Predictive Multiple Model Switching Control with the Self-Organizing Map

Mark A. Motter
NASA Langley Research Center
Mail Stop 132
Hampton, Virginia 23681-2199
m.a.motter@larc.nasa.gov

Abstract —A predictive, multiple model control strategy is developed by extension of self-organizing map (SOM) local dynamic modeling of nonlinear autonomous systems to a control framework. Multiple SOMs collectively model the global response of a nonautonomous system to a finite set of representative prototype controls. Each SOM provides a codebook representation of the dynamics corresponding to a prototype control. Different dynamic regimes are organized into topological neighborhoods where the adjacent entries in the codebook represent the global minimization of a similarity metric. The SOM is additionally employed to identify the local dynamical regime, and consequently implements a switching scheme that selects the best available model for the applied control. SOM based linear models are used to predict the response to a larger family of control sequences which are clustered on the representative prototypes. The control sequence which corresponds to the prediction that best satisfies the requirements on the system output is applied as the external driving signal.

I. Introduction

Many modern approaches to the control of complex industrial process are quite naturally based on a model which accurately describes the evolution of the process as a function of its current state and the application of control inputs over a reasonable interval into the future [1], [12] . Multiple models of the process may provide a convenient means of providing this description under a wide variety of conditions [4], [5]. The use of multiple models necessitates a means of switching among the available models to the one that best describes the current operating environment. In a multiple model predictive controller framework, the control signals are generated by first switching to the model of the process that best matches the recently observed input-output behavior, then determining the best control signal by predicting what the process will do, while either implicitly or explicitly observing known constraints on the state of the system and the control. When a finite number of models are used to cover a broad range of system dynamics, coverage of the full dynamical space becomes an issue.

The Kohonen self-organizing map (SOM) [2], [3], is employed as the basis for dynamic modeling [9] and extended here to a control framework, where the modeled system is nonautonomous. The idea here is that the SOM, trained with responses from the full operating range, provides a basis for local dynamic models that fully cover the dynamical space corresponding to a *representative* or *prototype* control. For our application, control of the Mach number in a transonic wind tunnel, we were able to cluster the inputs onto a small set of prototypes. Local dynamic models which are linear in the control are derived from the SOM, enabling computationally efficient prediction of the system response to a larger set of pre-defined control inputs. We exploit the advantage of an approximate local model that is linear in the control input in contrast to an exact model which has a nonlinear dependence on the control as established by Narendra [8].

II. Self-Organizing Map

The Kohonen self-organizing map (SOM) , was adopted as the neural architecture for the experiment. The SOM was chosen based on its ability to transform an incoming signal of arbitrary dimension into a lower dimensional, discrete, topologically ordered map, one dimensional in this case. The spatial location of the processing elements (PEs), arranged in a one dimensional lattice, or linear array, corresponds to intrinsic features of the input signal.

One of the most control-specific applications of the SOM reported in the literature is the visuomotor control of a robot arm by Ritter, et al. [10]. In this application, the SOM is used as a look-up table, where the input pattern, identified by the "winner", specifies an SOM location associated with specified values of control parameters, which were learned adaptively. Some recent advances in this approach are elaborated in [11].

For our application, the SOM discretizes an n-dimensional space composed of output sequences of the system, $y(k), y(k-1), \ldots y(k-(n-1))$, which are considered to be the responses of the system to a prototype control input $u(k-1), u(k-2), \ldots u(k-m)$. Thus, the prototype input is the control parameter associated with all the nodes in the

0-7695-0619-4/00/$10.00 (C) 2000 IEEE

lattice, which is here, one-dimensional corresponding to the single control input to the system, u. In our application, the linearization is done around the “winner” to predict responses to candidate controls:

$$M_p = A_c(u_i - u_c) + M_c \ . \tag{1}$$

where M_c is the winner, A_c, is the Jacobian, derived directly from the SOM, u_c is the control prototype associated with the SOM and u_i is one of the candidate control sequences. In our application we replace the slow adjustment of control parameters by an external scheme, as in Ritter’s application, with the ability to switch, at discrete intervals, among the discrete local linear models associated with each node in the SOM. This highlights the difference between a slowly adaptive control scheme, and our application, which is designed to switch rapidly to accommodate abrupt changes in the system characteristics.

