

PWATOOL: A MATLAB Toolbox For Piecewise-Affine Controller Synthesis

Mohsen Zamani Fekri ^a, Behzad Samadi ^b, Luis Rodrigues ^c

Abstract—In this paper, PWATOOL which is a free, third-party MATLAB toolbox for the study of piecewise-affine (PWA) systems is introduced. PWA controller synthesis are among the promising and practical control methodologies for nonlinear systems. PWA control systems design are computationally demanding and usually lead to bilinear matrix inequalities (BMI) which are NP hard to solve. Furthermore, PWA control synthesis, by Lyapunov approaches, can be done in a variety of ways which make it more complex when combined with the fact that not every Lyapunov approach converges for a given problem. However, so far, there has not been a unique software for studying the PWA systems, capable of analyzing the nonlinear systems using Lyapunov-based methods and synthesizing PWA controller for them. PWATOOL, as a software with the ability to analyze and synthesize PWA controllers, has been written to serve as an educational as well as a design tool in this way. PWATOOL uses Yalmip to solve the problems which verify the sufficient conditions for the stability of the models or the synthesis of PWA controllers. In addition, PWATOOL has only a few commands for modeling, analysis, synthesis and simulation which make it a simple and efficient tool for analysis and design.

I. INTRODUCTION

It is well known that these are usually the nonlinear controllers rather than linear controllers that have good stability and performance results for nonlinear systems [9], [20], [11] and [14]. This has motivated a lot of research on developing tools and nonlinear methods that can yield acceptable nonlinear controllers in terms of stability and performance. In this regard, piecewise-affine control synthesis, which builds on the previously known linear control design methodologies with proven stability and performance characteristics, is promising and practical. Piecewise-affine systems have widely been studied and used in the literature for modeling, stability and performance analysis [8], [6], [3], [7]. Lyapunov functions have been exploited as one of the main tools to analyze and design PWA controllers. However, PWA control systems design, by both global and piecewise quadratic (PWQ) Lyapunov function methods, are computationally demanding and usually lead to bilinear matrix inequalities (BMI) which are NP hard to solve. Nevertheless, there have only been few efforts in the literature to fill the gap between the theory of PWA systems and the tools which can comprehensively (as well as simply) perform the PWA system analysis and PWA controller synthesis [15], [16], [17] and [18]. PWATOOL is a free, third-party MATLAB toolbox for the study of PWA systems which has been developed in

Concordia University to address this issue. PWATOOL starts by importing a nonlinear or piecewise-affine (PWA) model. If the model is nonlinear, PWATOOL approximates it with Piecewise-affine Differential Inclusions (PWADI). When a PWA or PWADI model has been created, PWATOOL can analyze it for stability or synthesize PWA controllers for it, using Lyapunov function methodology. PWATOOL has a user interface and also several functions for creating and analyzing the PWA and PWADI models and synthesizing PWA controller for them. PWATOOL can be used for analysis and synthesis of small- to medium-sized models. It is based on solving feasibility or optimization problems subject to LMI or BMI constraints. PWATOOL uses Yalmip to solve the problems which verify the sufficient conditions for the stability of the nonlinear models or the synthesis of PWA controllers.

Different Lyapunov-based approaches in synthesis state-feedback PWA controllers are addressed by PWATOOL. These include both global and piecewise quadratic Lyapunov function methods, formulated as BMIs, by either ellipsoidal or quadratic curve approximation of the regions, as well as an LMI approach by ellipsoidal approximation of the regions. Likewise, for PWA systems analysis, four LMI methods, counterparts of those BMIs in the synthesis, are solved by PWATOOL. Solving the analysis or synthesis problems by several approaches comes in useful in studying and comparing these approaches for a trainee while a designer can use this feature to get as many results as possible from an initialization. However, the user can restrict the methods which PWATOOL uses for analysis or synthesis. This option is specially useful when some methods do not yield to efficient controllers or they are time consuming in computation.

Checking the systems for stability is only a matter of the convergence of feasibility problems. In this regard, the PWA controller synthesis differs from the analysis process in that synthesis usually needs to be excited by the user in certain regions of interest. That is, in PWA synthesis problem, unlike the analysis process, we like to have control over reshaping the controller so that each synthesis iteration can yield to a different PWA controller if desired. This ability adds to flexibility in control system design and would make us able to improve the performance of the closed-loop system through different runs of the code. In PWA controller synthesis, PWATOOL uses the approach in [13] to locally design a linear controller in the equilibrium point region and extend it to the whole domain of definition using global or piecewise quadratic Lyapunov functions.

^{a,c}: Electrical and Computer Engineering Department of Concordia University, Montreal, Canada. Email: luisrod@encs.concordia.ca
^b: Electrical Engineering Department of Amirkabir University, Tehran, Iran

The following sections are organized as follows. Section II studies the background and gives the definitions. It reviews the concepts we have used from the literature or the terms we use in PWATool framework. Section III gives an overview of the structure of PWATool and its main modules. Section IV explains the synthesis process and also gives the inequalities which are solved there. Section V explains the inequalities which are solved in analysis. Section VI solves a comprehensive example including modeling, analysis and synthesis by PWATool. Section VII gives a brief summary of the paper and addresses the possible future extensions to PWATool.

II. PRELIMINARIES

In this section, we explain the terms we use. We begin with the definitions of PWA and nonlinear systems, in the context that has been defined for PWATool.

A. A PWA Model; The Definition

A PWA model is described by NR set of linear dynamics, including affine terms, in NR regions R_i ($i = 1, 2, \dots, NR$) that form a \mathcal{C}^1 -function as follows:

$$\frac{dx}{dt} = A_i x + a_i + B_i u \quad \text{if } x \in R_i \quad (1)$$

where A_i is the linear dynamics, a_i the affine term and B_i the input gain in region R_i for $i = 1, 2, \dots, NR$. All the matrices are defined in MATLAB as variables of class *double*.

B. A Nonlinear Model; The Definition

A nonlinear model in PWATool is of the form

$$\frac{dx}{dt} = Ax + a + f(x) + B(x)u \quad (2)$$

where A is the linear dynamics, a the affine term, $f(x)$ the nonlinear dynamics and $B(x)$ the input gain. Here, A and a are defined to be variables of class *double* in MATLAB, whereas $f(x)$ would be a variable of class *function handle*. The parameter $B(x)$ is either a variable of class *double* or a *function handle* depending on whether it is a constant matrix or a function of x .

