SYMBOLIC COMPUTATION ENVIRONMENT FOR NONLINEAR L_2 CONTROL. APPLICATION EXAMPLES

A. Consegliere, M.J. López

Dpto. Ingeniería de Sistemas y Automática. Universidad de Cádiz.
agustin.conseglieres@uca.es, manueljesus.lopez@uca.es

Keywords: Nonlinear H_∞ control, design methodology, symbolic computation, CACSD.

Abstract

In this work a computer based environment for symbolic calculation applied to nonlinear L_2 control is described. A very practical design and analysis methodology for nonlinear H_∞ control is presented. Starting from a model of the plant in affine form, and a state feedback controller (or state estimated feedback controller), a design procedure is established using nonlinear L_2 control theory. Functions $h_z(x)$, $k_{uz}(x)$ and parameter γ are used as design parameters. Due to the fact that exhaustive symbolic computation is necessary in nonlinear L_2 control, the Maple program has been used. The application is also based on Matlab and Simulink in order to implement functions from Control Toolboxes and simulation facilities. An additional advantage of the control theory used and the design procedure implemented in this work is that it can be applied to scalar and multivariable nonlinear systems. The design methodology and the software environment are tested by means of the simulations studies carried out with SISO, MISO, SIMO and MIMO systems.

1 Introduction

Design methods based on complex control technologies require, in general, high computation charge, which hinders control engineering training and use of those techniques in practice. For nonlinear control methods, such us L_2 or H_∞ control, symbolic computation tools are advisable (van der Schaft, 1996), (J.W. Helton, 1999). If linear theory is chosen, controller design is made using a linearized mathematical model of the plant for a given operation conditions, and conventional numerical calculation methods are sufficient. In case of nonlinear H_∞ control theory a linear approximation is not needed, and a nonlinear mathematical model of the system can be directly employed if this model satisfies some restrictions; for which suitable mathematical transformations are previously applied. For practical application, it is convenient that once the plant model is obtained in suitable form, a systematic procedure must be implemented for designer so that he can focus his attention in the engineering problem and not in mathematics. This has been our main motivation in this work. Some of the nonlinear control techniques that we have analyzed to get a general perspective have been: Passivity-based approach (R. Ortega, 1998), feedback linearization (Khalil, 1996), backstepping control (M. Krstic, 1995); and nonlinear H_∞ control (van der Schaft, 1996), (J.W. Helton, 1999). In this paper we describe a method to apply nonlinear H_∞ control theory and to obtain a robust nonlinear regulator for SISO and MIMO plants. Our design methodology and software (implemented using Maple and Matlab) are combined in a practical environment for the designer, which has a computer aided control system design (CACSD) package for nonlinear control systems analysis and controllers synthesis based on L_2 theory.

The paper is structured as follows: in paragraph two the nonlinear H_∞ control problem is described and the design methodology is presented, in paragraph three the main functions of the software environment are described, fourth paragraph presents application examples and simulations results for SISO, SIMO, MISO and MIMO systems; and finally conclusions are summarized.

2 Nonlinear H_∞ Control

In this section basic theory of non-linear L_2 state feedback controller is introduced (van der Schaft, 1992); (van der Schaft, 1996); (J.W. Helton, 1999).

We consider the mathematical model (Σ) of the ship to control and design specifications can be expressed in the following form, which is affine in w and u:

$$\begin{aligned} \dot{x} &= f(x) + g_w(x)w + g_u(x)u \\ z &= h_z(x) + k_{uz}(x)u \\ y &= h_y(x) + k_{wy}(x)w + k_{uy}(x)u \end{aligned}$$

The first equation describes a plant with state x, with control input u, and subject to exogenous inputs w. The second equation defines performance vector z, which may include tracking error component. The third equation defines measured output y. These equations define a *input-affine system*.

