

Mauricio Becerra-Vargas

maubeus@sc.usp.br  
University of São Paulo - EESC  
Department of Aeronautical Engineering  
13560-970 São Carlos, SP, Brazil

Eduardo Morgado Belo

belo@sc.usp.br  
University of São Paulo - EESC  
Department of Aeronautical Engineering  
13560-970 São Carlos, SP, Brazil

# Application of $H_\infty$ Theory to a 6 DOF Flight Simulator Motion Base

*The purpose of this study is to apply inverse dynamics control for a six degree of freedom flight simulator motion system. Imperfect compensation of the inverse dynamic control is intentionally introduced in order to simplify the implementation of this approach. The control strategy is applied in the outer loop of the inverse dynamic control to counteract the effects of imperfect compensation. The control strategy is designed using  $H_\infty$  theory. Forward and inverse kinematics and full dynamic model of a six degrees of freedom motion base driven by electromechanical actuators are briefly presented. Describing function, acceleration step response and some maneuvers computed from the washout filter were used to evaluate the performance of the controllers.*

**Keywords:** inverse dynamics control,  $H_\infty$  theory, flight simulator motion base, Stewart platform

## Introduction

Most flight simulator adopt the Stewart platform as the motion base, which is composed of a moving platform linked to a fixed base through six extensible legs. Each leg is composed of a prismatic joint (i.e an electromechanical or electrohydraulic actuator), one passive universal joint and one passive spherical joint making connection with the base and the moving platform, respectively, as shown in Fig. 1.

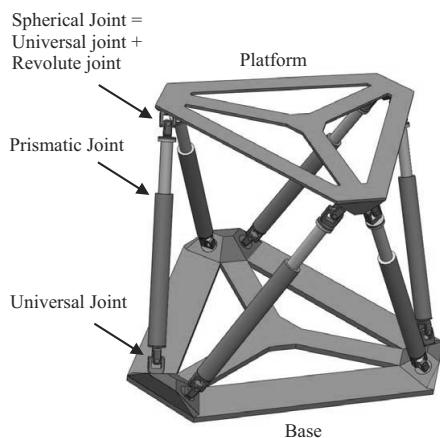


Figure 1. The Stewart platform.

Most motion control schemes concerning flight simulator motion bases are focused on the washout-filter (Nahon and Reid, 1990), forward and inverse kinematics; and an independent joint linear controller is implemented for each actuator (Salcudean et al., 1994; Graf et al., 1998). On the other hand, the effects of the motion-base dynamics are ignored or a linearized model of motion-base dynamics is used (Idan and Sahar, 1996).

Inverse dynamics control (Sciavicco and Siciliano, 2005; Spong and Vidyasagar, 2006) is an approach to nonlinear control design whose central idea is to construct an inner loop control based on the motion base dynamic model which, in the ideal case, exactly linearizes the nonlinear system and an outer loop control to drive tracking errors

to zero. This technique is based on the assumption of exact cancellation of nonlinear terms. Therefore, parametric uncertainty, unmodeled dynamics and external disturbances may deteriorate the controller performance. In addition, a high computational burden is paid by computing on-line the complete dynamic model of the motion-base (Koekebakker, 2001). Robustness can be regained by applying robust control techniques in the outer loop control structure as is shown in Becerra-Vargas et al. (2009).

In this context, this work presents the application of a control strategy applied in the outer loop of the feedback linearized system for robust acceleration tracking in the presence of parametric uncertainty and unmodeled dynamics, which is intentionally introduced in the process of approximating the dynamic model in order to simplify the implementation of this approach.

The forward and inverse kinematics and the dynamic model of six degrees of freedom motion base are briefly presented. Then, electromechanical actuator dynamics are included in order to obtain a full dynamic model.

The control strategy consists in introducing an additional term to the inverse dynamics controller which provides robustness to the control system. The robust control term is designed for a disturbance rejection problem via  $H_\infty$  control, where the controller considers the uncertainties as disturbances affecting the linearized system.

Finally, standard methods to characterize the performance of a flight simulator motion base are presented and used to evaluate the performance of the controller.

This paper is structured as follows: in Section II, the forward and inverse kinematics and dynamic model of six degrees of freedom motion base are briefly presented. Then, electromechanical actuator dynamics are included in order to obtain a full dynamic model; in Section III, the control structure in cartesian coordinates is presented and the control strategy is applied. Then, dynamic model matrices that will be used in the controller are defined; in Section IV, three methods to evaluate the controllers's performance are defined; in Section V the results obtained from simulation are shown and discussed; and finally, in Section VI, conclusions of the present work are discussed.

