Abstract- Takagi-Sugeno (TS) fuzzy modeling and control techniques are applied to the classical nonlinear Ball and Beam problem. The nonlinear model is segmented to different local linear models. Local controllers are hence designed using LMI theory for achieving a robust control behavior for ball position. TS fuzzy modeling is applied to generate suitable fuzzy models, and the associated controllers. Using fuzzy if-then, a model knowledge based fuzzy control is then designed to regulate the original system response and behavior while relying on LMI-Lyapunov stability synthesis technique. That was achieved while interfacing the Ball and Beam system with fuzzy Gain Scheduling mechanism coding. For validating such control methodology, fuzzy control system has been synthesized and implemented practically.

Keywords- Takagi-Sugeno (TS) Fuzzy Modeling, LMI, Lyapunov Stability, Nonlinear Dynamics.

I. INTRODUCTION

We have witnessed rapidly growing interest in fuzzy control in recent years. This is largely due to the many successful applications of fuzzy control to nonlinear. There has been a great deal of research in addressing this issue as in [1],[2],[3],[4],[5],[6]. Among various fuzzy modeling themes, the wellknown TS model [7], as one of the most popular modeling framework. A general TS model employs an affine model with a constant term in the consequent part for each rule. This is often referred as an affine TS model. We shall refer to this type of TS fuzzy model as a linear models. The appeal of a linear TS fuzzy model is that it renders itself naturally to Lyapunov based system analysis and design techniques [8],[9]. A commonly held view is that a linear TS model has limited capability in representing a nonlinear system [10]. In [11], a controller structure called parallel distributed compensation PDC is introduced. This structure utilizes a fuzzy state feedback controller which mirrors the structure of the associated linear TS model. The idea is that for each local linear model, a linear feedback control is designed. The resulting overall controller, which is nonlinear in general, is a fuzzy blending of each individual linear controller. Applications of TS model together with PDC controller have achieved many successes in real systems [12],[13],[14]. A number of different nonlinear control approaches to this problem have been reported, refer to [15],[16]. Here a straightforward and intuitive technique is proposed, based on well known linear control theory, which leads to a gains scheduled controller that achieves a faster settling time than above cited approaches. The proposed method begins with constructing a TS type of a fuzzy model of the nonlinear system to be controlled. Here, a necessary algorithm enabling decomposition of the “white-box” nonlinear model into multiple linear sub-models describing locally the dynamics, is briefly included. Once the local linear TS fuzzy model is obtained, the corresponding fuzzy controller can be designed as a fuzzy gain-scheduler consisting of locally valid linear sub-controllers. The global control law is then obtained via fuzzy interpolation employing weighted contributions of all the local sub-controllers, by means of a soft fuzzy switching. The entire controller design can then be produced to searching for a solution of a set of LMI's. For these problems, powerful numerical algorithms have been developed recently .Such a design is constructive & guarantees stability within the considered wide-working range; also a performance bound can be prescribed [17],[18], [19]. On the other hand, since this approach is based on Lyapunov’s direct method, it leads inherently to conservative results. It can happen that when searching for a controller based on the given TS fuzzy model, no feasible solution exists. An alternative to this constructive approach to FGS base on LMI's are methods for mere heuristic designs (separate design of all local controllers). These methods are based on the multiple Lyapunov linearization approach and on subsequently designed local state feedback controllers that can yield good results only for very slowly varying trajectories. In the present approach applied to the ball and beam system, LMI techniques to solve locally multi objective linear time-invariant design problems having no offset terms have been utilized. Then fuzzy techniques have been employed to schedule among the resulting local controllers to reach the desired NL control law. This method is less conservative than those based on global LMI specifications for the corresponding local linear controllers. The reason for this is, such design algorithm does not automatically
guarantee stability and prescribed performance of the closed loop system for rapid changes of variables.