The L_2-norm of a signal $f(t)$ is defined as

$$||f||_2^2 = \int_0^\infty ||f(t)||^2 dt$$

where $||.||$ on the right hand side denotes the Euclidean norm of a vector. The system Σ has L_2-gain $\leq \gamma$, for $\gamma \geq 0$ if

$$||z||_2 \leq \gamma ||w||_2$$

for all $w \in L_2$, where w $||w||_2 < \infty$.

The goals of nonlinear H_∞ control are: 1) To achieve closed-loop asymptotic stability, 2) to yield the L_2 gain from exogenous inputs w to penalty variable z for the closed loop system less than some prescribed $\gamma \geq 0$.

Nonlinear State Feedback H_∞ Control

The H_∞ suboptimal control problem is simplified significantly if all states in the input affine system $\sum$ are available. If assumption A1) holds the system is modelled by

$$\dot{x} = f(x) + g_w(x)w + g_u(x)u \quad (1)$$

$$z = h_z(x) + k_{uz}(x) \quad (2)$$

Restricting the feedback to be a static state feedback

$$u = \alpha_x(x), \quad \alpha_x(x_0) = 0$$

the problem is referred to as the *state-feedback suboptimal H_∞ control problem*. The solution of this problem is closely related to the existence of a solution to a first order partial differential equation, which is called a *Hamiltom-Jacobi equation* (HJE)

$$V_x(x)f(x) + \frac{1}{2}V_x(x)m(x)V_x^T(x) + \frac{1}{2}h_z^T(x)h_z(x) = 0$$

where $m(x) = \left[\frac{1}{\gamma^2}g_w(x)g_w^T(x) - g_u(x)g_u^T(x)\right]$, $V_x \equiv \partial V/\partial x$.

If exists a solution $V \geq 0, \quad V(x_0) = 0$ to the HJE, then the closed-loop system for the feedback

$$u = l(x) = -g_u^T(x)V_x^T(x)$$

has L_2-gain from w to z less than or equal to γ. It can be demonstrated that this happens too if V is a solution to the Hamilton-Jacobi inequality (HJE where "$=$" is changed to "$\leq$").

In general it is impossible to find exact solutions to nonlinear partial differential equations, making some approximation scheme necessary. Different methods have been proposed in the literature to obtain approximate solutions of the HJE (Helton and James, 1999; van der Schaft, 1992), such as power series, successive approximations metod, use of Poisson series, using viscosity solutions to partial derivative equations, and nonlinear matrix inequalities. In this paper the problem is treated using power series expansions (van der Schaft, 1992, 1996; (Moller-Pedersen and Petersen, 1995). The design methodology has been implemented in a Matlab/Maple toolbox for H_∞ nonlinear control. Maple is used to solve the Hamilton-Jacobi equation, to obtain the linearized plant and to obtain the suboptimal control law. Matlab is used to solve Riccati algebraic equation, to improve numerical conditioning of the control system matrices and to carry out simulations in a realistic environment. Interface between Matlab and Maple is made by means of data files. The general design procedure consists of the following steps:

Step 1. Affine system structure (**??**, **??**, **??**) is obtained from the system mathematical model.

Step 2. The generalized plant is linearized for an equilibrium point, x_0, and the following linear representation is obtained

$$\Sigma_{lin}: \begin{aligned} \dot{x} &= Fx + G_w w + G_u u \\ z &= H_z x + K_{wz} w + K_{uz} u \\ y &= H_y x + K_{wy} w + K_{uy} u \end{aligned}$$

Step 3. A polynomic approximation of the HJE is obtained for the linearized system
$\frac{1}{2}V_x Fx + \frac{1}{2}x^T F^T V_x^T + \frac{1}{2}V_x\left[\frac{1}{\gamma^2}G_w G_w^T - G_u G_u^T\right]V_x^T +$
$+\frac{1}{2}x^T H_z^T H_z x \leq 0$ if $V(x) = \frac{1}{2}x^T X_\infty x$, the HJE is transformed into $x^T N x \leq 0$, where $N = \left(X_\infty F + F^T X_\infty + X_\infty\left[\frac{1}{\gamma^2}G_w G_w^T - G_u G_u^T\right]X_\infty + H_z^T H_z\right)$
this corresponds to Riccati algebraic equation (ARE), which is solved with Matlab using the Hamiltonian matrix

$$\begin{bmatrix} F & \frac{1}{\gamma^2}G_w G_w^T - G_u G_u^T \\ -H_z^T H_z & -F^T \end{bmatrix}$$

