This article was published in an Elsevier journal. The attached copy is furnished to the author for non-commercial research and education use, including for instruction at the author's institution, sharing with colleagues and providing to institution administration.

Other uses, including reproduction and distribution, or selling or licensing copies, or posting to personal, institutional or third party websites are prohibited.

In most cases authors are permitted to post their version of the article (e.g. in Word or Tex form) to their personal website or institutional repository. Authors requiring further information regarding Elsevier's archiving and manuscript policies are encouraged to visit:

http://www.elsevier.com/copyright
Multiple-step-ahead prediction in control systems with Gaussian process models and TS-fuzzy models

Rainer Palm*

Technical University Örebro, Sweden

Received 10 August 2005; received in revised form 12 October 2006; accepted 11 February 2007
Available online 20 April 2007

Abstract

In this paper one-step-ahead and multiple-step-ahead predictions of time series in disturbed open loop and closed loop systems using Gaussian process models and TS-fuzzy models are described. Gaussian process models are based on the Bayesian framework where the conditional distribution of output measurements is used for the prediction of the system outputs. For one-step-ahead prediction a local process model with a small past horizon is built online with the help of Gaussian processes. Multiple-step-ahead prediction requires the knowledge of previous outputs and control values as well as the future control values. A “naive” multiple-step-ahead prediction is a successive one-step-ahead prediction where the outputs in each consecutive step are used as inputs for the next step of prediction. A global TS-fuzzy model is built to generate the nominal future control trajectory for multiple-step-ahead prediction. In the presence of model uncertainties a correction of the so computed control trajectory is needed. This is done by an internal feedback between the two process models. The method is tested on disturbed time invariant and time variant systems for different past horizons. The combination of the TS-fuzzy model and the Gaussian process model together with a correction of the control trajectory shows a good performance of the multiple-step-ahead prediction for systems with uncertainties.

Keywords: Gaussian process models; Prediction; TS-fuzzy models

1. Introduction

Modeling of control systems is motivated by many aspects. One aspect is the use of the model as prior knowledge in a control loop (Ljung and Glad, 1994). Other system models are used for observers for nonmeasurable states e.g. for fault diagnostics in complex systems (Isermann, 1997). Another important point of view is the prediction of the behavior of a system in a complex environment in the presence of disturbances and uncertainties for which the Kalman filter is one of the most useful tools (Welch and Bishop, 2002).

The classical way of modeling is based on physical laws usually formulated in terms of analytical state equations of the system (Ljung and Glad, 1994). If the physical knowledge is poor or not available a black-box model of the system may be extracted from training data. In most cases, however, a mixture of data-driven and analytical methods is superior to pure analytical or black-box techniques. Fuzzy multiple modeling is such a technique where a global nonlinear system is approximated by a number of fuzzily blended local linear or nonlinear models whose structures are supposed to be defined in advance (Johansen et al., 2000; Palm et al., 1996; Takagi and Sugeno, 1985; Tanaka and Sugeno, 1992). Location and area of validity of each local model are obtained from fuzzy clustering of training data (Palm and Stutz, 2003).

On the other hand, modeling with Gaussian processes is a probabilistic and non-parametric method taking into account disturbances and the uncertainty of the model. This is of great advantage for one-step-ahead or even multiple-step-ahead predictions with noisy time series and disturbed closed loop control systems (Girard et al., 2002; Gregoric and Lightbody, 2002; Kocijan et al., 2004;
Murray-Smith et al., 2003; Williams and Rasmussen, 1996). A good and comprehensive introduction to modeling with Gaussian Processes can be found in MacKay (1998).

A “naive” multiple-step-ahead prediction is a successive one-step-ahead prediction whereas the outputs in each consecutive step are considered as inputs for the next step of prediction (Girard et al., 2002). This assumes the previous outputs and control values to be known as well as the future control values. In these publications the model is trained offline which is reasonable for large sets of training data. The result is a global model of the system. In contrast to existing methods in the present paper a local online modeling with a Gaussian estimator using only few data is proposed. To compute an estimate of the \( (n+1) \)th output sample at each time step \( n \) the one-step-ahead prediction takes \( n+1 \) previous consecutive input samples into account plus a new input and a so-called output target vector. The target vector represents the previous \( n+1 \) outputs of the Gaussian estimator. This modeling works as a filter with a constant but moving time horizon for the past. Furthermore, in the previous literature the future input trajectory to the Gaussian estimator is always supposed to be known. However, only the nominal output trajectory is usually known in advance. The corresponding control trajectory is only known up to the present time step \( n \) but not for the future steps because of the uncertainties and disturbances in the control loop.

In this paper a new way to obtain the future controls is proposed. To obtain the future control inputs for the multiple-step-ahead prediction the system is modeled by an off-line trained TS-fuzzy model to generate a nominal control trajectory in closed loop for a given nominal output trajectory. In the case of larger model uncertainties the so generated control trajectory may differ too much from the real future control trajectory. Therefore, a correction of the predicted control trajectory is implemented. This correction is done by feedback of the difference between the outputs of the Gaussian estimator and the TS-fuzzy model. Additionally a stability check of the feedback loop between the linearized Gaussian process model and the linearized TS-fuzzy model is presented. Simulations with nonlinear systems with built-in uncertainties illustrate the good performance of the multiple-step-ahead prediction using a combination of Gaussian process models and TS-fuzzy models.