II. INTELLIGENT MODELING

2.1 NARX Models

The logical structure of rules facilitating the understanding and analysis of a model in a semi-qualitative manner, is close to the way human reason about the real world. Given the state of a system with a given input, the next state can be determined. In the sense of discrete-time setting, it can be written as in Eq.1:

\[ x(k + 1) = f(x(k), u(k)) \]  

where \( x(k) \) and \( u(k) \) are the state and the input at time respectively, and \( f(\cdot) \) is a static function. Fuzzy models of different types can be used to approximate the state-transition function. As the state of a system is often not measured, input-output modeling is usually applied. The most common is the NARX (Nonlinear Auto-Regressive with Exogenous input) model, as expressed by Eq.2:

\[ y(k + 1) = f(y(k), y(k - 1), ..., y(k - ny + 1), u(k), u(k - 1), ..., u(k - nu + 1)) \]  

In Eq.2, \( y(k), y(k - 1), ..., y(k - ny + 1) \) and \( u(k), u(k - 1), ..., u(k - nu + 1) \) denote the past model outputs and inputs respectively. In addition, \( (ny) \) and \( (nu) \) are integers related to the model order. For instance in Eq. 2, a linguistic fuzzy model of a dynamic system may consist of rules of the following formula:

\[ R_i: \text{If } x(k) \text{ is } A_i \text{ and } y(k-1) \text{ is } A_i \text{ and } ... \text{ then } y(k+1) \text{ is } C_i \]  

Eq.3 can approximate any observable and controllable modes of a large class of discrete-time nonlinear systems. A common source of information for building fuzzy models are the prior knowledge and data. Prior knowledge can be of a rather approximate nature (qualitative knowledge, heuristics), which usually originates from “experts”, i.e., system operators.

2.2 Takagi–Sugeno Model

The linguistic model, introduced in the previous section describes a given system by means of linguistic if-then rules, with fuzzy proposition in the antecedent as well as in the consequent. The Takagi–Sugeno (TS) fuzzy model (Takagi and Sugeno, [11]), on the other hand, uses crisp functions in the consequents. Hence, it can be seen as a combination of linguistic and mathematical regression modeling in the sense that the antecedents describe fuzzy regions in the input space in which consequent functions are valid. Hence, the TS rules have the following form:

\[ y = f(x) \]  

A simple and practically useful parameterization is the affine (linear in parameters) form, yielding the rules:

\[ y(x) = a_i x + b_i \]  

(5)

\( a_i \) is a parameter vector, and \( b_i \) is a scalar offset. This model is called an affine TS model.

2.3 Inference Mechanism

The inference formula of the TS model is a straightforward extension of the singleton model inference:

\[ y = \sum_{i} \beta_i y(x) = \sum_{i} \beta_i \left( a_i x + b_i \right) \]  

(6)

once the antecedent fuzzy sets define distinct but overlapping regions in the antecedent space and the parameters \((a_i)\) and \((b_i)\) correspond to a local linearization of a nonlinear function, the TS model can be regarded as a smoothed piece-wise approximation of that function.

2.4 TS model as a Quasi-Linear System

The affine TS model can be regarded as a quasilinear system (i.e., a linear system with input dependent parameters). To see this, denote the normalized degree of fulfillment by:

\[ \gamma_i = \beta_i(x)/\sum_{j=1}^{N} \beta_j(x) \]  

(7)

For the Linear Matrix Inequalities (LMI), many control problems and design specifications have LMI formulation. This is especially true for Lyapunov based analysis and design, but also for optimal LQG control, covariance control, etc. Further applications of LMIs arise in estimation, identification, optimal design, structural design, and so forth. The main strength of LMI formulations is the ability to combine various design constraints or objectives in a numerically tractable manner. A linear matrix inequality LMI is any constraint of the form:

\[ A(x):= A_0 + x_1 A_1 + ... x_n A_n < 0 \]  

(8)

where \( x = (x_1, ..., x_n) \) is a vector of unknown scalars (the decision or optimization variables) are given symmetric matrices \(<0\) stands for “negative definite”, i.e., the largest eigenvalue is negative. Note that the constraints \( A(x) > 0 \) and \( B(x) > 0 \) are special cases of Eq.8, since they can be rewritten as \( -A(x) < 0 \) and \( A(x) - B(x) < 0 \), respectively. Its solution set, called the feasible set, is a convex subset of \( \mathbb{R}^n \).