## Nomenclature

$\mathbf{a}$	= acceleration vector of the moving platform centre of gravity
$\mathbf{b}_i$	= position vector of the $i$ -th base point
$\mathbf{e}$	= Cartesian coordinates tracking error vector
$F_i$	= driving force generated by the $i$ -th electromechanical actuator
$\mathbf{g}$	= acceleration vector due to gravity
$\mathbf{I}_p$	= moment of inertia matrix of the moving platform
$\mathbf{J}_{l,\omega}$	= Jacobian matrix relating to $\boldsymbol{\omega}$ and $\dot{L}_i$
$\mathbf{J}_{l,q}$	= Jacobian matrix relating to $\dot{L}_i$ and $\dot{\mathbf{q}}$
$L_i$	= length of the $i$ -th leg
$M$	= mass of the moving platform (including payload)
$\widetilde{\mathbf{M}}$	= simplified representation of the matrix $\mathbf{M}$
$\widetilde{\mathbf{N}}$	= simplified representation of the matrix $\mathbf{N}$
$\mathbf{p}_i$	= position vector of the $i$ -th platform point (in moving platform frame)
$\mathbf{q}$	= Cartesian space coordinates vector
$\mathbf{q}_n$	= platform neutral position
$(\mathbf{q}_p)_i$	= $\Re \mathbf{p}_i$
$(\tilde{\mathbf{q}}_p)_i$	= skew-symmetric matrix associated to $(\mathbf{q}_p)_i$
$\mathbf{Q}_i$	= matrix depending on the $i$ th leg inertial properties
$\mathbf{R}$	= position vector of the centre of gravity
$\widetilde{\mathbf{R}}$	= skew-symmetric matrix associated with $\mathbf{R}$
$\Re$	= rotation matrix of moving platform frame relative to base platform frame
$\mathbf{s}_i$	= unit vector along the direction of the $i$ -th leg
$\mathbf{t}$	= translation vector of the upper platform centroid, $(x, y, z)$
$\mathbf{u}$	= robust control vector
$\mathbf{V}_i$	= vector depending on the $i$ -th leg dynamic properties
$\mathbf{w}$	= uncertainty vector

## Greek Symbols

$\Theta$	= Euler angles vector, $(\phi, \theta, \psi)$
$\boldsymbol{\omega}$	= angular velocity vector of the moving platform
$\omega_b$	= bandwidth of the flight simulator motion base
$\boldsymbol{\alpha}$	= angular acceleration vector of the moving platform
$\tau_i$	= driving torque generated by the $i$ th electromechanical actuator

## Subscripts

$i$	= Relative to actuator $i$
-----	----------------------------

## Superscripts

$\tau$	= Represents transposition
--------	----------------------------

## Motion Base Kinematics and Dynamics

The Newton-Euler approach (Dasgupta and Mruthyunjaya, 1998) was adopted to calculate the Stewart platform nonlinear dynamic model in cartesian coordinates. The cartesian space coordinates  $\mathbf{q}$  are defined as

$$\mathbf{q} = [\mathbf{t}^T \quad \Theta^T]^T, \quad (1)$$

where  $\mathbf{t} = [x \ y \ z]^T$  is the translation vector of the moving platform centroid and  $\Theta = [\phi \ \theta \ \psi]^T$  is the Euler angles vector defining its orientation.

The leg vector with respect to the inertial base frame  $\{\mathbf{B}\}$ , as shown in Fig. 2, can be denoted as

$$\mathbf{S}_i = \Re \mathbf{p}_i + \mathbf{t} - \mathbf{b}_i \quad (2)$$

Equation (2) represents the inverse kinematics problem in the sense one can compute the legs' lengths, i.e., norms of  $\mathbf{S}_i$ , from the given position ( $\mathbf{t}$ ) and orientation ( $\Re$  being function of  $\Theta$ ) of the platform.

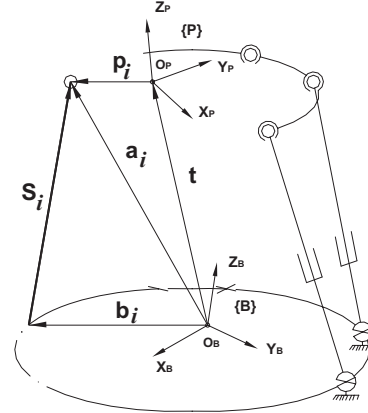


Figure 2. UPS Stewart platform.

The forward kinematics problem comprises the determination of the position and orientation of the platform from the given actuator lengths. It is observed from Eq. (2) that the forward kinematics problem involves solving six simultaneous nonlinear equations for the values of the six unknown variables representing the position and orientation of the platform. Consequently, iterative numerical methods are employed to solve the set of nonlinear equations (Nguyen et al., 1993).

Kinematic analysis of the legs can be derived from Eq. (2). Dynamic analysis of the legs can be derived taking force and moment balance of each leg (left side of Figure 3) and then combining both equations. Thus, one gets

$$(\mathbf{F}_s)_i = \mathbf{Q}_i \ddot{\mathbf{t}} - \mathbf{Q}_i (\tilde{\mathbf{q}}_p)_i \boldsymbol{\alpha} + \mathbf{V}_i - F_i \mathbf{s}_i \quad (3)$$

where  $(\mathbf{F}_s)_i$  is the force applied to the leg  $i$  by the upper platform,  $\mathbf{Q}_i$  depends on the  $i$ th leg inertial properties and  $\mathbf{V}_i$  depends on the  $i$ th leg dynamic properties.