III. Control Using Multiple Models and Switching

The multiple model structure with switching has been proposed by Narendra et al., [4], [6], [7], when the overall system is required to operate in multiple environments. Sudden changes in parameter values, failures of sensors or subsystems, and external disturbances taken to be the output of an unforced stable dynamical system, can be considered as different environments a control system may be required to cope with. In these cases, the need to use multiple models arises naturally, since a different mathematical model may be needed to represent the behavior of the plant in each of the environments.

A recent paper [8] introduces two classes of approximate non-linear input-output models which reduce the computational complexity of designing a controller based on the fact that the approximate models are linear in the control input. This was essentially the approach taken in this experiment, where the converged SOM provides multiple, approximate models of the input-output behavior of the plant for a given class of input. These approximate models were then used as the basis for linear predictions of the response to a set of control candidates to determine the control input that minimized the error between the predicted output and the desired output.

IV. Modeling The Dynamics

The goal of the research was to capture the underlying dynamics of a nonautonomous system from observations of time-dependent, input-output data. The motivation for this approach came from previous work by Principe and Wang [9], using the self-organizing map as the infrastructure for local dynamic modeling of chaotic time series. Their work focused on modeling autonomous systems, that is systems where the state trajectory evolves without an external, or exogenous input signal driving the trajectory from one region to another in the state space. That work is adapted here to provide localized predictions of the system response, *p* steps ahead, to a predetermined set of input or *control* sequences which will drive the system toward the desired region of operation.

The assumption is that the state of the underlying nonautonomous system can be described as a differential equation of the form:

$$\frac{d\,\mathbf{x}(t)}{dt} = f\,(\mathbf{x}(t),\, u(t)) \tag{2}$$

where $\mathbf{x}(t)$ are the system states, *u(t)*, the *control signal*, is an *exogenous* input to the system, and f is the vector field that maps a Cartesian product of the state space, *S*, and the control space, C, $S \times C \subset \Re^n \ x \ \Re$, to a tangent space $T \subset \Re^n$. If a closed-form solution for (2) exists, that is : $\Phi\!: S \times C \to S$, then for a given initial condition **x**(0) and *u(t)* specified for all *t*, $\Phi(\mathbf{x}(0), u(t))$, represents a state-space trajectory of the system, or system flow.

An arbitrary, discrete non-linear dynamical system Σ, can be represented as:

$$\begin{aligned} \Sigma\!: x(k{+}1) &= f[x(k), u(k)] \\ y(k) &= h[x(k)] \end{aligned} \tag{3}$$

where $\{u(k)\}, \{x(k)\}$, and $\{y(k)\}$ are discrete-time sequences with $x(k) \in \Re^n$, $u(k) \in \Re, y(k) \in \Re,\ f: \Re^n \times \Re \to \Re^n, h: \Re^n \to \Re$, and $f, h \in C^\infty$. Here *f* is a map from the space of system states and input to the space of system states $\Re^n \times \Re \to \Re^n$, and *h* is a map from the space of system states to the output $\Re^n \to \Re$.

Our goal here is to determine the system output *y(k)*, over *p* steps into the future, in response to the application of a set of candidate control sequences $\mathbf{U}_{\mathbf{c}_i}$ where :

$$\mathbf{U}_{\mathbf{c}_i} = [u_{c_i}(k)\ u_{c_i}(k+1) \ldots\ u_{c_i}(k+p-1)] \tag{4}$$

is the *ith* candidate control sequence, and:

$$\mathbf{M}_{\mathbf{p}_i} = [y_{p_i}(k+1)\ y_{p_i}(k+2) \ldots\ y_{p_i}(k+p)] \tag{5}$$

is the predicted response from the *ith* candidate control sequence.