The paper is organized as follows. In Section 2 Gaussian processes as a method for modeling and regression is briefly presented. Section 3 deals with modeling of dynamical systems by Gaussian processes and an efficient one-step-ahead prediction with a constant but moving time horizon. Section 4 deals with the so-called “naive” multiple-step-ahead prediction for open loop systems. In Section 5, a TS-fuzzy model is introduced for multiple-step-ahead prediction in a closed loop control system. In Section 6, simulation examples are presented. Section 7 finishes with a conclusion.

2. Gaussian processes for modeling and regression

The following approach is based on the Bayesian framework (MacKay, 1998) which can be clarified by a scalar stochastic Gaussian process \( y = f(z) \) reflecting the input–output mapping \( z \to y \).

The goal is to infer the conditional probability

\[
P(y(n+1)|y_n, z_n) = \frac{P(y(n+1), y_n, z_n)}{P(y_n, z_n)} = \frac{P(y_n|y(n+1), z_n) \cdot P(y(n+1))}{P(y_n|z_n)}
\]

for an output measurement \( y(n+1) \) given the previous output measurements \( y_n = (y(1), \ldots, y(n))^T \) and input measurements \( z_n = (z(1), \ldots, z(n))^T \). This conditional distribution is used to make a prediction about \( y(n+1) \).

To be more general, a stochastic Gaussian process is a collection of random variables \( \{Y(z) | z \in Z \} \) with a multivariate Gaussian probability distribution with mean functions \( \mu(z) = E[Y(z)] \) and the covariance function

\[
C(z, z') = E[(Y(z) - \mu(z))(Y(z') - \mu(z'))]
\]

between two \( m \)-dimensional input vectors \( z = (z_1, z_2, \ldots, z_m)^T \) and \( z' = (z'_1, z'_2, \ldots, z'_m)^T \). The scalar output \( Y \) is supposed to be Gaussian distributed. With respect to modeling it has to be mentioned that there is no a priori model available. \( Y = f(z) \) is an input-output mapping \( z \to Y \) where the model is trained by \( n \) pairs of input data \( z(i) \) and target (output) data \( t(i) (i = 1, \ldots, n) \). Then a matrix of input vectors \( \mathbf{Z} = (\mathbf{z}(1), \mathbf{z}(2), \ldots, \mathbf{z}(n))^T \) and a target vector \( \mathbf{t} = (t(1), t(2), \ldots, t(n))^T \) are defined where the inputs are \( m \)-dimensional vectors \( z(i) = (z_1(i), z_2(i), \ldots, z_m(i))^T \) and the targets \( t(i) \) are scalars. The following covariance function \( C(\mathbf{z}(i), \mathbf{z}(j)) \) for two data samples \( \mathbf{z}(i), \mathbf{z}(j) \) has been found to work well (Williams and Rasmussen, 1996):

\[
C_{ij} = C(\mathbf{z}(i), \mathbf{z}(j)) = v_0 \exp \left( -\frac{1}{2} \mathbf{D}^T - \mathbf{W} \cdot \mathbf{D} \right)
+ a_0 + a_1 \mathbf{D}^T (\mathbf{z}(j)) + v_1 \delta(i, j),
\]

where \( \mathbf{W} = \text{diag}(w_1, \ldots, w_m) \), \( \mathbf{D} = \mathbf{z}(i) - \mathbf{z}(j) \). The parameters \( v_0, v_1, a_0, a_1, w_1, \ldots, w_m \) are hyperparameters. \( C_{ij} \) consists of a nonlinear term with factor \( v_0 \), a constant bias term \( a_0 \), a bilinear regression term with factor \( a_1 \), and a noise term \( v_1 \delta(i, j) \) where

\[
\delta(i, j) = \begin{cases} 
1 & \text{if } i = j, \\
0 & \text{else}.
\end{cases}
\]

The \( w_1, \ldots, w_m \) control the scaling of the distances in each input dimension \( z_1(i), \ldots, z_m(i) \). The parameter \( v_0 \) is the overall scale of correlations, \( v_1 \) is the variance of noise. The exponential term suggests that short distant input vectors lead to highly correlated outputs while long distant inputs generate low correlated outputs. The aim is: given a new test input vector \( \mathbf{z}(n+1) \), find the predictive Gaussian distribution of the corresponding output \( y(n+1) \) on the basis of Bayes’ law (MacKay, 1998; Williams and...
Rasmussen, 1996) which is completely described by its mean and variance (MacKay, 1998)
\[ \mu^r_{\text{new}} = \mathbf{k}^T(z_{\text{new}}) \mathbf{K}^{-1} \mathbf{t}, \]
\[ \sigma^2_{\text{new}} = \mathbf{C}(z_{\text{new}}, z_{\text{new}}) - \mathbf{k}^T(z_{\text{new}}) \mathbf{K}^{-1} \mathbf{k}(z_{\text{new}}), \]
where \( \mathbf{K} \) is the covariance matrix
\[ \mathbf{K} = \begin{pmatrix} C_{11} & \ldots & C_{1n} \\ \vdots & \ddots & \vdots \\ C_{n1} & \ldots & C_{nn} \end{pmatrix} \]
and
\[ \mathbf{k}(z_{\text{new}}) = (\mathbf{C}(z_{\text{new}}, z(1)), \ldots, \mathbf{C}(z_{\text{new}}, z(n)))^T, \]
\[ \mathbf{K}_{ij} = C_{ij}, \]
\[ \mathbf{t} = (t(1), t(2), \ldots, t(n))^T. \]
The parameters \( w_1, \ldots, w_m \) and \( v_1 \) can be initialized by a first analysis of the corresponding data. Then the initial values of the parameters \( v_0, a_0, \) and \( a_1 \) are chosen in a reasonable range, e.g. \( v_0 = 1, a_0 = 0, a_1 = 0. \) After that the hyperparameters are optimized by maximizing the log-likelihood function
\[ L = -\frac{1}{2} \log(\det(\mathbf{K}(\mathbf{\Theta})) - \frac{1}{2} \mathbf{t}^T \mathbf{K}(\mathbf{\Theta})^{-1} \mathbf{t} - \frac{n}{2} \log(2\pi)), \]
where \( \mathbf{\Theta} = (v_0, v_1, a_0, a_1, w_1, \ldots, w_m)^T \) is the vector of hyperparameters (see Girard et al., 2002; Kocijan et al., 2004, and especially MacKay, 1998).