Similarly, force and moment balance on the moving platform (right side of Figure 3) can be written as (with no external forces):

$$M \mathbf{a} = M \mathbf{g} - \sum_{i=1}^6 (\mathbf{F}_s)_i \quad (4)$$

and

$$M \mathbf{R} \times \mathbf{g} - \sum_{i=1}^6 [(\mathbf{q}_p)_i \times (\mathbf{F}_s)_i] + \sum_{i=1}^6 \mathbf{f}_i = \mathbf{I}_p \boldsymbol{\alpha} + \boldsymbol{\omega} \times \mathbf{I}_p \boldsymbol{\omega} + M \mathbf{R} \times \mathbf{a}; \quad (5)$$

where  $\mathbf{f}_i$  is the moment of viscous friction at the  $i$ -th spherical joint.

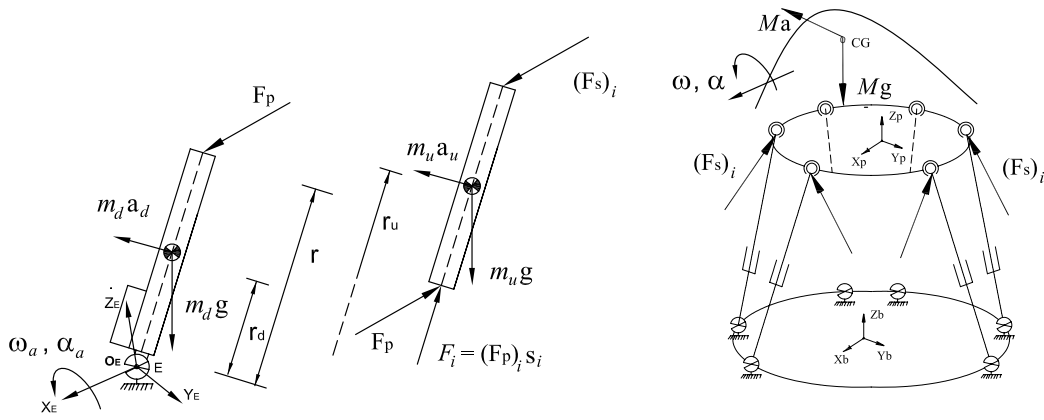


Figure 3. Vector analysis - Motion base.

Substituting the expression for  $(\mathbf{F}_s)_i$  from Eq. (3) into Eqs. (4) and (5) and then combining them, one gets

$$\mathbf{M}_t(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}_t(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{B}_t(\dot{\mathbf{q}}) + \mathbf{G}_t(\mathbf{q}) = \mathbf{J}_{l,\omega}\mathbf{F} \quad (6)$$

where:

$$\mathbf{M}_t = \mathbf{M}_p + \mathbf{M}_a; \quad \mathbf{C}_t = \mathbf{C}_p + \mathbf{C}_a; \quad \mathbf{G}_t = \mathbf{G}_p + \mathbf{G}_a;$$

and

$$\mathbf{F} = [ F_1 \quad F_2 \quad F_3 \quad F_4 \quad F_5 \quad F_6 ]^T.$$

The detailed elements of the above matrices are given in the Appendix. Information about derivation of the Stewart platform's dynamic model is not detailed here since it is not the scope of this paper.

### 0.1 Inclusion of actuator dynamics

With the improvement in electrical servo actuation technology, there is a trend to use electrically driven motion systems instead of those hydraulically driven.

An electromechanical servo-actuator, shown in Figure 4, consists of a servo-motor and an actuator driven by a servo-drive. As the closed loop bandwidth between the servo-drive and the servo-motor is much higher than that of the motion system, electrical dynamics can be omitted and the mechanical dynamics is the only one considered. Thus, the motor torque can be considered proportional to the motor current.

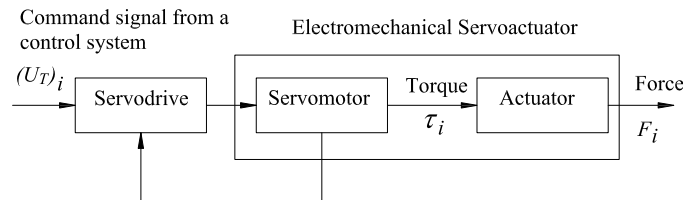


Figure 4. Representation of the electromechanical servo-actuator.

The equation of motion of the electromechanical actuator (Figure 5) can be written in matrix form:

$$\mathbf{F} = \mathbf{K}_a \mathbf{T}_m - \mathbf{D}_a \ddot{\mathbf{L}} - \mathbf{B}_a \dot{\mathbf{L}}; \quad (7)$$

where

$$\mathbf{T}_m = [ \tau_1 \quad \tau_2 \quad \tau_3 \quad \tau_4 \quad \tau_5 \quad \tau_6 ]^T; \\ \mathbf{L} = [ L_1 \quad L_2 \quad L_3 \quad L_4 \quad L_5 \quad L_6 ]^T;$$

and where  $\mathbf{D}_a$  are the actuator inertia matrix,  $\mathbf{B}_a$  is the actuator viscous damping coefficient matrix, and  $\mathbf{K}_a$  is the actuator gain matrix, which are detailed in the Appendix.