Finding a solution \( x \) to Eq.8, if any, is a convex optimization problem. Convexity has an important consequence: even though Eq.8. has no analytical solution in general, it can be solved numerically with
guarantees of finding a solution when one exists. Note that a system of LMI constraints can be regarded as a single LMI since:
\[ A(x) < 0 \]
\[ A(\dot{x}) < 0 \]
\[ \text{diag}(A_1(\dot{x}), \ldots, A_n(\dot{x})) < 0 \]

\( \text{diag}(A_1(\dot{x}), \ldots, A_n(\dot{x})) \) denotes the block-diagonal matrix with \( A_1(\dot{x}), \ldots, A_n(\dot{x}) \) on its diagonal. Hence, multiple LMI constraints can be imposed on the vector of decision variables (\( x \)) without destroying convexity. LMIs do not naturally arise in the canonical form Eq. 8, but rather in the form \( L(X_1, \ldots, X_n) < R(X_1, \ldots, X_n) \), where \( L(\cdot) \) and \( R(\cdot) \) are affine functions of some structured matrix variables \( X_1, \ldots, X_n \). An expression of inequality example, is the Lyapunov inequality:
\[ A^T X + X A < 0 \]  
\[ (9) \]
\( X \) is a symmetric matrix. Defining \( x_1, \ldots, x_N \) as independent scalar entries of \( X \), this LMI could be written in form of Eq. 8. Yet, it is convenient and efficient to describe it in its natural form as in Eq. 9.

**III. LM CONTROL SYNTHESIS**

3.1 Ball and Beam Nonlinear System

The ball and beam system is one of the most enduringly popular and important practical system for understanding control systems engineering. It has a very important property, it is open loop unstable.

There are two nonlinearities in Eq. 12, the \( x_1, x_2^2 \) term and the \( \sin(x_3) \) term. As well-known, most nonlinearity can be bounded by sector. For this system, assume \( x_3 \in (-\pi/2, +\pi/2) \) and \( x_4 \in (-d_1, +d_2) \). This is the region that the system will operate within. It follows that \( f(x) \) is expressed as:

\[ f(x) = \left( m_1 x_1 + m_2 x_2 + x_3 \right) v_1 + m_2 v_2 + \sin(x_3) \]

\[ g(x) = \left( 0, 0, 0, 0 \right)^T \]  
\[ (12) \]

![The Ball and Beam System](image)

**Fig.1. Ball and beam nonlinear dynamics system.**

3.2 Ball and Beam Linearized Model

The constants and variables for this example are defined as follows: (\( m \)) mass of the ball, (\( R \)) radius of the ball, (\( d \)) lever arm offset gravitational acceleration, (\( L \)) beam length, (\( f \)) ball’s moment of inertia, (\( d \)) ball position coordinate, (\( \theta \)) beam angle coordinate, (\( \theta \)) servo gear angle.

The Lagrangian equation of motion for the ball is given by the following:

\[ 0 = \left( \frac{I}{m} + m \right) \dot{\theta} + mg \sin(\theta) - mr(\dot{\theta})^2 \]  
\[ (10) \]

**State-Space:** The linearized system equations can also be represented in state-space form. This can be done by selecting the ball’s position (\( r \)) and velocity (\( \dot{r} \)) as the state variables and the angle \( \theta \) as the input. This is essentially controlling the torque of the beam. Below is the representation of this system:

\[ \begin{bmatrix} \dot{r} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{-mg}{R_m + m} \\ 0 \end{bmatrix} \begin{bmatrix} r \\ \theta \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u \\ y = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} r \\ \theta \end{bmatrix} \]  
\[ (11) \]

Nonlinear model of the system is given by the following equation:

\[ \left( 1 + \frac{2R^2}{5h^2} \right) \ddot{\theta} + \left( \frac{2R^2}{5h^2} \right) \dot{\theta} \dot{r} - g \sin(\theta) = h \]  

where \( h \) represents the rolling radius of the ball. Obtaining Linear TS fuzzy model for Ball & Beam System: the original model of Ball and Beam system is considered to be as illustrated in Fig. 1. Assuming no slipping between the ball and the beam, and while letting \( x = (r, \dot{r}, \theta, \dot{\theta}) \) as state of the system and \( y = (r) \) as the system output. The system can be expressed by the state space model as:

\[ f(x) = (x_2, B(x, x_4) - g \times \sin(x_3), x_4, 0)^T \\ g(x) = (0, 0, 0, 1)^T \]  
\[ (12) \]
3.3 Fuzzy Gain Scheduling