C. A PWADI Model; The Definition

A PWADI model builds on PWA modeling. Here, we have a pair of dynamics in each region which we may refer to them as lower- and upper-envelopes. A PWADI model, in regions R_i ($i = 1, 2, \dots, NR$), is described by envelopes σ_j ($j = 1, 2$) which are \mathcal{C}^1 -functions as follows:

$$\sigma_j(x) = A_{ij}x + a_{ij} + B_{ij}u \quad \text{if } x \in R_i \quad (3)$$

A PWADI model is used by PWATool to approximate a nonlinear model. By approximating a nonlinear model we mean that the nonlinear model is within the convex hull of σ_1 and σ_2 . This is to say, for a nonlinear dynamics $\frac{dx}{dt}$, defined by (2), we will have

$$\dot{x} \in \mathbf{conv}\{\sigma_1(x), \sigma_2(x)\} \quad (4)$$

A PWADI model has the characteristics that if it is stabilized by a PWA controller, then any nonlinear function defined

within its convex hull will be stabilized by that PWA controller [13].

D. Declaring The Variables In PWATool

PWATool starts by importing a nonlinear or PWA model. If the model is nonlinear, PWATool approximates it by a PWADI model. Other than the dynamics of the systems there are other parameters which need to be defined; These include the regions R_i , domains and a equilibrium point xcl which all are variables of class *double* in MATLAB [10]. However, declaring the variables in PWATool for creating PWA or PWADI models is easy and straightforward, based on the following rules:

- 1) *pwacreate* command in PWATool creates a pre-format m-file for the user.
- 2) The m-file, created by *pwacreate*, will contain the models variables and parameters as well as their sizes following a simple comment about them.
- 3) The file which is created by *pwacreate* can be updated, saved and run at any time to update the model.
- 4) PWATool accepts the model only if the sizes of the parameters, entered by the user, are consistent with each other.

For the sake of the space, we only explain the important features of variable declaration in this paper. A more detailed description can be found in Section VI where we solve an example and in the PWATool user manual [10].

Gridding The Regions: If the model is nonlinear, PWATool needs to know how the domains of the variables which appear in the nonlinear dynamics $f(x)$ in (2) should be gridded. The user specifies which of the following three gridding should be used by PWATool:

- 1) *Uniform*: The domain is gridded uniformly based on the number of regions (NR) which user chooses.
- 2) *OptimalUniform*: The domain is gridded uniformly based on the number of regions (NR) which user chooses. A norm of the approximation error is minimized in this mode.
- 3) *Multiresolution*: The domain is gridded based on a predetermined given regions. In this mode, the user can set certain points in the regions a priori.

If *Multiresolution* is chosen as the gridding type, the user can also specify a region $Rstar$ which will not be split by PWATool. $Rstar$ has the following applications:

- $Rstar$ can be defined such that the equilibrium point xcl is assured to be within it and not on the boundaries of two or more regions.
- $Rstar$ can be defined flexibly to increase the degree of freedom in defining the regions R_i 's.

E. Regions; Definitions and Approximations

Regions R_i , in modeling PWA and PWADI systems, are assumed to be either slab or convex polytopes:

$$R_i = \{x | E_i x + e_i \geq 0\}, \quad i = 1, 2, \dots, NR \quad (5)$$

where $E_i \in \mathbb{R}^{d \times n}$, $e_i \in \mathbb{R}^d$ and d is number of hyperplanes which describe the boundaries of the region. If the regions

R_i and R_j $i \neq j$ share a boundary with each other, then the parametric description of the intersection is contained in

$$\bar{R}_i \cap \bar{R}_j \subseteq \{x = F_{ijs} + f_{ij} | F_{ij} \in \mathbb{R}^{n \times n-1}, f_{ij} \in \mathbb{R}^n\} \quad (6)$$

where $s \in \mathbb{R}^{n-1}$ is a parameter and F_{ij} is a full rank matrix. However, in implementation of \mathbb{S} -procedure, it is sometimes desired to approximate the regions by ellipsoids (or degenerate ellipsoids in the case of slab regions) [12], [4] and [19]. A minimum volume ellipsoid containing a bounded polytopic region R_i [19] is described by

$$\varepsilon_i = \{x | \|E_i x + e_i\| < 1\} \quad (7)$$

If R_i 's are slab, (7) will yield a degenerate ellipsoid if $E_i = 2c_i^T / (\beta_2 - \beta_1)$ and $e_i = -(\beta_2 + \beta_1) / (\beta_2 - \beta_1)$ where we have assumed $R_i = \{x | \beta_1 < c_i^T x < \beta_2\}$ for some constant numbers $\beta_1, \beta_2 \in \mathbb{R}$.

We use the terms *ellipsoidal* or *quadratic curve* approximation to mean that respectively an ellipsoid or the polytopic region itself is used by \mathbb{S} -procedures in analysis or synthesis.

Remark 2.1: In this paper, we use the terms E_i and e_i for both the equations of the regions R_i and the ellipsoids which approximate the regions. Such similarity in names reflects the similar role that regions or their approximation play in analysis or synthesis inequalities. However, the context will clearly determine either regions or their ellipsoidal approximation has been used in \mathbb{S} -procedure.

F. Aggregated Vector Space $y = [x \ 1]^T$

For consistency purposes, PWATOOL, also saves the system data in an aggregated vector space $y = [x \ 1]^T$ in \mathbb{R}^{n+1} where x is the variable in the vector space \mathbb{R}^n .

III. PWATOOL; AN OVERVIEW

PWATOOL provides the inexperienced user with an interface which guides her or him, step by step, in learning how to use the commands and subroutines of PWATOOL. However, the main goal of using PWATOOL would be using its few commands rather than running its interface each time. Technically, PWATOOL has four main modules; those are modeling, analysis, synthesis and simulation modules where analysis is done by solving LMIs and synthesis is done by solving BMIs and LMIs for finding state-feedback PWA controllers. Figure 1 shows how the main four modules of PWATOOL may interact with each other. The important features in this interaction are:

- The model can be updated easily without changing the controller.
- The closed-loop system is simulated automatically, letting the user see the performance of the PWA controller, for every PWA controller which is obtained.
- Analysis and synthesis for PWA and PWADI systems are done with the same syntax (command).
- The controllers which are obtained in the synthesis can be checked again for stability by analysis module.