V. SOM-based Predictive Control

From (3), consider the output of the nonlinear system Σ :

$$\begin{aligned} y(k) &= h[x(k)] \equiv \Psi_1[x(k)] \\ y(k+1) &= h[f(x(k),u(k))] \equiv \Psi_2[x(k),u(k)] \\ y(k+2) &= h[f(\ f(x(k),u(k)),\ u(k+1)] \equiv \Psi_3[x(k),u(k),u(k+1)] \\ &\cdots \quad \cdots \quad \cdots \\ y(k+n) &= h \circ f^{\,n}[.,.] \equiv \Psi_{n+1}[x(k),u(k),u(k+1),\ldots,u(k+n-1)] \end{aligned} \tag{6}$$

where $f^{\,n}$ is an *n*-times iterated composition of f. Denoting the sequence *y(k*+1), …, *y(k + n)* by $Y_n(k)$ and the sequence *u(k), u(k*+1), …, *u(k + n* - 1) as $U_n(k)$, (6) can be expressed as :

$$\Psi[x(k),U_n(k)] = Y_n(k)\ . \tag{7}$$

For SOM-based predictive control, the main idea is that a set of maps can, collectively, be a global representation of these n-times iterated compositions of *f*, where an SOM winner represents the localized response of the system to a prototype control sequence, belonging to a larger set of control sequences, the *candidate controls*. Thus, the embedded state space is mapped into a neural field corresponding to a prototype control.

The second major idea is that predictors that are locally linear in the control can be constructed from the SOM winners. The construction of the locally linear predictors associated with the SOM winners is essentially a linearization around the weights of the winner:

$$\mathbf{M}_{\mathbf{p}_i} = \Phi_p[x(k),\mathbf{U}_p] + \ \nabla\Phi_p[x(k),\left\|\mathbf{U}_p - \mathbf{U}_c\right\|_1] \tag{8}$$

where $\left\|\mathbf{U}_p - \mathbf{U}_c\right\|_1$ is the $L1$ norm of the difference between the prototype control, $\mathbf{U}_p$ and the candidate control, $\mathbf{U}_c$, and $\nabla\Phi_p$ is the Jacobian with respect to the control, extracted from the converged SOM weights.

Ideally, perhaps, there would be an individual SOM, Φ_i , for each candidate control, $\mathbf{U}_{\mathbf{c}_i} = [u_{c_i}(k)\ \ u_{c_i}(k+1)\ \ldots\ u_{c_i}(k+p-1)\ ]$, and predictions of the tunnel response, $\mathbf{M}_{\mathbf{p}_i} = [y_{p_i}(k+1)\ \ y_{p_i}(k+2)\ \ldots\ y_{p_i}(k+p)\ ]$ would be made using the SOM winners:

$$\mathbf{M}_{\mathbf{p}_i} = \Phi_i[x(k),\mathbf{U}_{\mathbf{c}_i}]\ . \tag{9}$$

This would not have explored the concept of being able to extract a model that was locally linear in the control from the SOM and would have required excessive amounts of training data that was not available, i.e. an ensemble of responses for each candidate control over the entire operational range.

Thus the approach to modeling the tunnel dynamics evolved into a procedure consisting of two major components. First, the control input space was manually partitioned by the construction of significant prototype control vectors assumed to be capable of producing the general features of the desired wind tunnel response. Second, for each such partition of the control input space, a SOM was constructed from an ensemble of tunnel dynamic responses, i.e. the resulting Mach number response, covering the operating range. Each ensemble of Mach number responses was extracted from over 20 hours of actual wind tunnel data, covering the entire operational range. Collectively, the SOM(s) form an atlas of the global wind tunnel response due to the prototype control inputs.

The assumption here is that having an atlas for the system response to a set of control input prototypes provides a sufficiently complete modeling infrastructure, given the desired objective of predictively controlling the tunnel. There is no need to provide an infrastructure capable of modeling the response to all possible 3^p control sequences of length *p*, because it is assumed that the control inputs applied to the tunnel, at least in the PMMSC mode of operation, will come from the known set of candidate controls, which are either the control prototypes themselves, or close enough to the prototypes, by design, so as to predict the tunnel response by local models constructed from the response embedded in the input neural field of the corresponding SOM.