The controller design begins with the determination of the linear sub-models. Then, convex optimization techniques can be applied to solve a wide-range control problem.

\[ u(t) = -\sum_{i=1}^{n} h_i(\sigma(t))f_i(x(t)) \]  

The whole fuzzy controller, consisting of \( n \) valid local sub-controllers, are expressed as:

\[ u(t) = -\sum_{i=0}^{n-1} h_i(\sigma(t))f_i(x(t)) \]  

This represents a non-linear gain-scheduled control law. The closed loop system consisting of the fuzzy model and the fuzzy controller is obtained by substituting the controller Eq.15, into the state equation of the fuzzy model. The closed loop system is given by:

\[ x(t) = \sum_{i=1}^{n} \frac{1}{\sum_{j=1}^{n} r_{ij}} \left( A_i - B_i F_i \right) x(t) \]  

3.4 Regional Eigenvalue Constraints

LMI-based design techniques can be used to express performance specifications in terms of closed-loop eigenvalue locations for the underlying local sub-models. In the design proposed here, the eigenvalues are constrained to lie within an intersection of the vertical strip given by \( \sigma_1, \sigma_2 \) and the sector characterized by \( \theta \). A state feedback controller \( u(t) = -F_i x(t) \) places the eigenvalues of \( A_i + B_i F_i \) in such a region if the state feedback gain can be written as \( F_i = (K_i, Q_i)^{-1} \) and the matrices \( Q_i = K_i^T Q_i > 0 \) and \( K_i \) satisfy:

\[ L \otimes Q_i + M \otimes (A_i Q_i + B_i K_i) + M^T \otimes (A_i Q_i + B_i K_i)^T < 0 \]  

\( \otimes \) denotes the KRONECKER product, and:

\[ L = \begin{pmatrix} 2\sigma_2 & 0 & 0 & 0 \\ 0 & -2\sigma_2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \]

\[ M = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \sin \theta & -\cos \theta \\ 0 & 0 & \cos \theta & \sin \theta \end{pmatrix} \]

The set of solutions \( (Q_i, K_i) \) to the above linear matrix inequality represents the set of all state feedback gains that place the eigenvalues of local linear model in the specified region. This is used as constraint for minimizing a quadratic performance index:

\[ V = \int_0^T (x^TW_1x + u^TW_2u)dt \]  

To transform this integral cost function into an equivalent algebraic cost function, one can define a fictitious output signal:

\[ z = C_1 x + D_2 \]

\( C_1 \) and \( D_2 \) are weight matrices, to be chosen, such that the above cost function is equal to:

\[ V = \int_0^T z^T z dt \]  

Introducing a new matrix variable \( W_i \), one can show that the problem of finding a local state feedback gain matrix \( F_i \) that minimizes \( V \) locally is equivalent to the problem:

\[ (A_i Q_i + Q_i A_i^T + B_i K_i + K_i^T B_i + X_i) < 0 \]  

\( X_i = (Q_i)^{-1} \) for \( i = 1, 2, \ldots, r \). From the solution \( (K_i, Q_i, W_i) \) the optimal state feedback gain is obtained as \( F_i = -K_i Q_i^{-1} \). Once the LMIs of Eq 20 is added as an additional constraint, the solution is an optimal state feedback gain, that places the system poles over the best and optimal specified region.

IV. LM FUZZY CONTROL SYNTHESIS

System open loop is unstable because the system output, the ball position, increases without limit for a fixed input (beam angle). Feedback control must be used to keep the ball in a desired position over the beam. A controller that utilizes full state feedback control is designed for this nonlinear system. A schematic of the controller is in Fig.2.

4.1 State Feedback Synthesis

A and \( BK \) matrices are both (4x4). For this design, we desire an overshoot of less than (5%) which corresponds to \( \theta = 0, 7 \). For the reference input, now we want to get rid of the steady-state error. In contrast to the other design methods, where we feedback the output and compare it to a reference input to compute the error. With a full-state feedback controller we are feeding back both states.