3. Gaussian processes for one-step-ahead prediction in dynamical systems

The previous results are now applied to dynamical systems. Let a dynamical system be defined by
\[ x(n+1) = f(x(n), u(n)), \]
\[ y(n) = x(n), \]
where for a time step \( n \) the measured inputs to the system are \( x(n) = (x_1(n), x_2(n), \ldots, x_r(n))^T \)
\[ y(n) = (u_1(n), u_2(n), \ldots, u_u(n))^T \]
\[ z(n) = (x_1(n), x_2(n), \ldots, x_r(n), u_1(n), u_2(n), \ldots, u_u(n))^T \]
\[ = (z_1(n), z_2(n), \ldots, z_{r+u}(n))^T, \]
with the output vector \( y(n+1) \) at time step \( n+1 \) (see Fig. 1)
\[ y(n+1) = x(n+1) = (x_1(n+1), x_2(n+1), \ldots, x_r(n+1))^T. \]
It should be stressed that \( u(n) \) is the new control vector which leads to the new state \( x(n+1) \). Therefore, although containing the current state vector \( x(n) \), \( z(n) = (x(n), u(n))^T \) is the new input vector.

Given the input–output data \( \{z(i), y(i)\} i = 1 \ldots n \) we compute an estimate of the output \( y(n+1) \) using (3). According to (7)–(9) there are \( r \) individual models
\[ x^k(n+1) = f^k(z(n)) \]
\[ y^k(n) = x^k(n), \]
with the covariance matrices
\[ \mathbf{K}^k(n-1) = \begin{pmatrix} C_{11} & \cdots & C_{1,n-1} \\ \vdots & \ddots & \vdots \\ C_{n-1,1} & \cdots & C_{n-1,n-1} \end{pmatrix}, \]
i, j = 1 \ldots n - 1, \ \mathbf{K}_{ij}^k = C_{ij}^k(z_i, z_j) \]
the vectors
\[ \mathbf{k}^k(n-1) = (C^k(z(n), z(1)), \ldots, C^k(z(n), z(n-1)))^T \]
\[ \mathbf{t}^k(n-1) = (y^k(1), y^k(2), \ldots, y^k(n-1))^T \]
the weighting matrix
\[ \mathbf{W} = \text{diag}(w_1, \ldots, w_r, w_{r+1}, \ldots, w_{r+u}) \]
and the covariance function
\[ C_{ij}^k = C^k(z(i), z(j)) = v_0 \exp\left(-\frac{1}{2}(\Delta z^T \cdot \mathbf{W} \cdot \Delta z)\right) + a_0 + a_1 \Delta z^T(i) \Delta z(j) + v_1 \delta(i,j), \]
\[ \Delta z = z(i) - z(j). \]
From this we obtain for mean and variance
\[ \mu_{ij}^k(z(n)) = (\mathbf{k}^k)^T(\mathbf{k}^k)^{-1} \mathbf{t}^k, \]
\[ (\sigma^2_{ij}^k)^k = C^k(z(n), z(n)) - (\mathbf{k}^k)^T(\mathbf{k}^k)^{-1} \mathbf{k}^k. \]
How can this approach be used for prediction? In fact, with the help of a data set for the states, control inputs, and outputs one can find an estimate for the new output \( y^k(n+1) \) given the current state vector \( x(n) \) and the new control input vector \( u(n) \). Since for the feedback system (7) the new input vector is \( u(n) \) instead of \( u(n+1) \), the index for the covariance matrices (11) is \( n - 1 \) instead of \( n \). Once the model has been generated it is used for an interpolation between given points in the \( \{z(n), y(n)\} \)-space. In the following the mean \( \mu_{ij}^k \) is set to be equal to the estimate \( y^k(n+1) \). In order to compute \( y^k(n+1) \) the one-step-ahead prediction accounts for \( l+1 \) previous consecutive
samples $Z = (z(n-1), \ldots, z(n-1))^T$ plus $z(n)$ and a target vector $t^T(n-1) = (t^T(n-1), \ldots, t^T(n-1))$ representing the previous $l+1$ outputs of the Gaussian estimator. Once a new sample $n$ has arrived the oldest input/output pair is cancelled and a new inverse covariance matrix $(K^k(n))^{-1}$ and new vectors $(k^k(n))^T$ and $t^T(n)$ are calculated (block scheme see Fig. 2).