Step 4. X_∞ is substituted in

$$V(x) = \frac{1}{2}x^T X_\infty x$$

and this is the second order polynomic approximation of the HJE.

Step 5. With this $V(x)$, the following control law is obtained

$$\alpha_u(x) = -g_u^T(x)V_x^T(x)$$

Step 6. For obtaining nonlinear control laws, higher order approximations are calculated (van der Schaft, 1992).

Step 7. The nonlinear state feedback is obtained

$$\alpha_u(x) = -g_u^T(x)V_x^T(x)$$

3 Software environment

The software environment developed for nonlinear H_∞ control is based on Maple and Matlab. The basics elements and functions to carry out are:

1) Generation and manipulation of affine systems (affine form). This is composed by the following functions:

- Conversion to affine form.
- Structure for nonlinear affine plants.
- Linearization module.
- Structure for linear affine plants.
- Conversion to state space form.
- Conversion to transfer function.
- Affine representation of the nonlinear closed loop system.

2) Modelling functions. The following functions are considered:

- Affine models for actuators.
- Affine models for sensors.

- Affine models for noises and disturbances.
- Affine representation of design specifications (penalty or weighting functions).

3) Functions concerned with L_2 control problem

- Tests for DGKF (Doyle-Glover-Khargonekar-Francis) restrictions.
- Riccati equation (ARE) tests and solution.
- Set out the Hamilton-Jacobi Equation (HJE).
- Obtain an approximate solution of HJE (for order n).
- Calculate the feedback state nonlinear control law.
- Calculate the nonlinear observer.
- Obtain affine form of the nonlinear controller.

4) Controller robustness and performance analysis. In this module the following functions are implemented:

- Previous analysis of the linear system. Stability and performance. Time domain and frequency domain analysis.
- Closed loop system response simulation (nonlinear system).
- Temporary response parameters (rise time, overshoot, settle time, stationary error).
- Robustness and performance indicators with respect to: parameters variations, setpoint signals, disturbances and delays in the system.

5) Communication between Maple and Matlab interface.

Maple and Matlab functions

A toolbox for nonlinear H_∞ controller analysis and design has been developed. Next the main functions of this toolbox are indicated.

- The following variables, functions and parameters are defined: x, f, g_u, g_w, h_z, k_{uz}, k_{wz}, h_y, k_{wy}, k_{uy}, $x0$, $const$. For that, `x, f, gu, gw, hz, kuz, kwz, hy, kwy, kuy, x0` and `const` are employed.
- Plant model in affine form is made by means of `MakeSystem(x,f,gu,gw,hz,kuz,kwz,... hy,kwy,kuy,x0,const)`
- Affine representation of the system is shown with `DisplaySystem()`, and `DisplaySystem(ApplyConstants())` is used for particular values of the parameters.
- Actuators and sensors dynamics, disturbances and weighting functions (design parameters) are considered by means of the following instructions: `InputControlModel(), OutputMeasurementModel(), InputNoiseModel(), OutputPenaltyModel()`.
- Generalized plant or augmented plant with actuators, sensors and weighting functions (design parameters), is obtained with `GeneralizedPlant()`.
- Test for Doyle-Glover-Khargonekar-Francis (DGKF) conditions is made with `CheckDGKF()`.
- Hamilton-Jacobi equation (HJE) is outlined by means of the function `HJStateFunction()`
- By means of `StdLinealiza()` linearization of the generalized plant model is obtained. For particular values of the parameters `SimplifyLinSystem()` is used.
- The Hamilton-Jacobi equation for linearized system is solved combining Maple and Matlab functions. These are: `HJStateLinear(), ricschr.m, mapquest.m` and `mapquest2.m`.
- Second order approximation for $V(x)$ is obtained with `SolV2()` and `eval()` displays result. With this result linear state feedback controller is obtained.
- High order approximations (more than three) of the Hamilton-Jacobi Equation are calculated by means of `HJStaten()` and `HJState2()`.
- Nonlinear state feedback controller is obtained with `OptimalSF()`. While the worst possible disturbance is obtained with (from Differential Game Theory) `WorstDisturb()`.
- L_2 suboptimal controller is synthesized with the instruction `MakeController()`.
- Closed loop system (for a state feedback controller) is calculated using the function `ApplySFController()`.
- Nonlinear Estimated State Feedback Controller (NESFC) is obtained with the following functions: `FindIG()` and `FindIG2()`. In this case, the closed loop system in affine form is calculated with `ClosedLoop()`.