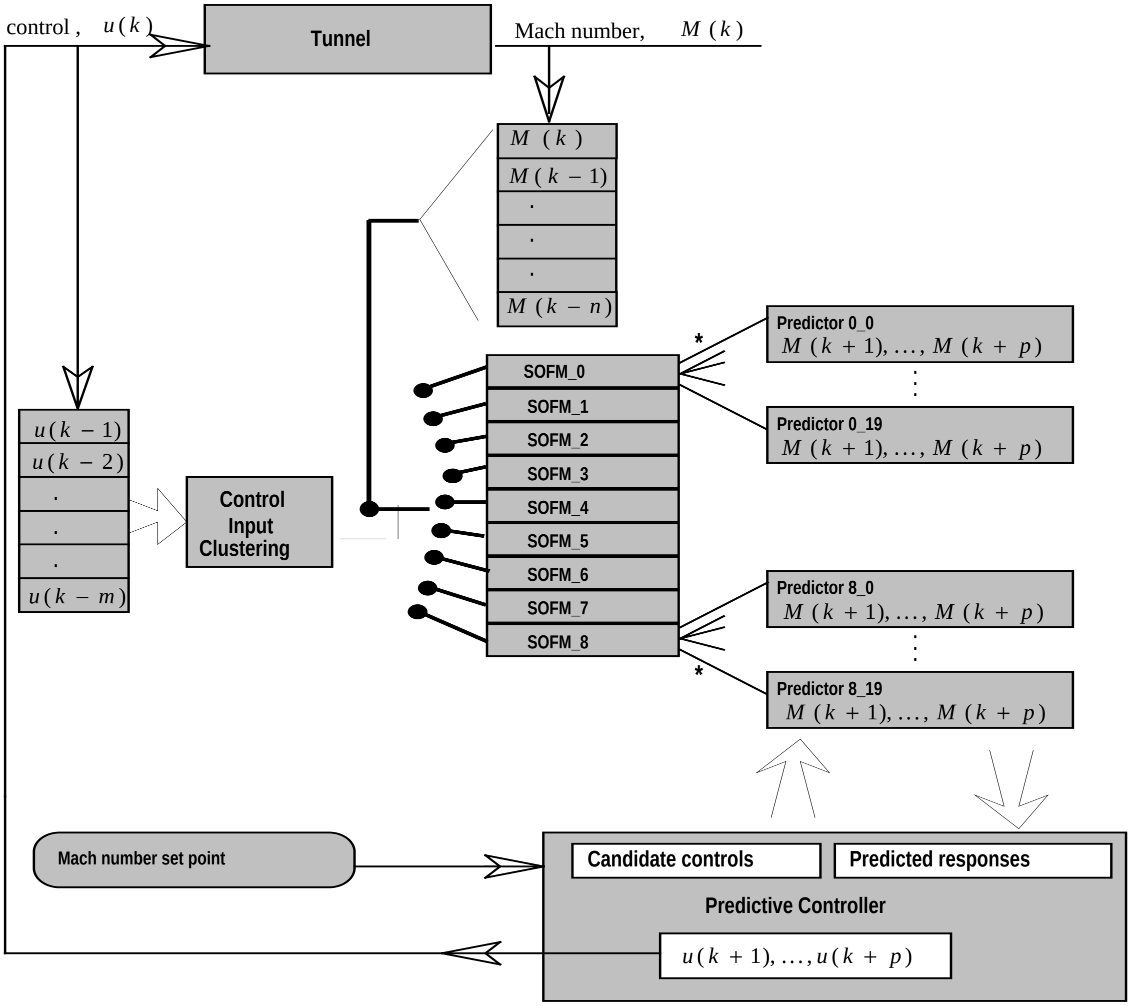

Figure 1. Experimental Framework with PMMSC

A. Experimental Framework

The experimental framework that evolved was essentially a predictive control scheme that used multiple models of the plant with *switching*. The controller switches between multiple, SOM-based models which, collectively, describe the global input-output behavior of the tunnel. The tunnel response to a set of candidate controls is predicted *p* steps ahead, using the currently selected model. The overall system, which will be referred to in the sequel as the PMMSC, for **P**redictive **M**ultiple **M**odel **S**witching **C**ontroller, is shown in Figure 1. It is composed of the following major functions:

1. The recent control input, $u(k-1),u(k-2),...,u(k-m)$, is clustered on a set of prototype control inputs which will choose one of the Kohonen self-organizing maps (SOM)
2. The selected SOM identifies the local dynamics of the tunnel based on the past $n+1$ Mach number measurements, $\mathbf{M} = M(k), M(k-1),\ldots, M(k-n)$, and chooses a winning processing element (PE)

3. A linear predictor associated with each PE predicts the Mach number response *p* steps into the future for each of the candidate controls
4. The predicted effectiveness of the candidate control inputs is evaluated over the last $(p-l)$ steps of the *p* steps-ahead predictions
5. The control input that provides the best response with respect to the Mach number set point is chosen as the next control, *u(k).*

The local model associated with the winning PE captures the dynamical regime of the wind tunnel. The controller still must decide what is the most appropriate control input to meet the set point specification. The controller sends candidate input sequences for *p*-step ahead prediction to the predictor of the winning node. The controller evaluates the relative effectiveness of the candidate control inputs in reducing the error between the predicted Mach number sequence, $\mathbf{M_P}$, and desired Mach number, $\mathbf{M_{SP}}$. This is accomplished by a suitable metric, the Euclidean norm over the error, $\left\|\mathbf{M_P}-\mathbf{M_{SP}}\right\|$ where $\mathbf{M_P} = M(k+l+1), M(k+l+2),\ldots, M(k+p)$ and $\mathbf{M_{SP}}$ is a $(p-l)$ length constant vector of M_{SP}.

Finally, the control input that provides the smallest error is sent to the wind tunnel fan control.

B. Experimental Results

Derived metrics to quantify the comparisons between the existing gain scheduled control, an expert operator, and the PMMSC are the *time out of tolerance* and the *L*1 norm of the control input, *u*. The *time out of tolerance* is cumulative sum of time that the measured Mach number deviates beyond the required tolerance of 0.003, and the L1 norm is the sum of the absolute value of the input commands over interval of interest.

	Existing controller	Expert Operator	PMMSC	% Reduction Auto / manual
Out of tolerance	329 s	310 s	266 s	19.1 / 16.5
L1 norm [u]	424.2	466.2	374.3	11.7 / 19.7

Table 1. Comparison for controlling to several different set points over a 28 minute interval

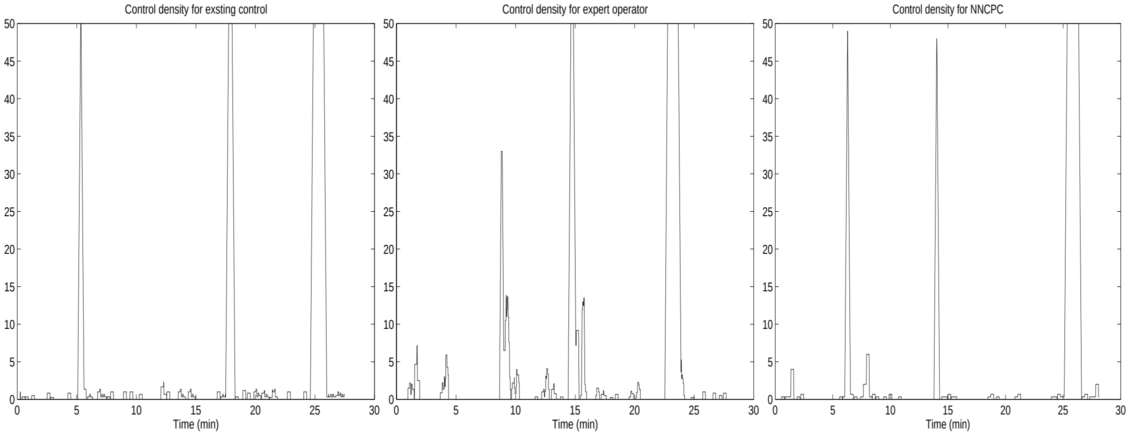

Figure 2. Comparison of Control Densities during set point changes

An additional metric on the control, the *control density,* ξ, was calculated by taking the sum of the absolute value of the control over a 50 sample sliding window:

$$\xi\,(k) = \xi\,[u(k)]=\sum_{i=0}^{i=49}\left|u(k-i)\right|. \tag{9}$$

The differences in the control density for the three cases are illustrated in Figure 2. The variation in the control density is greatest for the expert operator and least for the existing controller. The PMMSC falls between the two cases in terms of variation of the control density, while requiring less overall control effort to provide less time out of tolerance.