From the structure of the mean value (14) can be concluded that the Gaussian estimator works as a filter. The predictive mean

$$\hat{\mu}_k^{\hat{y}}(z(n)) = (k^k)^T (K^k)^{-1} t^k,$$

(15)

can be seen as a weighted sum of the previous targets $t^k$

$$\hat{\mu}_k^{\hat{y}}(z(n)) = s^T t^k,$$

(16)

where $s^T$ is the so-called smoothing kernel with the far left element of $s^T$ as the oldest one. The older the target elements the smaller are the weights for the target vector.

Let, without loss of generality, a 1st order linear SISO system illustrate this issue

$$x(n+1) = 1.14 \cdot x(n) + 0.1 \cdot u_d(n) + w(n),$$

$$u_d = 1.387 \sin(0.01 \cdot n);$$

$$y(n) = x(n) + v(n),$$

where $v(n), w(n)$, Gaussian noise with $\sigma = 0.004$. Fig. 3(a–d) shows the vector $s^T$ over the targets $t$ for four different past horizons $l = 150, 50, 20, 10$. From this example it can be concluded that a past horizon larger than $l = 50$ neither contribute much to the predictive mean nor to the RMSE. The convergence of the smoothing kernel for a growing past horizon can be easily shown by the fact that $k^k$ represents a

---

**Fig. 2.** Block scheme of one-step-ahead prediction by Gaussian processes.

**Fig. 3.** Evolution of the smoothing kernels for different past horizons $l = 150, 50, 20, 10$. 
vector of the covariances between the newest input and the previous inputs. Since in a stationary Gaussian process the correlation between two measurements declines with growing time distances, the covariance elements of $k^i$ also decline and the elements of $s^i$ as well. This fact is important to know for the simple practical reasons to choose an optimal past horizon.

4. “Naive” iterative multiple-step-ahead prediction

The next goal is to make a prediction with the last $l + 1$ samples from the real system and $r$ future samples from the Gaussian process model while knowing the samples from the real system and further one-step-ahead prediction with the input vector $y$ of prediction. In (Girard et al., 2002) a theoretical reason how to choose an optimal past horizon contributes to each future step which increases the advantage of this new approach is that the full past constant while computing the predictive outputs. The propagating the uncertainty is shown.

5. On-line generation of the control trajectory using TS-fuzzy rules

In the previous examples the control trajectory was supposed to be given in advance. However, in most cases this cannot be assumed. Usually one is faced with a closed loop system controlled by a given control law $u_d = R(y, y_d)$. Let furthermore the nominal outputs $y_d(n)$ be defined in advance up to a given prediction horizon $r$. Then the control inputs $u(n)$ of the system up to the present time step $n$ are known but not the future steps. To achieve the nominal values for $u(x) (n + k) (k = 1 \ldots r)$ a TS-fuzzy model (18) is built. The TS-fuzzy model generates the required control values in closed loop using the same control law as for the real system. Consider the real open loop system of nth order $x(n + 1) = f(x(n), u(n))$,

$$y(n) = x(n),$$

where $x(n) = (x(n), \ldots, x(n - m + 1))^T$ is the state, $u(n)$ is the control variable, $y(n)$ is the output, $f$ is a nonlinear function of $x$ and $u, n = m \ldots N$ is the running discrete time index. Based on $i$ training data sets $\{x(n), x(n+1), u(n)\}$ ($i = 1 \ldots M$) let (17) be approximated by an offline trained TS-fuzzy system represented by $c$ fuzzy rules

$$R^i: \text{if } x(n) \text{ is } X^i \text{ and } u(n) \text{ is } U^i \text{ then } x(n+1) = A_i x(n) + b_i u(n) + q_i. \quad (18)$$

where $X^i, U^i$ are fuzzy sets, $A_i \in \mathbb{R}^{m \times m}, b_i \in \mathbb{R}^{m \times 1}, q_i \in \mathbb{R}^{m \times 1}$ are local matrices, $c$ is the number of local linear models which are computed by fuzzy clustering and subsequent local modeling (linear regression) concluding with the multiple model

$$x(n + 1) = \sum_{j=1}^{c} w_j (A_j x(n) + b_j u(n) + q_j). \quad (19)$$

$$w_j = \frac{m_i}{\sum_{j=1}^{c} m_i}, \quad w_j = w_j(x, u) \in (0, 1), \sum_j w_j = 1.$$

is a weighting function (Palm et al., 1996) (Block scheme see Fig. 5).

$$m_i = m_i^{lX} \times m_i^{lU} \times \cdots \times m_i^{lX} \times m_i^{lU}$$

where $m_i^{lX} = p = 0 \ldots m - 1$ is the degree of membership for which “$x(n - p)$

![Fig. 4. Block scheme of the iterative $r$-step-ahead prediction.](image-url)
is $X_i^*$ and $m_{iU}$ is the degree of membership for which "$u(n)$ is $U_i^*$".