4 Application examples

In this paragraph results obtained for the following examples are summarized: 1) Robot link with flexible joint (simple-input simple-output, SISO), 2) Vehicle: Simple model of a car (multiple-input simple-output, MISO), 3) Inverted pendulum (simple-input multiple-output, SIMO), 4) Manipulator with three links (multiple-input multiple-output, MIMO). In all cases, the design parameters used are: h_z, k_{uz} and γ. The values employed for these parameters have been obtained by means of iterative procedure. Values given here have not been optimized, only have been chosen to get satisfactory results. If a more sophisticated procedure is employed, better performance can be obtained. The main objective of this paper is to show the validity of the design methodology and software application. Logically, for a particular design problem results can be improved, but our objective in this paper is not optimize performance neither improve results obtained with other techniques. The objective is to illustrate how the combination of the design methodology and software application give satisfactory results with reduced effort for the user.

1) Robot link with flexible joint

The dynamic equations of a single link robot arm with a revolute elastic joint rotating in a vertical plane are given by

$$\begin{aligned} J_m\ddot{q}_1 + F_m\dot{q}_1 - k(q_1 - q_2) &= u \\ J_l\ddot{q}_2 + F_l\dot{q}_2 + k(q_1 - q_2) - MgL\sin(q_2) &= 0 \\ y &= q_2 \end{aligned}$$

in which q_2 and q_1 are the link displacement and the rotor displacement, respectively. The link inertia J_l, the motor rotor inertia J_m, the elastic constant k, the link mass M, the gravity constant g the center of mass L and the viscous friction coefficients F_l, F_m are positive constant parameters. The control u is the torque delivered by the motor, and the controlled variable is q_2. The state vector has four components,

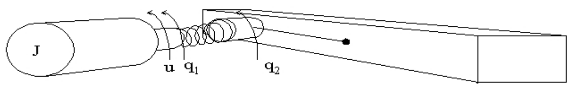

Figure 1: Robot link with flexible joint

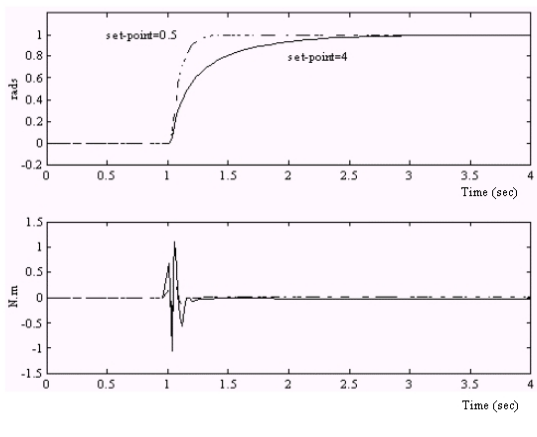

Figure 2: Normalized time responses for changes in setpoint (robot link with flexible joint)

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} \dot{q}_1 \\ q_1 \\ \dot{q}_2 \\ q_2 \end{bmatrix}$$