Each module is performed by a command and all the commands are common to PWA and PWADI models. This

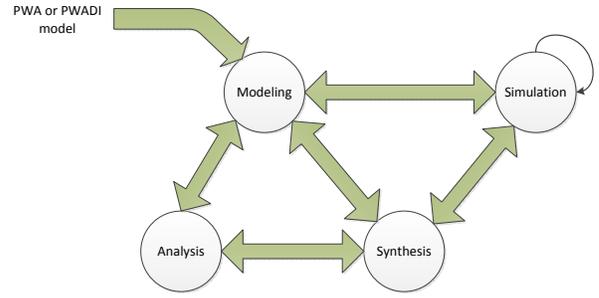


Fig. 1. Interaction Of PWATOOL Modules

TABLE I
LIST OF THE COMMANDS IN PWATOOL

Module	Command	Calling Format
Modeling	pwcreate	pwcreate(n,m, 'myfilename.m', NR) pwcreate(n,m, 'myfilename.m')
Analysis	pwaanalysis	pwaanalysis(pwsys, setting) pwaanalysis(pwsys, setting, Kbar)
Synthesis	pwasynt	pwasynt(pwsys, x0, setting)
Simulation	pwasimul*	pwasimul(pwsys, x0, Kbar)

makes PWATOOL an efficient and easy-to-use software in studying PWA systems. The list of PWATOOL commands is given in Table I.

In brief, the modeling module, depending on how it is called, either directly imports a PWA model or approximates a nonlinear model by PWADIs. After a PWA or PWADI model is constructed, analysis or synthesis modules can be called in any order. The model can also be updated without interrupting the existing controllers.

IV. THE THEORY OF SYNTHESIS; PWA AND NONLINEAR MODELS

In this section, the synthesis theories, used by PWATOOL, are explained. The synthesis module in PWATOOL in each iteration, if not restricted by the user, solves five feasibility problems: four BMI and one LMI approaches. The type of BMI approaches depend on whether the Lyapunov function is global or PWQ and also the approximation of the regions is ellipsoidal or by quadratic curves. The LMI approach is also addressed in [2]. In the following we explain the four BMI approaches in calculating PWA controllers. PWATOOL synthesizes PWA controllers for PWA and PWADI (or nonlinear) models to stabilize them around a desired equilibrium point x_{cl} . In synthesizing a PWA controller, x_{cl} need not necessarily be an open-loop equilibrium point; It will rather be a closed-loop equilibrium point where controller gains K_i and k_i are found to stabilize the system. PWATOOL uses the *Local Controller Extension* method [13] to design a local controller at the equilibrium point region and extend it later to the whole regions.

Remark 4.1: For the sake of simplicity, we provide the details of above approaches for only the PWADI (or nonlinear) models since the BMIs which are obtained for nonlinear

models can easily be used for PWA models considering the number of envelopes, there, is one (index j in Sections IV-A and IV-B will take only the value of 1).

Notation 4.2: Throughout this paper, for the sake of the space, we may use $\bar{A}_{ij} = A_{ij} + B_{ij}K_i$ and $\bar{a}_{ij} = a_{ij} + B_{ij}k_i$ to respectively denote the closed-loop dynamics and affine term.

A. Global Quadratic Lyapunov Synthesis

Consider the PWADI model (3) and let the controller outputs be obtained as

$$u = K_i x + k_i.$$

Global quadratic Lyapunov synthesis problem for a positive definite matrix Q and a constant positive scalar α solves the following inequalities:

- (1) $V(x) = x^T Q x > 0 \quad x \in \sigma_j$
- (2) $dV/dt = d(x^T Q x)/dt < -\alpha V(x) \quad x \in \sigma_j.$

The following BMIs are obtained:

Problem 4.3: Global Quadratic Synthesis (quadratic curve approximation); For a constant positive scalar α , find Q , K_i , k_i and Z_i that satisfy the following BMIs for $i, h = 1, 2, \dots, NR$ and $j = 1, 2$:

- (1) $Q > 0, Z_i > 0$
- (2) In regions R_i which contain the equilibrium point xcl

$$Q\bar{A}_{ij} + \bar{A}_{ij}^T Q + \alpha Q < 0$$
- (3) In regions R_i which do NOT contain the equilibrium point xcl

$$\begin{bmatrix} Q\bar{A}_{ij} + \bar{A}_{ij}^T Q + \alpha Q + E_i^T Z_i E_i & * \\ (Q\bar{a}_{ij} + E_i^T Z_i e_i)^T & e_i^T Z_i e_i \end{bmatrix} < 0$$
- (4) Continuity of the control outputs (OPTIONAL)

$$\begin{bmatrix} (\bar{A}_{ij} - \bar{A}_{hj}) * F_{ih} \\ (\bar{a}_{ij} - \bar{a}_{hj}) * f_{ih} \end{bmatrix} = 0 \quad \text{for } F_{ih} \text{ not empty}$$
 where F_{ih} and f_{ih} are defined by (6)

A similar set of BMIs are obtained if the S-Procedure uses the ellipsoidal approximation of the regions:

Problem 4.4: Global Quadratic Synthesis (ellipsoidal approximation); For a constant positive scalar α , find Q , K_i , k_i and μ_i that satisfy the following BMIs for $i, h = 1, 2, \dots, NR$ and $j = 1, 2$:

- (1) $Q > 0, \mu_i < 0$
- (2) In regions R_i which contain the equilibrium point xcl

$$Q\bar{A}_{ij} + \bar{A}_{ij}^T Q + \alpha Q < 0$$
- (3) In regions R_i which do NOT contain the equilibrium point xcl

$$\begin{bmatrix} [Q\bar{A}_{ij} + \bar{A}_{ij}^T Q + \alpha Q + \mu_i E_i^T E_i & * \\ (Q\bar{a}_{ij} + \mu_i E_i^T e_i)^T & -\mu_i(1 - e_i^T e_i) \end{bmatrix} < 0$$
- (4) Continuity of the control outputs (OPTIONAL)

$$\begin{bmatrix} (\bar{A}_{ij} - \bar{A}_{hj}) * F_{ih} \\ (\bar{a}_{ij} - \bar{a}_{hj}) * f_{ih} \end{bmatrix} = 0, \quad F_{ih} \neq \emptyset$$