It is evident that in the future samples no unknown noise and uncertainty can be taken into account. To generate a nominal control trajectory $u^*(n+k) = u_d(n+k)$, $k = 1 \ldots r$ for the future steps $n+k$ of the real system the nominal output trajectory $y_d(n+k)$ is supposed to be given. To generate nominal control values that are close to the real ones, the control laws $R$ and $R^*$ for the closed loop system and the closed loop TS-fuzzy model have to be chosen to be identical. Then at each time step $n$ the TS-fuzzy model calculates a forecast for the next $r$ time steps. In each prediction step $k = 2 \ldots r$ the future control values $u^*(n+k)$ are fed into the Gaussian estimator together with the previous inputs and outputs of the system and the $r$ output estimates $\hat{y}(n+r)$ (see Fig. 6). The inputs for the Gaussian estimator are the same as for the pure naive approach except that the future control values $u^*(n+1) \ldots u^*(n+r)$ are not predefined but generated by the TS-fuzzy model. In this way we have two models working: the local Gaussian estimator which is an offline generated model predicting the outputs for a given time horizon from noisy measurements, and an online trained global TS-fuzzy model predicting the corresponding nominal control values.

5.1. Correction of the control trajectory by internal feedback

The method described above works well if the TS-fuzzy model fits the real system with a sufficient accuracy. If the model diverges too much from the real system then the generated control trajectory $u^*(n+1) \ldots u^*(n+r)$ leads to future outputs $\hat{y}(n+1) \ldots \hat{y}(n+r)$ which may differ from the real outputs $y(n+1) \ldots y(n+r)$ considerably. In order to compensate for this deviation the control trajectory $u^*(n+k)$ is corrected for both the TS-fuzzy model and the Gaussian estimator. It can be assumed that in the last prediction step the TS-fuzzy model is accurate enough to generate a first control input $u^*(n+1)$ for the Gaussian estimator. In the following prediction steps the output of the Gaussian estimator serves as a reference value because it uses the real data of the system from the past horizon. Therefore, the resulting local model is expected to be better than an inaccurate TS-fuzzy model. In the prediction steps $k = 1 \ldots r-1$ a correction $\Delta u$ of the control input $u^*$ is computed from the difference between the reference output $\hat{y}$ and the output of the TS-fuzzy model $y^*$

$$\Delta u(n+k) = K_f \cdot (\hat{y}(n+k-1) - y^*(n+k-1)), \quad k = 2 \ldots r,$$

$$\Delta u(n+1) = 0,$$

$$\hat{u}^*(n+k) = u^*(n+k) + \Delta u(n+k). \quad (20)$$

This correction is added to the control input $u^*$ of the TS-fuzzy model and the resulting $\hat{u}^*$ serves as the necessary control input for the Gaussian estimator in the steps $k = 2 \ldots r$ (see Fig. 7).

This feedback loop needs to be analyzed with regard to the stability for the combination of the Gaussian estimator and TS-fuzzy model. Because of the high nonlinearity of the Gaussian estimator a stability analysis is rather difficult but in the vicinity of the desired output $y_d$ local stability can be shown by linearization around the nominal state $x_d$ and control $u_d$. Let the control law of the TS-fuzzy model also be linearized

$$u^*(n) = R^C^* (x^*(n) - x_d(n)), \quad (21)$$

where

$$C^* = C = \begin{pmatrix} 1 & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & 0 \end{pmatrix}$$

is the output matrix with $y^*(n) = C^* \cdot x^*(n)$. 

---

![Fig. 5. Block scheme of the TS model.](image)

![Fig. 6. Block scheme for on-line generation of the control trajectory.](image)

---

Author's personal copy
A linearization of the TS-fuzzy model during prediction yields
\[
x^*(n+k+1) = (A^* + B^* R C^*) \cdot (x^*(n+k) - x_d(n+k)) \\
+ B^* Kf \cdot C^* (\hat{x}(n+k) - x^*(n+k)) \\
= (A^* + B^* R C^* - B^* Kf \cdot C^*) \cdot x^*(n+k) \\
+ B^* Kf \cdot C^* \cdot \hat{x}(n+k) \\
- (A^* + B^* R C^*) \cdot x_d(n+k).
\]

\(A^*(x_d, u_d), \quad B^*(x_d, u_d), \quad R^* = R, \quad k = 1 \ldots r.\) (22)

\(A^*(x_d, u_d)\) is assumed to be a stable system matrix whose eigenvalues lay inside a unit circle around zero, and the gain matrix \(R^*(x_d, u_d)\) is chosen in such a way that \(A^* + B^* R C^*\) is also stable. \(A^*(x_d, u_d)\) and \(B^*(x_d, u_d)\) can be easily obtained from (19)

\[A^*(x_d, u_d) = \sum_{i=1}^{c} w_i(x_d, u_d) A_i, \quad B^*(x_d, u_d) = \sum_{i=1}^{c} w_i(x_d, u_d) B_i.\]

