were found, including examination of stability and steady state performance. The characterization of system performance in terms of speed of response was more tedious. It is not clear what could be done, as –for a nonlinear system– behavior differs from state to state, and computation of expectations and worst case scenarios does not necessarily reveal feasible information. A partial remedy is provided by the simple –and extremely commonly used– approach of simulating the closed loop

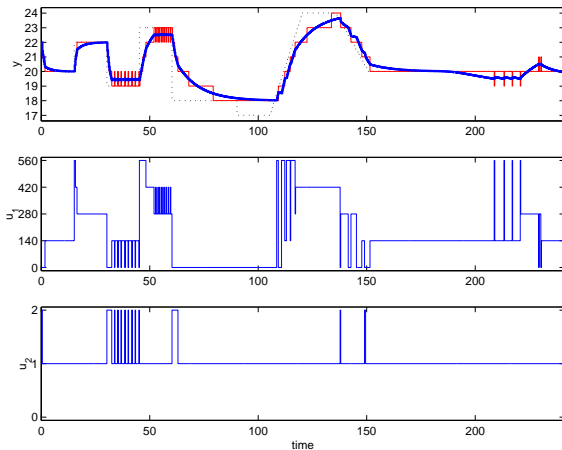


Figure 4: Closed-loop simulation. The set point trajectory consists of a series of steps and ramps. At the end of simulation, an input disturbance affects the system.

system under typical operating conditions. Figure 4 illustrates a trajectory following simulation with (a known) input disturbance. It can be concluded that the behavior of the closed loop system is adequate.

6 DISCUSSION AND CONCLUSIONS

In this preliminary work we have focused on using Markov chains and MDP as process control design tool, to be used bearing in mind the resolution of the problem set up (discretization into a finite state space). Continuing in the same direction, the problem of identification is then related to keeping the original model up-to-date (the ode, for example), or –at least– approximating the original model using function approximation techniques, rather than looking for clever tricks to make counting feasible in the finite state space, doomed to be huge. If doable, the benefits are clear: physical interpretation of estimated parameters. In many process engineering problems, this may turn out to be more fruitful than pure machine learning approaches.

Instead, the problem of uncertainty in measurements can potentially be handled in a very elegant and efficient fashion using finite Markov chains (Ikonen, 2004). Given the finite state probabilistic description of the plant, it is straightforward to construct cost functions taking into account the uncertainty in the predictions (other than discounted conditional expectations). Under the predictive control paradigm, also uncertainties in current state can be taken into account in plant predictions (i.e., there's no need to restrict to ML estimates, etc.). Conducting a literature review on these topics is a major direction in our future research.

REFERENCES

- Åkesson, B. M., Nikus, M. J., and Toivonen, H. T. (2006). Explicit model predictive control of a hybrid system using support vector machines. In *Proceedings of the 1st IFAC Workshop on Applications of Large Scale Industrial Systems (ALSIS'06)*, Helsinki-Stockholm, Finland-Sweden.
- Häggröm, O. (2002). *Finite Markov Chains and Algorithmic Applications*. Cambridge University Press, Cambridge.
- Hsu, C. S. (1987). *Cell-to-Cell Mapping - A Method of Global Analysis for Nonlinear Systems*. Springer-Verlag, New York.
- Ikonen, E. (2004). Learning predictive control using probabilistic models. In *IFAC Workshop on Advanced Fuzzy/Neural Control (AFNC'04)*, Oulu, Finland.
- Ikonen, E. and Najim, K. (2002). *Advanced Process Identification and Control*. Marcel Dekker, New York.
- Kaelbling, L. P., Littman, M. L., and Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285.
- Lee, J. M. and Lee, J. H. (2004). Approximate dynamic programming strategies and their applicability for process control: A review and future directions. *International Journal of Control, Automation, and Systems*, 2(3):263–278.
- Lunze, J. (1998). On the Markov property of quantised state measurement sequences. *Automatica*, 34(11):1439–1444.
- Lunze, J., Nixdorf, B., and Richter, H. (2001). Process supervision by means of a hybrid model. *Journal of Process Control*, 11:89–104.
- Najim, K., Ikonen, E., and Ait-Kadi, D. (2004). *Stochastic Processes - Estimation, Optimization and Analysis*. Kogan Page Science, London.
- Negenborn, R. R., De Schutter, B., Wiering, M. A., and Helendoorn, H. (2005). Learning-based model predictive control for Markov decision processes. In *16th IFAC World Congress*.
- Poznyak, A. S., Najim, K., and Gómez-Ramírez, E. (2000). *Self-Learning Control of Finite Markov Chains*. Marcel Dekker, New York.
- Puterman, M. L. (1994). *Markov Decision Processes – Discrete Stochastic Dynamic Programming*. Wiley et Sons, New York.