For this election, the functions of the affine representation of the systems are:

$$f = \begin{bmatrix} F_m\dot{q}_1 + \frac{k(q_2-q_1)}{J_m} \\ \dot{q}_1 \\ -\frac{MgL\sin(q_2)}{J_L} - \frac{k(q_2-q_1)}{J_L} - F_l\dot{q}_2 \\ \dot{q}_2 \end{bmatrix}$$

$$g_u = \begin{bmatrix} k/J_m \\ 0 \\ 0 \\ 00 \end{bmatrix},$$

$$g_w = g_u, \quad h_y = \begin{bmatrix} q_2 \\ q_1 \end{bmatrix}, \quad k_{uy} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$k_{wz} = k_{wy} = k_{uy}$$

The design is carried out with the following numerical values of the physical parameters (international system of units): $k = 100$, $J_l = 0.12$, $M = 1$, $Jm = 0.1$, $L = 0.3$, $g = 9.8$, $F_m = 0.01$, $F_l = 0.01$.

For this system, the following design parameters are used:

$$h_z = \begin{bmatrix} 10q_2 \\ 0 \end{bmatrix}, \quad k_{uz} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \gamma = 2$$

With these parameters and fourth order approximation for HJE, the following controller (third order) is obtained:

$$\begin{aligned} u(x) = [\quad & -14.858x_1 - 828.037x_2 - 0.366x_2^2x_4 \\ & -0.002x_2^2x_1 - 39.274x_3 - 0.066x_2x_3x_4 \\ & -0.001x_2x_3^2 - 1.125x_2x_4^2 - 0.003x_3^2x_4 \\ & -0.006x_4x_2x_1 - 0.014x_2^2x_3 \\ & -0.000x_3x_2x_1 - 0.000x_4x_3x_1 \\ & -0.134x_3x_4^2 - 0.000x_1^3 - 298.349x_4 \\ & -0.000x_3^3 - 0.000x_3x_1^2 - 3.111x_4^3 \\ & -0.008x_4^2x_1 - 0.055x_2^3 - 0.000x_2x_1^2 \\ & -0.000x_4x_1^2 - 0.000x_3^2x_1] \ \ 10^{-2} \end{aligned}$$

In figure 2 the normalized time responses for changes in set-point (0.5 and 4 radians respectively) are shown, for controlled variable (radians) and manipulated variable (Newton-m). Overshoot is avoid in both cases.

2) Vehicle: simple model of a car

It is considered the following simple model of a car,

$$\begin{aligned} \dot{x} &= \cos(\theta)u_1 \\ \dot{y} &= \sin(\theta)u_1 \\ \dot{\theta} &= \frac{1}{l}\tan(\phi)u_1 \\ \dot{\phi} &= u_2 \end{aligned}$$

Here (x, y) denote the Cartesian coordinates of the front axis,the angle θ measures the direction in which the car is headed, and ϕ is the angle made by the front axis with the car. There are two control inputs, u_1 denoting the driving velocity, and u_2 denoting the steering (rotational) velocity. In this case, u_1 is kept constant. Controlled variables are x, y. The follow-

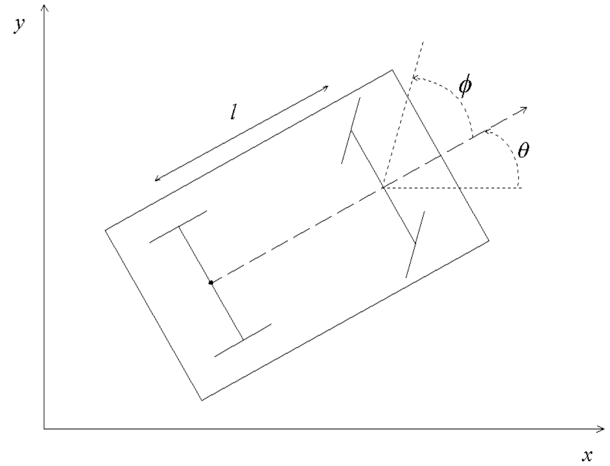

Figure 3: Vehicle: Simple model of a car

ing design parameters have been employed:

$$h_z = \begin{bmatrix} 0.1y \\ 0.1x \\ 0 \\ 0 \end{bmatrix}, \quad k_{uz} == \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \quad \gamma = 5$$

In figure 4 the reference and trajectory followed by the car are shown. In this case, a fourth order approximation is employed for HJE.