B. PWQ Lyapunov Synthesis

By PWQ Lyapunov synthesis approach, we mean to find matrices P_i , vectors q_i and scalars r_i that for a constant positive scalar α , make $V(x)$ a PWQ Lyapunov function for the PWADI system above:

- (1) $V(x) = x^T P_i x + 2q_i^T x + r_i > 0 \quad x \in \sigma_j$
- (2) $dV/dt < -\alpha V(x) \quad x \in \sigma_j$

Problem 4.5: PWQ Synthesis (quadratic curve); For a constant positive scalar α , find P_i , q_i , r_i , K_i , k_i , Z_i and W_i that satisfy the following BMIs for $i, h = 1, 2, \dots, NR$ and $j = 1, 2$:

- (1) In regions R_i which contain the equilibrium point xcl

$$I: P_i \bar{A}_{ij} + \bar{A}_{ij}^T P_i + \alpha P_i < 0$$

$$II: Z_i > 0, W_i > 0, \begin{bmatrix} P_i & q_i \\ q_i^T & r_i \end{bmatrix} > 0$$
- (2) In regions R_i which do NOT contain the equilibrium point xcl

$$I: \begin{bmatrix} P_i \bar{A}_{ij} + \bar{A}_{ij}^T P_i + \alpha P_i + E_i^T Z_i E_i & * \\ (P_i \bar{a}_{ij} + E_i^T Z_i e_i + \bar{A}_{ij}^T q_i)^T & e_i^T Z_i e_i + 2q_i^T \bar{a}_{ij} \end{bmatrix} < 0$$

$$II: \begin{bmatrix} P_i - E_i^T W_i E_i & -E_i^T W_i e_i + q_i \\ (-E_i^T W_i e_i + q_i)^T & -e_i^T W_i e_i + r_i \end{bmatrix} > 0$$
- (3) Continuity of the Lyapunov function

$$\begin{bmatrix} F_{ih}^T (P_i - P_h) F_{ih} & F_{ih}^T (P_i - P_h) f_{ih} \\ (F_{ih}^T (P_i - P_h) f_{ih})^T & f_{ih}^T (P_i - P_h) f_{ih} \end{bmatrix} = 0, \quad F_{ih} \neq \emptyset$$
- (4) Continuity of the control outputs

$$\begin{bmatrix} (\bar{A}_{ij} - \bar{A}_{hj}) F_{ih} \\ (\bar{a}_{ij} - \bar{a}_{hj}) f_{ih} \end{bmatrix} = 0, \quad F_{ih} \neq \emptyset$$
 where F_{ih} and f_{ih} are defined by (6)

Similar to Problem 4.4 we can also use the ellipsoidal approximation in S-Procedure in Problem 4.5 to have:

Problem 4.6: PWQ Synthesis (ellipsoidal approximation); For a constant positive scalar α , find P_i , q_i , r_i , K_i , k_i , μ_i and β_i that satisfy the following BMIs for $i = 1, 2, \dots, NR$ and $j = 1, 2$:

- (1) In regions R_i which contain the equilibrium point xcl

$$I: P_i \bar{A}_{ij} + \bar{A}_{ij}^T P_i + \alpha P_i < 0$$

$$II: \mu_i < 0, \beta_i < 0, \begin{bmatrix} P_i & q_i \\ q_i^T & r_i \end{bmatrix} > 0$$

- (2) In regions R_i which do NOT contain the equilibrium point xcl

$$I: \begin{bmatrix} P_i \bar{A}_{ij} + \bar{A}_{ij}^T P_i + \alpha P_i + \mu_i E_i^T E_i & * \\ (P_i \bar{a}_{ij} + \mu_i E_i^T e_i + \bar{A}_{ij}^T q_i)^T & Y \end{bmatrix} < 0$$

where $Y = -\mu_i(1 - e_i^T e_i) + 2q_i^T \bar{a}_{ij}$

$$II: \begin{bmatrix} P_i - \beta_i E_i^T E_i & -\beta_i E_i^T e_i + q_i \\ (-\beta_i E_i^T e_i + q_i)^T & \beta_i(1 - e_i^T e_i) + r_i \end{bmatrix} > 0$$

- (3) Continuity of the Lyapunov function

$$\begin{bmatrix} F_{ih}^T (P_i - P_h F_{ih}) & F_{ih}^T (P_i - P_h) f_{ih} \\ (F_{ih}^T (P_i - P_h) f_{ih})^T & f_{ih}^T (P_i - P_h) f_{ih} \end{bmatrix} = 0, \quad F_{ih} \neq \emptyset$$

- (4) Continuity of the control outputs

$$\begin{bmatrix} (\bar{A}_{ij} - \bar{A}_{hj}) F_{ih} \\ (\bar{a}_{ij} - \bar{a}_{hj}) f_{ih} \end{bmatrix} = 0, \quad F_{ih} \neq \emptyset$$

C. PWA Controller Synthesis By PWATOOL

Using PWATOOL to design a PWA controller is easy. Suppose the variable *pwainc* contains a PWADI model whose convex hull contain a nonlinear model *nlsys*. Furthermore, assume the desired closed-loop equilibrium point is *xcl*.

Remark 4.7: Note that *xcl* can be any point within the domain of definition that satisfy the symbolic, algebraic equilibrium point equations. That is any point within the domain of definition that zeros the dynamics parametrically. PWATOOL starts the synthesis only if such symbolic equations can be satisfied.

The command to synthesize a PWA controller in PWATOOL is

```
pwasynt(pwainc, x0, setting);
```

Here *x0* is the initial point at which the final closed-loop system is simulated and *setting* is a structure containing the fields in Table II. If any of the fields is not assigned by the user, then PWATOOL will assign a default value to that parameter [10].

For example, if *setting* has been restricted as below

```
setting.ApxMeth='quadratic';
```

TABLE II
SETTING STRUCTURE

Fields	Description
alpha	decay rate
xcl	equilibrium point
ApxMeth	approximation method (ellipsoidal or quadratic)
Lyapunov	Lyapunov type (global or PWQ)
SynthMeth	synthesis method (BMI or LMI)
QLin	initial matrix Q
RLin	initial matrix R
RandomQ	choosing Q randomly (0 or 1)
RandomR	choosing R randomly (0 or 1)
NormalDirectionOnly	choosing the direction at which continuity constraints are calculated
StopTime	period of simulation in seconds (0 or 1)
IterationNumber	maximum number of iteration in case of the failure in converging
failure	indicates whether failure has happened before synthesis is done or stopped

then, PWATOOL uses global and PWQ Lyapunov approaches with quadratic curve approximation as BMI approaches as well as an LMI approach which is independent from above selection. This may yield up to three different PWA controllers for the system *pwainc*. Such stabilizing PWA controllers, as shown in [13], will stabilize the nonlinear function *nlsys*, contained in the convex hull of *pwainc*.