We need to compute what the steady-state value of the states should be, multiply that by the chosen gain \( K \). This is done by adding a constant gain \( \bar{N} \) after the reference. For the fuzzy definition in Fig.3, a fuzzy logic controller is added in order to get rid of error signal. Such LMI synthesis has resulted in \( \bar{N} = (1828), K = (1828, 1028, 2008, 1) \). The fuzzy logic is designed to have two input variables \( (\bar{N}, \bar{U}) \), and one output variable, this is the error.

4.2 Non model Knowledge Based Fuzzy Control

In this section fuzzy rules are used to represent both the linear model of the system & the controller itself depending on our knowledge in the behavior of the system. Ball position and Beam angle are designed to
be the inputs of (mandani model) and the ball position also represents the system output. Seven membership functions are specified for both inputs as follows.

Ball Position: \((3.2.1.0,-1,-2,-3)\), where position \((0)\) refer to the center of the beam, and position \(3\) refer to the extreme right edge of the beam, and position \(-3\) (extreme left edge). Beam angles: \((90,60,30,0,-30,-60,-90)\). The angle \((0)\) means that there is no beam movement. System angles are measured with reference to the negative \(x\) axis. The output of this Non-model fuzzy control is stable, and this model represents the ball and beam system without any control being used.

For the Fuzzy Gain scheduling & LMI synthesis, the nonlinear model of the ball and beam system will be linearized using the technique described earlier, so that the operating region of the system will be centered into many linear operating regions according to the system input possibilities. For the ball and beam system models, and for the conditional case of \(x_{/4} - (0)\), and \(\theta=(\pi/4)\):

\[
A_1 = \begin{pmatrix}
0 & 1 & 0 & 0 \\
0 & 0 & -1.6Bg & 0 \\
0 & 0 & 0 & 2 \\
0 & 0 & 0 & 0
\end{pmatrix}
\]

For the conditional case of \(x_{/4} - (d/2)\), and \(\theta=(\pi/4)\):

\[
A_2 = \begin{pmatrix}
0 & 1.2 & 0 & 0 \\
0 & 0 & -1.17Bg & -0.7Bd \\
0 & 0 & 0 & 1.2 \\
0 & 0 & 0 & 0
\end{pmatrix}
\]

For the conditional case of \(x_{/4} - (d/2)\), and \(\theta=(\pi/2)\):

\[
A_3 = \begin{pmatrix}
0 & 1 & 0 & 0 \\
0 & 0 & -0.6Bg & 0.5Bd \\
0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0
\end{pmatrix}
\]

For computing the LMI controllers, this is done while employing the MATLAB LMI optimization function. It was found that state controllers gains are:

\[
K_1 = \begin{pmatrix}
-1 & -2.02 & 3.3 & 3.5
\end{pmatrix}
\]

\[
K_2 = \begin{pmatrix}
-1 & -2.31 & 2.5 & 2.9
\end{pmatrix}
\]

\[
K_3 = \begin{pmatrix}
-1 & -2.43 & 1.7 & 2.2
\end{pmatrix}
\]

The used fuzzy gain scheduling is shown in Fig. 3 and Fig.4. Typical such a system response is illustrated in Fig.5, the step response which has final value of (zero m) is applied to the input of the system, this value represents the desired ball position on the beam. Adding the LMI fuzzy controller which is described above, caused the output of the system to be stable, and to satisfy the design specifications.
To stabilize the system, feedback is added, in order to eliminate the error signal. This will affect the system response, since states are feed back to the input, are subtracted from the desired input value. This is causing the value of the error signal to be neglected. This exactly describe the function of the fuzzy logic controller, which is designed as shown in Fig.2 and Fig.3. The design of local models and controllers has overcome the stability problems of the system with faster settling time and less overshoot. Hence, every designed local model will have different response from another models, as shown in Fig.5 and Fig.6., as the ball and beam measured outputs differ according to the used model in state feedback. Fig.7. shows fuzzy scheduling rule based definitions, for ball position and beam angle.

CONCLUSION

This paper has proposed a linear fuzzy gain scheduler, for nonlinear dynamic system. The concept was based on linearization using fuzzy models, hence to synthesis local robust controllers based on LMI theory. The validity of the designed system was also validated. The scheduling variables do change slowly enough, hence the obtained scheduler can be used without restrictions. The approach was feasible to design local robust controllers for nonlinear systems.

REFERENCES


***