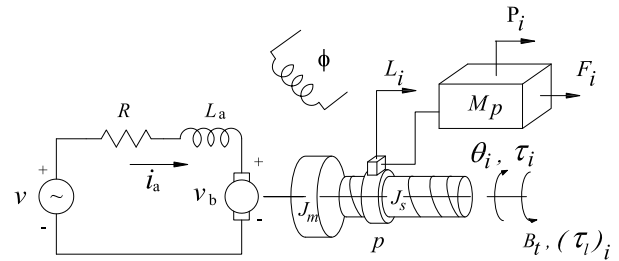


Figure 5. Electromechanical actuator.

The relationship between the cartesian coordinates and joint coordinates can be written as

$$\dot{\mathbf{L}} = \mathbf{J}_{l,q} \dot{\mathbf{q}} \\ \ddot{\mathbf{L}} = \mathbf{J}_{l,q} \ddot{\mathbf{q}} + \dot{\mathbf{J}}_{l,q} \dot{\mathbf{q}} \quad (8)$$

Substituting Eq. (8) into Eq. (7), one gets

$$\mathbf{F} = \mathbf{K}_a \mathbf{T}_m - \mathbf{D}_a \mathbf{J}_{l,q} \ddot{\mathbf{q}} - \mathbf{D}_a \dot{\mathbf{J}}_{l,q} \dot{\mathbf{q}} - \mathbf{B}_a \mathbf{J}_{l,q} \dot{\mathbf{q}} \quad (9)$$

And substituting Eq. (9) into Eq. (6), the full dynamic model in cartesian coordinates results as

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{N}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{u}_T \quad (10)$$

where

$$\mathbf{N} = \mathbf{C} + \mathbf{E} + \mathbf{G} \\ \mathbf{M} = \mathbf{K}_a^{-1} [ \mathbf{J}_{l,\omega}^{-T} \mathbf{M}_t + \mathbf{D}_a \mathbf{J}_{l,q} ] \\ \mathbf{C} = \mathbf{K}_a^{-1} [ \mathbf{J}_{l,\omega}^{-T} \mathbf{C}_t + \mathbf{D}_a \dot{\mathbf{J}}_{l,q} + \mathbf{B}_a \mathbf{J}_{l,q} \dot{\mathbf{q}} ] \\ \mathbf{E} = \mathbf{K}_a^{-1} \mathbf{J}_{l,\omega}^{-T} \mathbf{B}_t \\ \mathbf{G} = \mathbf{K}_a^{-1} \mathbf{J}_{l,\omega}^{-T} \mathbf{G}_t$$

and  $\mathbf{u}_T = \mathbf{T}_m$ , which is proportional to the motor current (not considering the actuator electrical dynamics) driving the servo-drive of the electromechanical actuator.

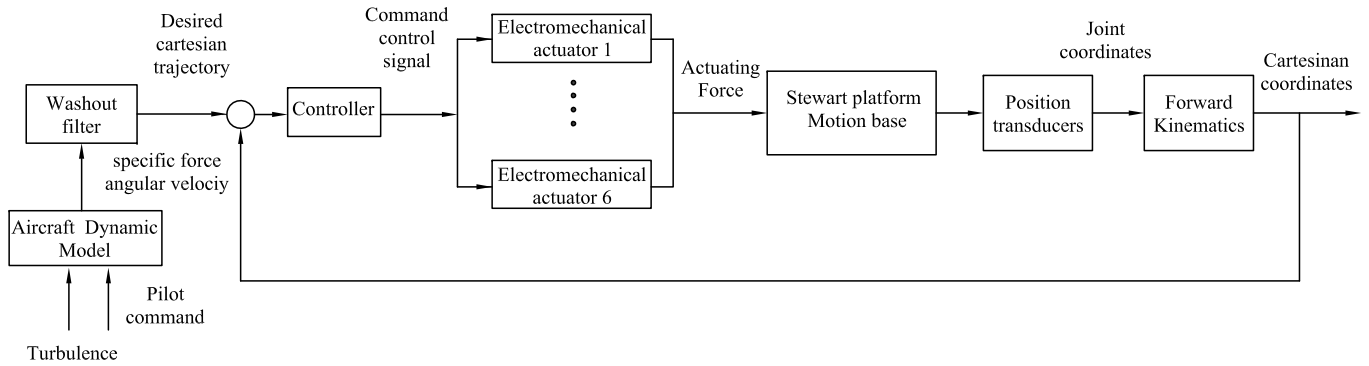


Figure 6. Cartesian space control framework.

## 1. Motion Base Controller Design

A flight simulator control system stems from two frameworks. The first one is based on the legs lengths tracking control and is called joint space control (Salcudean et al., 1994; Graf et al., 1998). On the contrary, the second one is based on the position and orientation tracking control and is called cartesian space control (Becerra-Vargas et al., 2009; Koekebakker, 2001).

Joint space control does not seem suitable in inverse dynamic control due to the fact that joint space dynamic equations are more complicated compared with the cartesian ones in Eq. (10), moreover, the terms of the joint space dynamic matrices will still depend on the cartesian coordinates.

Cartesian space control was adopted in this study as shown in Fig. 6. The pilot responds to the simulator cues and tracking or disturbance tasks by driving the aircraft control surfaces, then aircraft's response is calculated through an aircraft dynamic model.