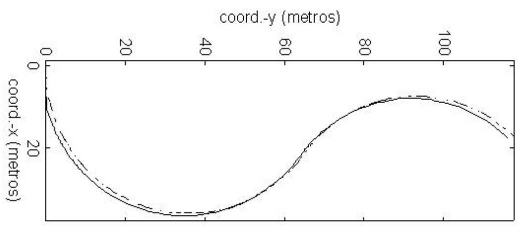

Figure 4: Trajectory following of the car

3) Inverted pendulum (SIMO)

The dynamic equations of an inverted pendulum on a moving cart are given by

$$\begin{aligned} (M+m)\ddot{x} + mL\cos(\theta)\ddot{\theta} - mL\sin(\theta)\dot{\theta}^2 &= u_1 \\ mL\cos(\theta)\ddot{x} + mL^2\ddot{(\theta)} - mgL\sin(\theta) &= u_2 \end{aligned}$$

in which θ is the angle displacement of the pendulum from the vertical configuration, x is the position of the cart, L is the length of the pendulum, M is the mass of the cart, m is the point mass attached at the end of the pendulum, g is the gravity constant. The input u is the force applied to the cart while the input u_2 is the torque applied at the base of the pendulum. Assuming the states (x, θ) are measured, (u_1, u_2) is to be designed in order to track $x_r(t), \theta_r(t)$, in particular $\theta_r(t) = 0$ and $u_2 = 0$.

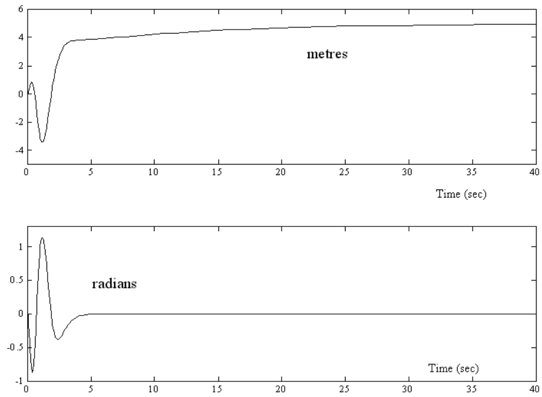

Figure 5: Normalized time responses for changes in setpoint (robot link with flexible joint)

For this system, the following design parameters are used:

$$h_z = \begin{bmatrix} 10\theta \\ 0.1x \\ 0 \end{bmatrix}, \quad k_{uz} == \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad \gamma = 4$$

Linear position of the cart (meters) and angular position of the inverted pendulum (radians) are shown in figure 5, for a setpoint change of 5 meters. The controller is obtained with a third order approximation for HJE.

4) Manipulator with three links (MIMO)

The dynamical model for this manipulator is given by,

$$M\ddot{q}_L + C\dot{q}_L + D\dot{q}_L^2 + E\dot{q}\dot{q}_L + Fg = K_L u$$

where the parameter matrices D, E, F and M are not constant, but are dependent upon trigonometric functions of variables q_i ($i = 1, 2, 3$). Physical meaning and notation use for the vectors and matrices are as follow: M (inertia matrix), C (friction matrix), $\ddot{q}_L$ (acceleration vector), $\dot{q}_L$ (velocity vector), $D\dot{q}_L^2$ (centrifugal forces), $E\dot{q}\dot{q}_L$ (Coriolis forces), Fg (gravity forces), K_L (gain matrix), u control input vector.