V. THE THEORY OF ANALYSIS; PWA AND NONLINEAR MODELS

PWATOOL analyzes a PWA or PWADI system by at most four approaches which are counterparts of Problems 4.3-4.6 where controller gains K_i and k_i are known, changing the BMIs in Section IV to LMIs. The user can again restrict PWATOOL to only certain approaches in checking the stability of the system. Since, PWATOOL checks only sufficient conditions for stability, it cannot conclude if a system is unstable. It rather shows if a system is stable. The command for checking the stability of a system in PWATOOL is

```
pwaaanalysis(pwainc, setting);
pwaaanalysis(pwainc, setting, gain);
```

where *setting* may be assigned any of the first five fields in Table II and *gain*, if passed to the function *pwaaanalysis*, would contain the PWA controllers in the aggregated vector space, i.e. $gain\{i\} = [K\{i\}, k\{i\}]$.

The point *setting.xcl* should satisfy the equilibrium point equations to be considered an equilibrium point in Analysis. These equations, as opposed to those in synthesis which are parametric, are rather algebraic equations with constant coefficients. The analysis proceeds only if *xcl* is verified to be an equilibrium point.

VI. EXAMPLE: ACTIVE FLUTTER SUPPRESSION

We study an airfoil dynamics with plunging and pitch degree of freedom [1] to show how a nonlinear system can be modeled by PWATOOL, analyzed and stabilized through a

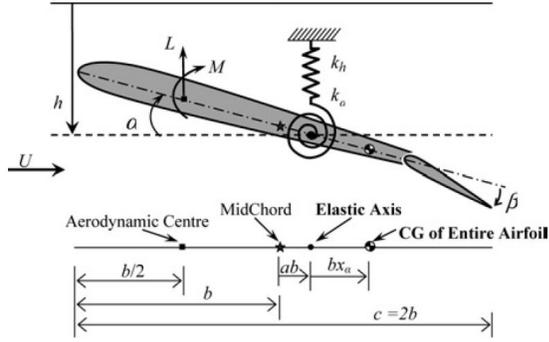


Fig. 2. Aeroelastic model[1]

TABLE III
AIRFOIL SYSTEM PARAMETERS

M	$= \begin{pmatrix} 12.3870 & 0.4180 \\ 0.4180 & 0.0650 \end{pmatrix}$	C_0	$= \begin{pmatrix} 27.43 & 0 \\ 0 & 0.036 \end{pmatrix}$
C_μ	$= \begin{pmatrix} 31.17 & 3.99 \\ 0.21 & -0.027 \end{pmatrix}$	K_0	$= \begin{pmatrix} 2844.4 & 0 \\ 0 & 0 \end{pmatrix}$
K_μ	$= \begin{pmatrix} 0 & 935.1 \\ 0 & -6.3 \end{pmatrix}$	B	$= \begin{pmatrix} -499.796 & -514.680 \\ -12.759 & -14.768 \end{pmatrix}$
μ	$= 176.609$		

PWA controller. This example has been solved on a machine with Windows XP operating system, Pentium 4 CPU and a memory of 1GB. The model dynamics of Figure 2 is described in (8):

$$\dot{x} = \begin{pmatrix} 0 \\ -M^{-1}(K_0 + K_u) \\ -M^{-1}(C_0 + C_u) \end{pmatrix} x + \begin{pmatrix} 0 \\ -M^{-1} \begin{pmatrix} 0 \\ x_2 K_\alpha(x_2) \end{pmatrix} \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ \mu M^{-1} \end{pmatrix} Bu \quad (8)$$

where $x = (x_1, x_2, x_3, x_4)^T$, $x_1 = h$, $x_2 = \alpha$, $x_3 = \dot{h}$, $x_4 = \dot{\alpha}$ and the nonlinear function $K_\alpha(\alpha)$ is defined as follows

$$K_\alpha(\alpha) = 2.82\alpha - 62.322\alpha^2 + 3709.71\alpha^3 - 24195.6\alpha^4 + 48756.954\alpha^5$$

Table III shows the parameters of the system. It is desired to stabilize the system around the open-loop equilibrium point $x = [0, 0, 0, 0]^T$.

A. Approximating The Nonlinear Model By PWADIs

We use `pwacreate` to create a nonlinear model for the airfoil model. The system is of order four and has two inputs. Therefore we type

```
pwacreate(4, 2, 'ActiveFlutter_non.m');
```

By doing so, an m-file called 'Flutter_Nonlinearity.m' will be created (if does not exist before) or rewritten (if existed before) in the current directory. This file will contain the complete dynamics of the systems. We fill the system dynamics as follow:

```
model.A = [0 0 1 0;
           0 0 0 1;
```

```
-M \ (K_0 + K_u) -M \ (C_0 + C_u)];
model.Bx = [0 0; 0 0; muu*inv(M)*B];
model.aff = [0;0;0;0];
model.fx = @Flutter_Nonlinearity;
model.xcl = [0;0;0;0];
model.Domain = {[ -1 1], [ -1 1], [ -1 1], [ -5 5]};
model.NonlinearDomain = [0;1;0;0];
model.mtd = 'Uniform';
model.NR = 4;
```

where `model.Domain` shows the domain of each variable x_i ($i = 1, 2, 3, 4$), `model.NonlinearDomain` shows which variable appears in the nonlinear dynamics described by the function handle `@Flutter_Nonlinearity`, `model.mtd` shows the type of gridding and `model.NR` the number of regions.

The nonlinear dynamics addressed by `@Flutter_Nonlinearity` has been coded as:

```
function F = Flutter_Nonlinearity(x)
```

```
qx2 = 2.82*x(2) - 62.322*x(2)^2 + 3709.71
      *x(2)^3 - 24195.6*x(2)^4
      + 48756.954*x(2)^5; |
```

```
M = [12.387 0.418; 0.418 0.065];
```

```
F = [0;0;-inv(M)*[0;qx2]];
```

We save the file 'ActiveFlutter_non.m' and run it. The data which we have entered will be checked for dimension consistency. A message is printed which shows the modeling has been accepted and the data has been stored as follows:

- `nlsys`: Original nonlinear model
- `pwainc`: PWADI approximation of the nonlinear system

We use `pwainc` for our analysis and design, however, in our simulation, we would use the nonlinear model `nlsys` above rather than the PWADI model `pwainc`. The stabilizing PWA controller which we obtain will stabilize the nonlinear model `nlsys` as shown in [13].