Because of the limited motion envelope of the motion base, filtering (by washout filter (Nahon and Reid, 1990)) is required between the calculated aircraft trajectories and the commanded motion base trajectories. Then, the controller attempts to null the cartesian coordinate error by commanding a torque signal (voltage or current) to the servo-drive of the electromechanical actuator.

Thus, the force driving the motion base is governed by the equation of motion of the electromechanical actuator in Eq. (9).

Cartesian space control needs information of a 6 degrees of freedom sensor to measure the position and orientation of the platform. However, when only legs lengths measurements are available, the forward kinematic problem must be resolved.

### 1.1 Imperfect compensation of the inverse dynamics control

As it is well known, inverse dynamics control (Sciavicco and Siciliano, 2005; Spong and Vidyasagar, 2006) is an approach to nonlinear control design whose central idea is to construct an inner loop control which, in the ideal case, exactly linearizes the nonlinear system and an outer loop control to drive tracking errors to zero. The global linearization of the motion base dynamics can be obtained by the inverse dynamics control

law:

$$\mathbf{u}_T = \mathbf{M}(\mathbf{q})\mathbf{v} + \mathbf{N}(\mathbf{q}, \dot{\mathbf{q}}) \quad (11)$$

where  $\mathbf{M}(\mathbf{q})$  and  $\mathbf{N}(\mathbf{q}, \dot{\mathbf{q}})$  represent the matrices in Eq. (10) and:

$$\mathbf{v} = \ddot{\mathbf{q}}_d + \mathbf{K}_d \dot{\mathbf{e}} + \mathbf{K}_p \mathbf{e}, \quad (12)$$

and where  $\mathbf{K}_d$  and  $\mathbf{K}_p$  represent the gain matrices and the tracking error is defined as

$$\mathbf{e} = \mathbf{q}_d - \mathbf{q}, \quad (13)$$

where  $\mathbf{q}_d$  is the desired cartesian space coordinates. Now, substituting Eq. (11) into Eq. (10), and simplifying it, leads to the system of second-order equation:

$$\ddot{\mathbf{e}} + \mathbf{K}_d \dot{\mathbf{e}} + \mathbf{K}_p \mathbf{e} = 0, \quad (14)$$

where asymptotic stability is reached by choosing the matrices  $\mathbf{K}_d$  and  $\mathbf{K}_p$  as:

$$\begin{aligned} \mathbf{K}_p &= \text{diag} \{ \omega_{n1}^2, \dots, \omega_{n6}^2 \}; \\ \mathbf{K}_d &= \text{diag} \{ 2\zeta_1 \omega_{n1}, \dots, 2\zeta_6 \omega_{n6} \} \end{aligned} \quad (15)$$

where  $\omega_{ni}$  and  $\zeta_i$  characterize the response of the tracking error in Eq. (14)

Practical implementation of the inverse dynamics control law in Eq. (11) requires the parameters of the matrices  $\mathbf{M}(\mathbf{q})$  and  $\mathbf{N}(\mathbf{q}, \dot{\mathbf{q}})$  to be accurately known, the matrices to be modeled accurately and to be computed in real-time. These requirements are difficult to satisfy in practice. Model imprecision may come from parametric uncertainties and purposeful choice of a simplified representation of the matrices  $\mathbf{M}(\mathbf{q})$  and  $\mathbf{N}(\mathbf{q}, \dot{\mathbf{q}})$  (unmodeled dynamics intentionally introduced). In other words, there will be always inexact cancellation (imperfect compensation) of the nonlinearities in the system due to these uncertainties also the burden of computing these matrices at each sample instant. Therefore, the control law in Eq. (11) can be rewritten as (Figure 7)

$$\mathbf{u}_T = \widehat{\mathbf{M}}(\mathbf{q})\mathbf{v} + \widehat{\mathbf{N}}(\mathbf{q}, \dot{\mathbf{q}}); \quad (16)$$

where the term  $\mathbf{v}$  is modified as:

$$\mathbf{v} = \ddot{\mathbf{q}}_d + \mathbf{K}_d \dot{\mathbf{e}} + \mathbf{K}_p \mathbf{e} + \mathbf{u}, \quad (17)$$