For this system the following design parameters have been used:

$$h_z = \begin{bmatrix} q_1 \\ q_2 \\ q_3 + \dot{q}_3 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad k_{uz} == \begin{bmatrix} 0_{3\times3} \\ I_{3\times3} \end{bmatrix}, \quad \gamma = 2$$

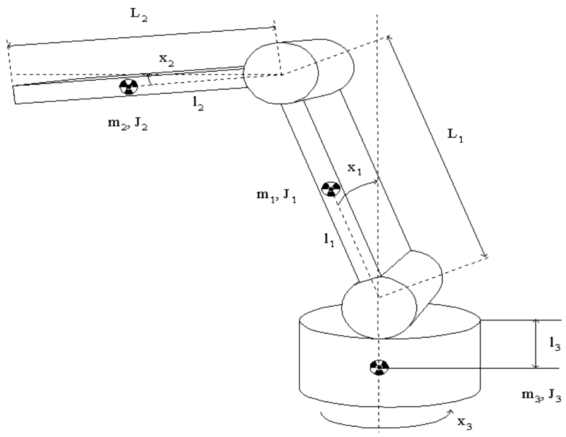

Figure 6: Diagram of manipulator

In figure 7 the temporary response to setpoints change are shown, for 1.5, 0.5 and 1 radians respectively. For all articulations zero or low overshoot is obtained, and null stationary error.

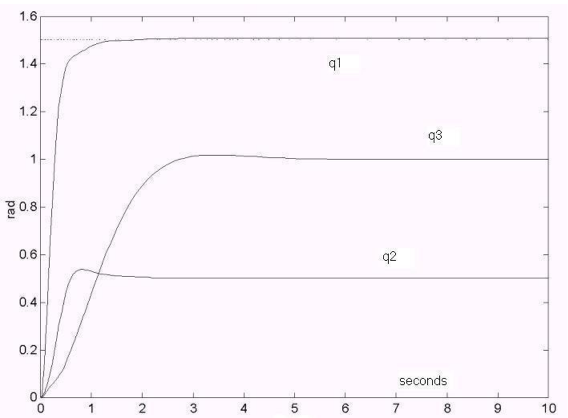

Figure 7: Time response for MIMO system

5 CONCLUSIONS

A software application has been developed for implementing a method based on nonlinear L_2 control theory (state feedback and estimated state feedback). Our control methodology employs as design parameters two functions (h_z, k_{uz}) and a scalar (γ), which are used for tuning controller to satisfactory time response. Design method requires input affine model of the plant, and it is applicable to scalar and multivariable systems. In order to test the validity of the proposed methodology, simulation studies have been carried out for SISO, MISO, SIMO and MIMO systems; for which simulation results are suitable in a first step of design. Better results can be obtained if more complex design functions are used, but our main contribution has been to facilitate to control engineers and to students a computer aided control system design (CACSD) package for nonlinear control systems analysis and controllers synthesis based on L_2 theory.

References

J.W. Helton, M.R. James (1999). *Extending H_∞ Control to Nonlinear Systems*. SIAM.

Khalil, H. K. (1996). *Nonlinear Systems*. MacMillan.

M. Krstic, I. Kanellakopoulos, P. Kokotovic (1995). *Nonlinear and Adaptive Control Design*. John Wiley and Sons.

Moller-Pedersen, J. and M. P. Petersen (1995). *Control of Nonlinear Plnats*. Institute of Automation and Mathematical Institute. Technical University of Denmark.

R. Ortega, A. Loira, P.J. Niclasson H. Sira-Ramirez (1998). *Passivity-based Control of Euler-Lagrange Systems*. Springer.

van der Schaft, A. (1992). l_2-gain analysis of nonlinear systems and nonlinear state feedback h_∞ control. *IEEE Transactions on Automatic Control*.

van der Schaft, A. (1996). *L_2-Gain and Passivity Techniques in Nonlinear Control*. Springer.