The input–output mapping of the Gaussian estimator is highly nonlinear. However, a linearization of (15) in the vicinity of \(y_d\) during prediction yields

\[
\hat{x}(n+k+1) = \hat{A} \cdot \hat{x}(n+k) + \hat{B} \cdot (u^*(n+k) + \Delta u(n+k)) \\
+ \hat{f}_1([x(n), \ldots, x(n-l)]) + \hat{f}_2(x_d(n+k)) \\
= \hat{A} \cdot \hat{x}(n+k) + \hat{B} \cdot (Kf \cdot C^* \cdot x^*(n+k) \\
+ \hat{f}_1([x(n), \ldots, x(n-l)]) + \hat{f}_2(x_d(n+k)) \\
= (A + \hat{B} \cdot Kf \cdot C) \cdot \hat{x}(n+k) \\
+ (\hat{B} R - \hat{B} Kf C) \cdot x^*(n+k) \\
+ \hat{f}_1([x(n), \ldots, x(n-l)]) + \hat{f}_2(x_d(n+k)),
\]

where \(\hat{f}_1([x(n), \ldots, x(n-l)])\) includes the past horizon of the system, and \(\hat{f}_2(x_d(n+k))\) includes all nominal terms \(x_d(n+k)\) during prediction. Combining system equations (22) and (23) we obtain

\[
\begin{align*}
\hat{x}(n+k+1) \\
x^*(n+k+1)
\end{align*} = \left( \begin{array}{c}
\hat{A} \\
A^* + B^* R C^*
\end{array} \right) \cdot \hat{x}(n+k) \\
+ \left( \begin{array}{c}
\hat{B} \\
B^* - B^*
\end{array} \right) \cdot Kf \cdot C \\
\times \left( \begin{array}{c}
\hat{x}(n) \\
x^*(n+k)
\end{array} \right) + \left( \begin{array}{c}
\hat{f}_1 + \hat{f}_2 \\
-(A^* + B^* R C^*) \cdot x_d(n+k)
\end{array} \right)
\]

from which the feedback gain \(Kf\) can be designed in such a way that the combination works stable during prediction. The matrices \(A(x_d, u_d), \hat{B}(x_d, u_d),\) and the vectors \(\hat{f}_1(x_d, u_d)\) and \(\hat{f}_2(x_d, u_d)\) can hardly be computed analytically. The way out is a generation of a local linear black-box model of the Gaussian estimator from the input/output data sets gained from the uncorrected case.

6. Simulation examples

In order to illustrate the performance of the prediction methods a nonlinear SISO system is simulated for the following prediction examples:

1. One-step-ahead prediction with Gaussian process model.
2. Two-step-ahead prediction with Gaussian process model and time-varying system parameters.
3. Naïve multiple-step-ahead prediction with Gaussian process model and known control values.
In the last two examples a comparison with a prediction only based on the TS-fuzzy model is done. The stability analysis for a given correction feedback gain $K$ is illustrated for the 1st order and the 2nd order system example. In all the examples the following parameters for the Gaussian estimator have been chosen: $v_0 = 1; v_1 = 0.1; w_1 = w_2 = 1; a_0 = a_1 = 0$. 

Fig. 8. One-step-ahead prediction for a nonlinear system corrupted with noise $\sigma = 0.004$. (a) system output $y$; (b) error $e_y = \hat{y}(n+1) - y(n+1)$.

Fig. 9. One-step-ahead prediction for a nonlinear system corrupted with noise $\sigma = 0.04$. (a) system output $y$; (b) error $e_y = \hat{y}(n+1) - y(n+1)$.

Fig. 10. One-step-ahead prediction for a time variant nonlinear system with noise $\sigma = 0.04$. (a) system output $y$; (b) error $e_y = \hat{y}(n+1) - y(n+1)$. 
6.1. One-step-ahead prediction example

Define a 1st order nonlinear SISO system
\[ x(n + 1) = a_{11}x(n) + a_{12}x^2(n) + bu_d(n) + w(n), \]
\[ y(n) = x(n) + v(n), \]  
(25)

\[ a_{11} = 0.86; \quad a_{12} = -0.03; \quad b = 0.1; \quad u_d = 1.387 \sin(\omega \cdot n); \]
\[ \omega = 0.01, \text{where } v(n), w(n), \text{Gaussian noise with } \sigma = 0.004 \]
and zero mean, number of previous samples \( l = 10 \) steps.

Fig. 8 shows the simulation \( x(n) \) of the system and the error \( e = y(n + 1) - y(n + 1) \) between estimated and real output with a very good result showing a root mean square error (RMSE) between the estimated and the real output, RMSE = 0.0061. Increasing the standard deviation of the added noise by a factor of 10 leads to a 10 times higher RMSE = 0.0582 (see Fig. 9).

6.2. Time-variant system parameters

In order to show that the method also works with time varying systems let the system parameter \( a_{11} \) be time-dependent \( a_{11} = 0.86 + 0.13 \sin(0.01 \cdot n) \). The standard deviation of the added noise is \( \sigma = 0.04 \). The regarding results are shown in Fig. 10. The resulting RMSE = 0.065.

\[ \text{RMSE} = \frac{1}{N} \sum_{n=1}^{N} (\hat{y}(n+1) - y(n+1))^2 \]

where \( \hat{y}(n+1) \) is the predicted output, \( y(n+1) \) is the real output, and \( N \) is the number of samples.