B. Synthesis A PWA Controller

The airfoil dynamics is an example of where suitable settings in synthesis improve the closed-loop system performance significantly. In the PWADI approximation, in previous section, we had split the domain of the second variable into four regions (`model.NR = 4`;). We first, synthesize a PWA controller for this approximation and later we show how it can be improved.

We use the following settings:

```
x0=[.4 .63 .5 1.2]';
setting.alpha=.1;
setting.SynthMeth='bmi';
setting.RandomQ= 0;
setting.RandomR= 0;
```

```
setting.QLin= 1.0e+003 *
[2.7378 1.9536 2.1565 1.5109
 1.9536 1.8326 1.5729 0.8522
 2.1565 1.5729 1.9228 1.2733
```

```
1.5109    0.8522    1.2733    1.1032];
```

```
setting.RLin= 1.0e+003 *
[2.7512    3.0551
 3.0551    3.3928];
```

Above setting imply PWATOOL would use only BMI methods, i.e. Problems 4.3-4.6, with the decay rate $\alpha = .1$. $QLin$ and $RLin$ are the matrices which are used for solving a *robust LQR* problem [5] and [13] in the equilibrium point region. $RandomQ = 0$ and $RandomR = 0$ indicate PWATOOL would not choose $QLin$ and $RLin$ randomly. The controller which are obtained in equilibrium point region is then extended to the whole regions to yield the PWA controller for the system. Therefore, we use the *pwasynt* command as follows

```
ctrl=pwasynt(pwainc, x0, setting);
```

PWATOOL converges in 19 seconds. Three sets of PWA controllers are obtained and stored in the variable *ctrl*. Let $\bar{K}_i = [K_i \ k_i]$ be the PWA controller gains in the aggregated vector space \mathbb{R}^{n+1} . Then, for instance, the PWA controller which is obtained by quadratic curve approximation and PWQ lyapunov function (as BMIs) is

$$\bar{K}_1 = \begin{pmatrix} -97.90 & -5576.60 & -81.80 & -76.20 & 19.00 \\ 88.70 & 5043.80 & 74.10 & 68.70 & 20.80 \end{pmatrix},$$

$$\bar{K}_2 = \begin{pmatrix} -97.90 & -5614.70 & -81.80 & -76.20 & 0 \\ 88.70 & 5002.30 & 74.10 & 68.70 & 0 \end{pmatrix},$$

$$\bar{K}_3 = \begin{pmatrix} -97.92 & 13.57 & -81.82 & -76.22 & -1.19 \\ 88.70 & -19.93 & 74.12 & 68.69 & 1.18 \end{pmatrix},$$

$$\bar{K}_4 = \begin{pmatrix} -97.92 & 11.20 & -81.82 & -76.22 & 0 \\ 88.70 & -17.56 & 74.12 & 68.69 & 0 \end{pmatrix}.$$

Large values of the gain in regions R_1 and R_2 will have bad effects on the closed-loop system response. The closed-loop system behavior, simulated automatically by PWATOOL, also confirms these effects. The error signals are shown in Figures 3. The controller output signals are also shown in Figures 4. It is seen that the signals are very large. Although this result have been obtained for certain initialization $QLin$ and $RLin$, similar results were obtained for any other initial values for $QLin$ and $RLin$ (assuming the synthesis converges) that we tried. This is, in part, because of the low resolution of PWADI approximation. We may be able to improve the closed-loop system behavior by improving our PWADI approximation.

To improve the approximation resolution, we increase the number of the regions to 6 in modeling and run the file 'ActiveFlutter_non.m' again:

```
model.NR=6;
run ActiveFlutter_non
```

This updates the PWADI approximation of the nonlinear system and should give us a better approximation. Again, we use the model *pwainc* for the synthesis. The fields *pwainc.E*

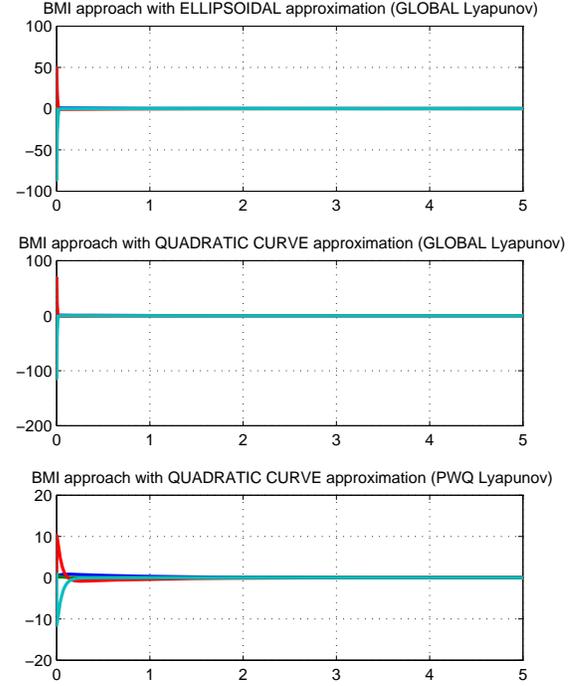


Fig. 3. Airfoil Dynamics Error Signals with Uniform gridding; $NR = 4$, $x_o = [4, .63, .5, 1.2]^T$