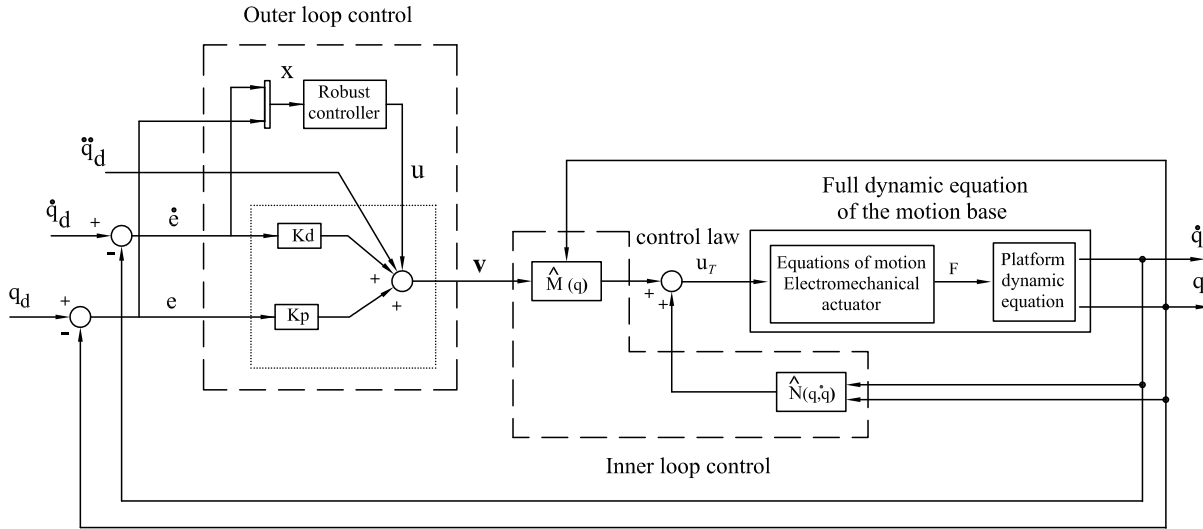


Figure 7. Inverse dynamic control, imperfect compensation.

and  $\widehat{\mathbf{M}}$ ,  $\widehat{\mathbf{N}}$  represent simplified versions of  $\mathbf{M}$ ,  $\mathbf{N}$  and are defined in Section III-C. The term  $\mathbf{u}$  in Eq. (17) is included to overcome imperfect compensation effects, in this case, the simplification of the matrices  $\mathbf{M}$ ,  $\mathbf{N}$ . The block that represents the full dynamic equation of the motion base in Fig. 7 corresponds to Eq. (10). Now, substituting Eq. (16) into Eq. (10) and simplifying it, one gets

$$\ddot{\mathbf{e}} + \mathbf{K}_d \dot{\mathbf{e}} + \mathbf{K}_p \mathbf{e} = \mathbf{w} - \mathbf{u}; \quad (18)$$

where

$$\begin{aligned} \mathbf{w} &= (\mathbf{I} - \mathbf{M}^{-1} \widehat{\mathbf{M}}) \mathbf{v} - \mathbf{M}^{-1} \Delta \mathbf{N} \\ \Delta \mathbf{N} &= \mathbf{N} - \widehat{\mathbf{N}} \end{aligned} \quad (19)$$

The state space representation of the tracking error dynamics described by Eq. (18) is given as

$$\dot{\mathbf{x}} = \mathbf{A} \mathbf{x} + \mathbf{B} (\mathbf{w} - \mathbf{u}), \quad (20)$$

where the state vector  $\mathbf{x}$  consisting of the error and its derivative is written as:

$$\mathbf{x} = [\mathbf{e} \quad \dot{\mathbf{e}}]^T, \quad (21)$$

and where

$$\mathbf{A} = (\mathbf{H} - \mathbf{BK}) \quad , \quad \mathbf{K} = [\mathbf{K}_p \quad \mathbf{K}_d] \quad (22)$$

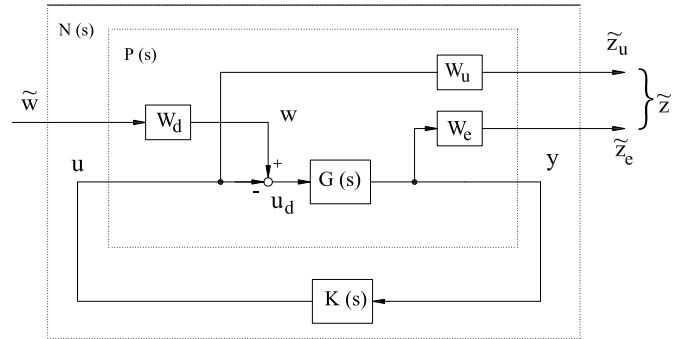
and

$$\mathbf{H} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix} \quad (23)$$

In this context, the term  $\mathbf{u}$  must be designed to stabilize the tracking error dynamics defined by Eq. (20) in the presence of the uncertainty  $\mathbf{w}$ , which represents the imperfect compensation effects. In the next section, the strategy will be designed in order to find this term.

## 1.2 Robust outer loop design by $H_\infty$ control

In order to use  $H_\infty$  control one needs to put the system in Eq. (20) into linear fractional transformation frame as shown in figure 8.


 Figure 8.  $H_\infty$  control problem.

$\mathbf{G}(s)$  is the transfer function of the state space system defined by Eq. (20), and  $\mathbf{W}_e(s)$ ,  $\mathbf{W}_d(s)$  and  $\mathbf{W}_u(s)$  represent the weighting functions diagonal matrices associated with the tracking error, disturbance and control signal, respectively (uncertainty  $\mathbf{w}$  in Eq. (19) is considered as disturbance).