6.3. “Naive” multiple-step-ahead prediction

The naive multiple-step-ahead prediction is applied to the same system as presented in the previous one-step-ahead example without time varying parameter. The future control trajectory is assumed to be given, and the noise signals added both to the system input and to the output have both standard deviations of \( \sigma = 0.004 \). The past horizon amounts to \( l = 5 \) and the prediction horizon is \( r = 10 \).

The performance of the prediction is measured by a final comparison of the predicted outputs and the outputs of the real system. Fig. 12 demonstrates an excellent simulation result of this “naive” prediction approach.

6.4. Generation of control values

This example refers again to the same system as in the last examples.
\[ x(n + 1) = a_{11}x(n) + a_{12}x^2(n) + bu_d(n) + w(n), \]
\[ y(n) = x(n) + v(n), \]
\[ e_y(n) = y(n) - y_d(n), \]
with the parameters \( a_{11} = 0.86; \quad a_{12} = -0.03; \quad b = 0.1, \]
\[ v(n), w(n), \text{Gaussian noise with } \sigma = 0.004 \text{ and zero mean}. \]
The control values are generated by the PI-control law
\[ u_d(n) = K_p \cdot e_y(n) + K_i \sum_{i=1}^{n} e_y(n), \]

Fig. 11. Evolution of the RMSE with the past horizon \( l, r = 1 \).

Fig. 12. (a) System output \( y(l = 5, r = 10) \); (b) error between predicted and real \( y \) RSME = 0.0322.
\[ K_P = -0.08, \]
\[ K_I = -0.01, \]

and the nominal output trajectory is \( y_d = \sin(0.01 \cdot n) \). The TS-fuzzy model is composed by six local models (see Palm and Stutz, 2003) using the following nominal trajectory for training and fuzzy clustering: \( y_d = \sin(0.01 \cdot n) - 0.5\sin(0.02 \cdot n) + 0.3\sin(0.02 \cdot n) + 0.5\sin^2(0.1 \cdot n) \).

Fig. 13 shows the results for a past horizon \( l = 5 \) and a prediction horizon \( r = 10 \). The prediction error is of the same small size compared with the last example without TS-fuzzy model. Fig. 13c shows the error \( e_f \) for a time variant parameter \( a_{11} = 0.86 + 0.03\sin(0.01 \cdot n) \).

An interesting point is the evolution of the RMSE depending both on the predictive horizon \( r \) and the past horizon \( l \). It can be assumed that the RMSE grows for a frozen past horizon \( l \) but an increasing predictive horizon \( r \). On the other hand an increasing past horizon \( l \) with a fixed predictive horizon \( r \) leads to a declining of the RMSE. This can be confirmed by the simulation examples the results of which are presented in Figs. 14 and 15. For the increase of \( r \) an almost linear increase of the RMSE can be identified (see Fig. 14). For the increase of \( l \) a steep decrease of the RMSE with a soon convergence to a constant value can be stated (see Fig. 15).

### 6.5. Correction of control values

In the next two examples it is shown how a correction of the control value in the presence of model uncertainties works.
6.5.1. 1st order nonlinear system

The system equation is

\[ x(n+1) = a_{11}x(n) + a_{12}x^2(n) + bu_d(n) + w(n), \]
\[ y(n) = x(n) + v(n), \]

with the same PI-control law as in the previous example. The TS-fuzzy model has been generated based on data from a system with the parameters \( a_{11} = 0.86; a_{12} = -0.03; b = 0.1 \). The feedback gain has been chosen to \( K_f = -1 \).

The parameters of the changed system are \( a_{11} = 0.84; a_{12} = -0.08; b = 0.1 \) where the TS-fuzzy model stays the same. The simulation results with correction improve the performance considerably which can be noticed by a much lower RMSE (see Fig. 16a–c). Using only the TS-fuzzy model for a prediction without any correction leads to very large errors if the TS-fuzzy model deviates strongly from the real system (see Fig. 16d).

A stability analysis in the framework of (24) is hardly feasible and even less feasible in the presence of integrators in the control loops of system and TS-model. In order to show comparable numerical results at all we simplify the analysis and check the stability of the feedback loop between the linearized Gaussian process model and the linearized TS-fuzzy model according to (24). In addition, the integrator gain \( K_I \) in the control law is set to zero taking into account that the system to be analyzed is only an approximation of the “real” system with \( K_I \neq 0 \). Linearization of the Gaussian process model and the TS-fuzzy model around zero leads to the parameters \( \hat{A} = 0.9376, \hat{B} = 0.0368, \hat{f}_1 + \hat{f}_2 = -0.0097, \end{equation} \( A^* = 0.8263, B^* = 0.1257 \). With \( K_f = -1 \) and \( R = R^* = -0.08 \) one obtains

\[ M = \left( \begin{array}{ccc}
\hat{A} & 0 & \hat{B} \hat{R} C \\
0 & A^* + B^* R^* C^* & \hat{B} \\
0 & B^* & -B^*
\end{array} \right) \cdot K_f \cdot C \]
with the eigenvalues $\text{eig}(M) = 0.9214 \pm 0.0619i$. This shows that the feedback between the two linearized models yields a stable behavior.