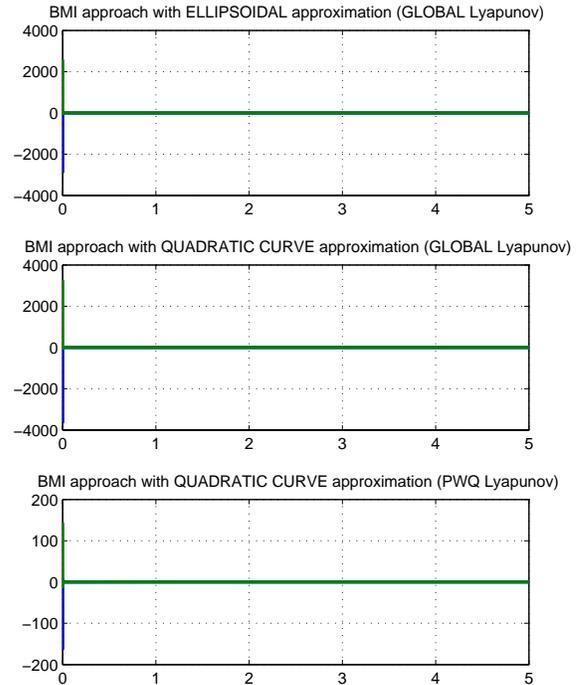


Fig. 4. Airfoil Dynamics Controller Output Signals with Uniform gridding; $NR = 4$, $x_o = [4, .63, .5, 1.2]^T$

and *pwainc.e* will specify the regions R_i ($i = 1, 2, \dots, 6$) when compared against (5):

$$\begin{aligned}
 R_1 &= \{x \mid -1 \leq x_2 \leq -.67\} \\
 R_2 &= \{x \mid 0 \leq x_2 \leq .33\} \\
 R_3 &= \{x \mid -.33 \leq x_2 \leq 0\} \\
 R_4 &= \{x \mid -.67 \leq x_2 \leq -.33\} \\
 R_5 &= \{x \mid .67 \leq x_2 \leq 1\} \\
 R_6 &= \{x \mid .33 \leq x_2 \leq .67\}
 \end{aligned} \tag{9}$$

Now, let us synthesize another PWA controller for the system. We keep our previous *setting* and we only change *QLin* and *RLin*:

```

setting.QLin= 1.0e+003 *
[2.0284    0.5278    0.7951    0.9888
 0.5278    0.5848    0.6961    0.5005
 0.7951    0.6961    1.7765    0.9525
 0.9888    0.5005    0.9525    0.9831];

```

```

setting.RLin= 1.0e+003 *
[0.1465    0.4483
 0.4483    1.3900];

```

We run the ‘*pwasynt*’ command again:

```
ctrl=pwasynt(pwainc, x0, setting);
```

PWATOOL converges in 26 seconds by three approaches. For instance, the PWA controller, in the aggregated vector space \mathbb{R}^{n+1} , which is obtained by quadratic curve approximation and PWQ lyapunov function (as BMIs) for regions R_i ’s in (9) are:

$$\begin{aligned}
 \bar{K}_1 &= \begin{pmatrix} -2.14 & 64.26 & -5.54 & -18.41 & 17.92 \\ 2.06 & -61.22 & 2.90 & 6.43 & -7.79 \end{pmatrix}, \\
 \bar{K}_2 &= \begin{pmatrix} -2.14 & -15.79 & -5.54 & -18.41 & 0 \\ 2.06 & 5.39 & 2.90 & 6.43 & 0 \end{pmatrix}, \\
 \bar{K}_3 &= \begin{pmatrix} -2.14 & -2.59 & -5.54 & -18.41 & 0 \\ 2.06 & -28.19 & 2.90 & 6.43 & 0 \end{pmatrix}, \\
 \bar{K}_4 &= \begin{pmatrix} -2.14 & 77.35 & -5.54 & -18.41 & 26.64 \\ 2.06 & -70.89 & 2.90 & 6.43 & -14.24 \end{pmatrix}, \\
 \bar{K}_5 &= \begin{pmatrix} -2.14 & 85.10 & -5.54 & -18.41 & -48.41 \\ 2.06 & -44.49 & 2.90 & 6.43 & 20.10 \end{pmatrix}, \\
 \bar{K}_6 &= \begin{pmatrix} -2.14 & 40.75 & -5.54 & -18.41 & -18.85 \\ 2.06 & -34.06 & 2.90 & 6.43 & 13.15 \end{pmatrix}.
 \end{aligned}$$

The controllers gains have decreased by at least one order of magnitude. Figures 5 and 6 show significant improvement in the system behavior and the controller output signals largeness.

C. Stability Analysis Of The System

The stability of the system, either open-loop or closed-loop, can be checked by the command *pwanalysis*. PWA-TOOL cannot comment on the stability of the open-loop

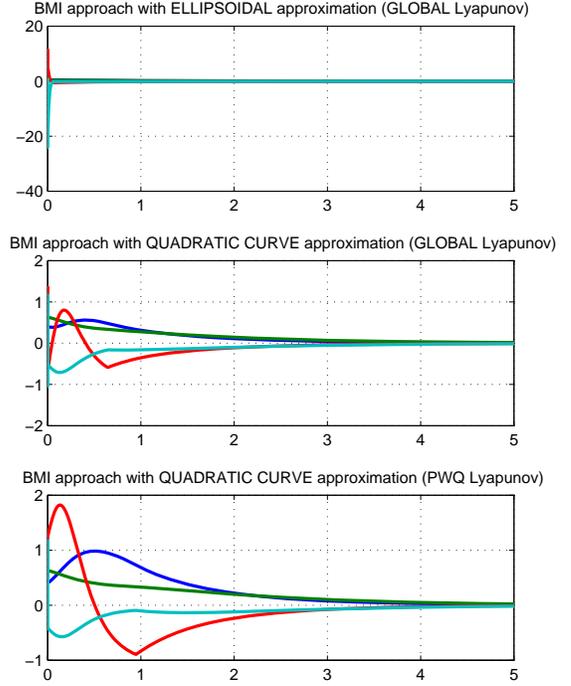


Fig. 5. Airfoil Dynamics Error Signals with Uniform gridding; $NR = 6$, $x_0 = [4, .63, .5, 1.2]^T$

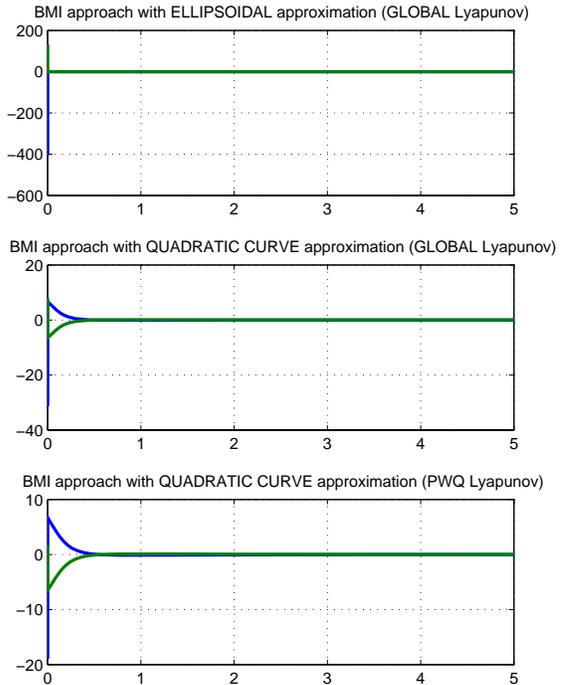


Fig. 6. Airfoil Dynamics Controller Output Signals with Uniform gridding; $NR = 6$, $x_0 = [4, .63, .5, 1.2]^T$

system in the Airfoil Dynamics Example as running the following line of code will show.

```
pwaanalysis(pwainc, setting);
```

where *setting* is a structure with the following fields and descriptions similar to those given in Table II.