Then the  $H_\infty$  suboptimal control problem is to find a stabilizing controller  $\mathbf{K}(s)$  which, based on the information in  $\mathbf{y}$ , generates a control signal  $\mathbf{u}$  that counteracts the influence of  $\tilde{\mathbf{w}}$  on  $\tilde{\mathbf{z}}$ , thereby minimizing the closed-loop norm from  $\tilde{\mathbf{w}}$  to  $\tilde{\mathbf{z}}$  to less than gamma via the selected weighting function matrices, that is

$$\left\| \begin{bmatrix} \mathbf{W}_u (\mathbf{I} + \mathbf{KG})^{-1} \mathbf{K} \mathbf{G} \mathbf{W}_d \\ \mathbf{W}_e (\mathbf{I} + \mathbf{GK})^{-1} \mathbf{G} \mathbf{W}_d \end{bmatrix} \right\|_\infty < \gamma \quad (24)$$

### 1.2.1 Weights selection

Because the tracking error should be zero at steady state, the weighting function  $W_e$  should include a pure integrator; however, to get a stable weight and to prevent numerical

problems in the algorithm used to synthesize the controller, it is necessary to place a pole very close to the origin in the left half plane, thus  $W_e$  is given as

$$W_e(s) = \frac{s/M_s + \omega_b}{s + \omega_b A_s} \quad (25)$$

where  $A_s$  is chosen small to force tracking at low frequencies,  $M_s$  is the bound in high frequencies and  $\omega_b$  is the bandwidth of the flight simulator motion base.

$W_d$  is chosen to model the characteristic frequency of the disturbance and it is given as

$$W_d = \frac{M_d s + \omega_b}{s + \omega_b / A_d} \quad (26)$$

where  $A_d$  lower bounds the transfer functions in Eq. (24), penalizing the tracking error and the controller energy, respectively, whereas  $M_d$  upper bounds the transfer functions. As the disturbance (modeling error of inexact cancellation of the inverse dynamic control strategy, Eq. (19)) is generated through inverse dynamic computation, the bandwidth frequency is the same as the one of the motion system.

$W_u$  has to have the character of a high pass filter in order to reduce the effect of noise on plants outputs in high frequencies, thus

$$W_u(s) = \frac{s/M_u + \omega_b}{s + \omega_b A_u} \quad (27)$$

where  $A_u$  normalizes the control signal (thus bounding the actuator signal) at low frequencies and  $M_u$  upper bounds the control sensitivity function. Thus, the weighting functions matrices are given as:

$$\mathbf{W}_e = \begin{bmatrix} W_e(s) & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & W_e(s) \end{bmatrix}; \mathbf{W}_d = \begin{bmatrix} W_d(s) & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & W_d(s) \end{bmatrix}$$

$$\mathbf{W}_u = \begin{bmatrix} W_u(s) & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & W_u(s) \end{bmatrix} \quad (28)$$

It is observed from Eq. (24) that the transfer functions,  $(\mathbf{I} + \mathbf{GK})^{-1}\mathbf{G}$  and  $(\mathbf{I} + \mathbf{KG})^{-1}\mathbf{KG}$  are two-sided weighted functions, therefore the terms  $A_s$ ,  $A_d$ ,  $A_u$  and  $M_s$ ,  $M_d$ ,  $M_u$  lower and upper bounds the spectrum of them.

### 1.3 Characteristics of the dynamic equations

Parallel manipulators motion bases have some drawback of relatively small workspace comparing to serial manipulators. In flight simulators motion bases, this is due mainly to the physical restriction in terms of position, velocity and acceleration of the actuators, e.g, for low frequencies motion, the velocity and position constraints limit the maximal attainable acceleration. Moreover, the high pass wash-out filter characteristics keep the motion system not very far away from the neutral position, to prevent the actuators from running out of stroke. Thus, the matrices  $\widehat{\mathbf{M}}(\mathbf{q})$  and  $\widehat{\mathbf{N}}(\mathbf{q})$  of the control law in Eq. (16) can be approximated to constant ones without introducing large modelling errors.

Based on these constant matrices, calculation of the inverse dynamics becomes much simpler, reducing computation time significantly.

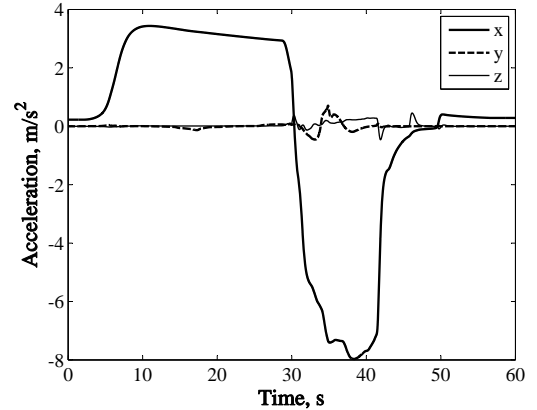
In this context, matrices  $\widehat{\mathbf{M}}(\mathbf{q})$  and  $\widehat{\mathbf{N}}(\mathbf{q})$ , considered in the control law in Eq. (16), are defined at the neutral position as:

$$\begin{aligned} \widehat{\mathbf{M}}(\mathbf{q}_n) &= \mathbf{K}_a^{-1} \mathbf{J}_{l,\omega}^{-T}(\mathbf{q}_n) \mathbf{M}_p(\mathbf{q}_n) \\ \widehat{\mathbf{N}}(\mathbf{q}_n) &= \widehat{\mathbf{G}}(\mathbf{q}_n) = \mathbf{K}_a^{-1} \mathbf{J}_{l,\omega}^{-T}(\mathbf{q}_n) \mathbf{G}_p(\mathbf{q}_n) \end{aligned} \quad (29)$$