VI. CONCLUSION

Modeling and controlling systems with a wide range of dynamic characteristics is a rich problem with many possible approaches. In this research, the method of local linear modeling based on the self-organizing map has been extended to a control framework as an approach to this problem.

The SOM based modeling method was employed to develop a set of models which, collectively, described the system dynamic characteristics over the entire range of operation, but individually, represented the response of the system in some restricted region of both the state and control spaces of the system. The extension of the method allowed us to predict the system response to a small, but effective set of inputs, using the model which best describes the local dynamics. The input corresponding to the prediction that best satisfied the requirements at the output was then applied as the control. The overall result was the development of a controller, the PMMSC, which predicted the system response by switching to the best available model.

Two problems which naturally arise from this approach are: how to guarantee that the collection of models adequately cover all the dynamic regimes of the system, and how to select the model which best describes the local dynamical regime. Our SOM based local linear modeling approach addresses both the problems with a computationally efficient method. The SOM guarantees that the repertoire of dynamics used for training are represented by the collection of local models and serves to identify the local dynamic regime. In a sense, the diverse plant dynamics are captured in a compact table look-up of linear models.

The PMMSC was implemented on inexpensive computing hardware and used to control the wind tunnel to within the strict research requirements. The performance of the PMMSC was compared to both the existing controller and expert human operators by several metrics. The PMMSC provided improved performance with decreased control effort over both the existing controller and expert human-in-the-loop control.

VIII. REFERENCES

[1] D.W. Clarke, C. Mohtadi, and P.S. Tuffs, "Generalized Predictive Control - Part I. The Basic Algorithm," *Automatica*, Vol 23, No. 2, pp. 137-148, 1987.

[2] T. Kohonen, "The Self-Organizing Map," *Proceedings of the IEEE*, Vol. 78, No. 9., 1990.

[3] T. Kohonen, *Self-Organizing Maps*, Berlin, Heidelberg, Germany: Springer-Verlag, 1995.

[4] K.S. Narendra and J. Balakrishnan, "Intelligent Control using Switching and Tuning," *Proceedings of the Eighth Yale Workshop on Adaptive and Learning Systems*, June 13-15, 1994.

[5] K.S. Narendra and J. Balakrishnan, "Adaptive Control Using Multiple Models," *IEEE Transactions on Automatic Control*, Vol.42, No. 2, pages 171-187, 1997.

[6] K.S. Narendra, J. Balakrishnan, and M.K. Ciliz, "Adaptation and Learning Using Multiple Models, Switching, and Tuning," *IEEE Control Systems Magazine*, Vol.15, No. 3, 1995.

[7] K.S. Narendra, S. Li, and J.B.D. Cabrera, "Intelligent Control using Neural Networks,' *Proceedings of the Eighth Yale Workshop on Adaptive and Learning Systems*, June 13-15, 1994.

[8] K.S. Narendra, and S. Mukhopadhyay, "Adaptive Control Using Neural Networks and Approximate Models," *IEEE Transactions on Neural Networks*, Vol.8, No.3, pages 475-485, 1997.

[9] J. Principe, and L. Wang, "Nonlinear Time Series Modeling with Self-Organizing Maps." *IEEE Workshop on Neural Networks for Signal Processing*, pp. 11-20, 1995.

[10] H. Ritter, M. Martinez, K. Schulten, *Neural Computation and Self-Organizing Maps.* Reading, MA: Addision-Wesley, 1992.

[11] H. Ritter, "Self-Organizing Maps for Robot Control," *Proceedings of the 7th Internantional Conference on Artificial Neural Networks*, pp. 675-684, 1997

[12] Ronald A. M Soeterboek,, et al. , "A Predictive Controller for the Mach Number in a Transonic Wind Tunnel," *IEEE Control Systems Magazine*, Vol. 11, Number 11, 1991.