### 6.5.2. 2nd order nonlinear system

Another example is a 2nd order system with a nonlinear term.

\[
x(n + 2) = a_{11} x(n + 1) + a_{12} x(n) + a_{13} x^2(n) + bu_1(n) + w(n),
\]

\[
y(n) = x(n) + v(n),
\]

with the PI-control law $u_1(n) = K_P \cdot e_1(n) + K_I \cdot \sum_{i=1}^n e_i(n)$, $K_P = -0.108$; $K_I = -0.01$ and the feedback gain $K_f = (-1 0)$. The TS-fuzzy model has been built in the same way as for the 1st order system. There are again six local models based on the same nominal input trajectory shown in the last example. The TS-fuzzy model has been generated based on data from a system with the parameters $a_{11} = 1.986$; $a_{12} = 1.006$; $a_{13} = 0$; $b = 0.01$. The parameters of the changed system are $a_{11} = 1.984$; $a_{12} = 1.004$; $a_{13} = -0.001$; $b = 0.01$ where the TS-fuzzy model stays the same. Fig. 17a–c shows results which show that a correction of the control values decreases the RMSE. Using only the TS-fuzzy model for a prediction without any correction leads to larger errors if the TS-fuzzy model deviates from the real system (see Fig. 17d). Finally we check again the stability of the feedback loop between the linearized Gaussian process model and the linearized TS-fuzzy model (see (24)). The integrator gain $K_I$ in the control law is again set to zero. Linearization of the Gaussian process model and the TS-fuzzy model around zero leads to the parameters

\[
\hat{A} = \begin{pmatrix} 0 & 1 \\ 0.1522 & 0.6279 \end{pmatrix}, \quad \hat{B} = (0 \ 0.1086)^T,
\]

\[
\tilde{f}_1 + \tilde{f}_2 = (0 \ -0.0028)^T,
\]

\[
A^* = \begin{pmatrix} 0 & 1 \\ -0.4604 & 1.3823 \end{pmatrix}, \quad B^* = (0 \ 0.0395)^T.
\]

With $K_f = (-1 0)$ and $R = R^* = (-0.108 0)$ one obtains

\[
M = \begin{pmatrix} A & B \hat{R}C \\ 0 & A^* + B^* R^* C^* \end{pmatrix} + \begin{pmatrix} \hat{B} & -\hat{B} \end{pmatrix} \cdot K_f \cdot C
\]


Fig. 17. Correction of control values, 2nd order nonlinear system: (a) system output $y(n)$, ($l = 10, r = 10$); (b) prediction without feedback RMSE = 0.0419; (c) prediction with feedback RSMSE = 0.0376; (d) prediction with TS-fuzzy model only RMSE = 0.0746.
TS-fuzzy model is presented. Simulations of nonlinear the linearized Gaussian process model and the linearized addition a stability check of the feedback loop between Gaussian process model and the TS-fuzzy model. In feedback of the difference between the outputs of the predicted control trajectory is implemented using a trajectory. For larger model uncertainties a correction of trajectory in closed loop for a given nominal output was trained in advance to generate a nominal control system is modeled by a global TS-fuzzy model which next step of prediction. For unknown future control inputs in each consecutive step are considered as inputs for the successive one-step-ahead prediction whereas the outputs multiple-step-ahead prediction has been applied which is known as well as the future control values. A “naive” combination of Gaussian process models and TS-fuzzy models plus correction. This combination is superior to the use of Gaussian process models and TS-fuzzy models without correction and the use of a TS-fuzzy model alone.

\[
\begin{pmatrix}
0 & 1 & 0 & 0 \\
0.0436 & 0.6279 & 0.0969 & 0 \\
0 & 0 & 0 & 1 \\
-0.0395 & 0 & -0.4251 & 1.3823
\end{pmatrix},
\]

with the eigenvalues \(\text{eig}(M) = (-0.0528, 0.4062, 0.8568, 0.8000)\). This shows that the feedback loop between the two linearized models is stable.

7. Conclusion

In this paper the prediction of time series with a combination of Gaussian process models and TS-fuzzy models is proposed. This method is of advantage for one-step-ahead or even multiple-step-ahead predictions with noisy time series and disturbed closed loop control systems. The Gaussian process model approach is based on the Bayesian framework where the conditional distribution of output measurements of the system is used to make a one-step-ahead prediction for the system output. Usually global Gaussian process models are trained offline with large training data sets. In this paper \textit{local online modeling} with only few data is proposed where in each time step \(n\) the one-step-ahead prediction takes / previous consecutive input samples into account. A multiple-step-ahead prediction assumes the previous outputs and control values to be known as well as the future control values. A “naive” multiple-step-ahead prediction has been applied which is a successive one-step-ahead prediction whereas the outputs in each consecutive step are considered as inputs for the next step of prediction. For unknown future control inputs the system is modeled by a global TS-fuzzy model which was trained in advance to generate a nominal control trajectory in closed loop for a given nominal output trajectory. For larger model uncertainties a correction of the predicted control trajectory is implemented using a feedback of the difference between the outputs of the Gaussian process model and the TS-fuzzy model. In addition a stability check of the feedback loop between the linearized Gaussian process model and the linearized TS-fuzzy model is presented. Simulations of nonlinear systems with built-in uncertainties show a good performance of the multiple-step-ahead prediction with the combination of Gaussian process models and TS-fuzzy-models plus correction. This combination is superior to the use of Gaussian process models and TS-fuzzy models without correction and the use of a TS-fuzzy model alone.

References