- setting.alpha
- setting.xcl
- setting.Lyapunov
- setting.ApxMeth

Again, note that most of the fields above can be left unassigned so that they keep their current values from the synthesis or take default values which PWATOOL will assign to them. PWATOOL prints the following message

```
I could not verify if the open-loop
nonlinear system is stable at xcl.
```

The uncertainty in reasoning the stability of the system is either because of numerical issues or that the system is unstable. We check the closed-loop system (closed through either of the controllers which we obtained in Figure 6) by

```
for i=1:3
    Gain=ctrl.Gain{i};
    pwaanalysis(pwainc, setting, Gain);
end
```

Here, *ctrl* is the controller that we obtained in Section VI-B for $NR = 6$. The stability of the closed-loop system around $xcl = [0\ 0\ 0\ 0]^T$ is verified as PWATOOL prints a confirmation message:

```
The closed-loop nonlinear system is
stable at xcl.
```

VII. CONCLUSION AND FUTURE WORKS

We introduced PWATOOL which is a free, third-party MATLAB toolbox for the study of PWA systems in this paper. PWATOOL has an interface for helping the inexperienced user. It also has several commands, by which, an experienced user can analyze a PWA or nonlinear system or synthesize PWA controllers for it. PWATOOL can analyze a system by four Lyapunov-based methods which solve sufficient conditions for the stability of a system. It also synthesizes PWA controllers by five Lyapunov-based methods from which four methods use BMIs and one method uses LMI. PWATOOL future versions can extend to include output-feedback controllers as well as other different PWA controller synthesis which exist in the literature. One important and useful advancement to PWATOOL would be in terms of its graphical interface. The current structure of PWATOOL's commands and the possible interaction between analysis and synthesis modules would benefit a lot from such graphical interface improvement.

REFERENCES

- [1] S. Afkhami and H. Alighanbari. Nonlinear control design of an airfoil with active flutter suppression in the presence of disturbance. *Control Theory and Applications*, 1(6):1638–1649, 2007.
- [2] M. Zamani Fekri and L. Rodrigues. Extension of local linear controllers to global piecewise affine controllers: a convex approach. *To be submitted*, 2011.
- [3] An L_2 gain analysis of piecewise affine systems by piecewise quadratic storage functions. E. morinaga and k. hirata. *Proceedings of the American Control Conference*, 6:5176–5181, 30 June–2 July 2004.
- [4] A. Hassibi and S. Boyd. Quadratic stabilization and control of piecewise-linear systems. *Proceedings of the American Control Conference*, 6:3659 – 3664, 1998.
- [5] A. Jadbabaie. Robust, non-fragile controller synthesis using model based fuzzy systems: a linear matrix inequality approach. *M.Sc Thesis*, The University of New Mexico, New Mexico, Albuquerque, 1997.
- [6] E.C. Kerrigan and D.Q. Mayne. Optimal control of constrained, piecewise affine systems with bounded disturbances. 2:152–1557, 10–13 December 2002.
- [7] M. Lazar, W.P.M. Heemels, and A.R. Teel. Subtleties in robust stability of discrete-time piecewise affine systems. *Proceedings of American Control Conference*, pages 3464–3469, 9–13 July 2007.
- [8] C.A. Lehalle and R. Azencott. How piecewise affine neural networks can generate a stable nonlinear control. *Proceedings of the IEEE International Symposium on Intelligent Control/Intelligent Systems and Semiotics*, pages 23–28, 1999.
- [9] X. Li and Y. Cao. Projection-based piecewise-linear response surface modeling for strongly nonlinear vlsi performance variations. *9th International Symposium on Quality Electronic Design*, pages 108–113, 17–19 March 2008.
- [10] Manual. Pwatoool. <http://concordia.ca/hycons/pwatoool>, May 2011.
- [11] S.K. Nair and M. Jaekyun. Hardware-efficient nonlinear equalizer and its performance. *IEEE International Conference on Communications, Gateway to Globalization*, 1:653–657, Seattle 1995.
- [12] L. Rodrigues and S. Boyd. Piecewise-affine state feedback for piecewise-affine slab systems using convex optimization. *Systems and Control Letters*, 54:835–853, 2005.
- [13] B. Samadi and L. Rodrigues. Extension of local linear controllers to global piecewise affine controllers for uncertain non-linear systems. *International Journal of Systems Science*, 39(9):867–879, 2008.
- [14] Chin Kwan Thum, Chunling Du, B.M. Chen, Eng Hong Ong, and Kim Piew Tan. A unified control scheme for track seeking and following of a hard disk drive servo system. *IEEE Transactions on Control Systems Technology*, 18(2):294–306, 2010.
- [15] Toolbox. Checkmate. <http://www.ece.cmu.edu/webk/checkmate>, May 2011.
- [16] Toolbox. Ictools. <http://www.control.lth.se/project/ictools>, May 2011.
- [17] Toolbox. Phaver. <http://www-verimag.imag.fr/frehse/phaver-web/index.html>, May 2011.
- [18] Toolbox. Sostools. <http://www.cds.caltech.edu/sostools>, May 2011.
- [19] L. Vandenberghe, S. Boyd, and S.P. Wu. Determinant maximization with linear matrix inequality constraints. *SIAM Journal on Matrix Analysis and Applications*, 9(2):499–533, 1998.
- [20] Q. Zheng and F. Wu. Improving nonlinear control performance with a hybrid control strategy. *46th IEEE Conference on Decision and Control*, pages 3208–3213, 12–14 December 2007.