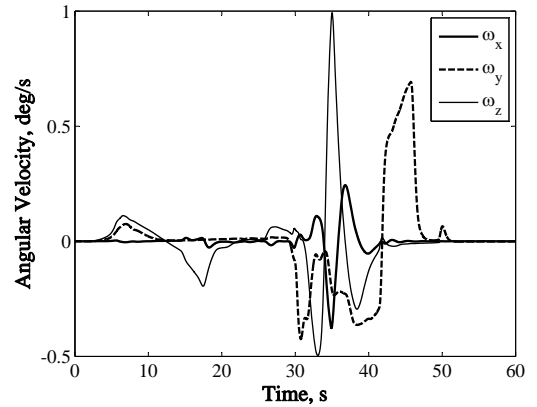
where  $\mathbf{q}_n$  represents a neutral position and was chosen to be at half stroke of all the actuators,  $\mathbf{M}_p(\mathbf{q}_n)$  is the inertia matrix (Eq. (31)) calculated at the neutral position,  $\mathbf{G}_p(\mathbf{q}_n)$  is the gravity vector (Eq. (34)) calculated at the neutral position and  $\mathbf{J}_{l,\omega}^{-T}(\mathbf{q}_n)$  is the jacobian (Eq. (38)) calculated at the neutral position. Coriolis and centrifugal forces, and leg effects, are not considered.

## 2. Controller's Performance Evaluation

At present, only one standard method to characterize the performance of a motion system is known (Koekebakker, 2001). This is described in the AGARD advisory Report (Lean, 1979).



(a) Acceleration Components



(b) Angular Velocity Components

Figure 9. Rejected take-off maneuver.

As an initial control design, two methods defined in the report should be considered: describing function as

a frequency domain evaluation and the step acceleration response in time domain. For each degree of freedom, six describing functions can be calculated.

The primary describing function is the comparison of the response of the motion-base in the driven degree of freedom to the excitation signal (sine acceleration input) and the other five describing functions (crosstalk describing functions) the comparison of pure parasitic motion (motion in other than the degree of freedom excited) to the excitation signal (Grant, 1986). Furthermore, to evaluate the system in its normal operating mode, some standard maneuvers should be evaluated as well.

In order to evaluate the control strategy, the amplitude of the excitation signals (sine acceleration inputs, step acceleration inputs and the desired trajectories from the manouvers) was chosen to keep the motion base approximately 70% of the system limits in position, velocity and acceleration.

Desired aircraft trajectories were generated from a flight-simulation model of a Boeing 747-400 in the UTIAS (University of Toronto Institute for Aerospace Studies) research flight simulator, and then passed through the well-known classical washout filter (Nahon and Reid, 1990) in order to obtain the desired motion base trajectories.

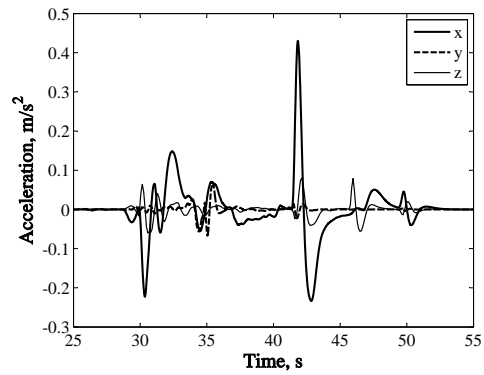
The aircraft rejected take-off maneuver acceleration and angular velocity as shown in Figure 9. From  $t = 10$  seconds, the aircraft accelerates to takeoff ( $x$  forward direction,  $a_x = 4m/s^2$ ). After 20 seconds ( $t = 30$  sec), the pilot decides to abort the take-off by applying brakes and retracting flaps as necessary ( $a_x = -8 m/s^2$ ). At  $t = 50$  seconds the take-off has been aborted. The angular velocity of the aircraft is shown in Figure 9-b.

The filtered aircraft trajectory is shown in Figure 10 and represents the flight simulator motion base desired trajectory. It can be observed from Figure 10 that sustained forward acceleration ( $x$  acceleration component) is represented through a tilt coordination (angle  $\theta$  in Figure 10-d) and tilt angular velocity is limited to prevent the sensation of the angular rotation rate associated with the tilting (approx. 3 deg/s as shown in Figure 10-b).

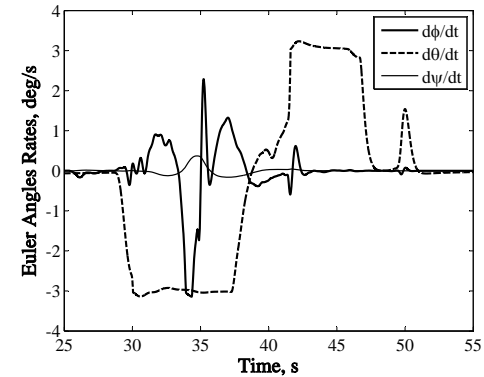
### 3. Numerical Results and Discussions

The performance of the proposed controllers is verified by numerical simulations, and results are presented only for surge ( $x$ ), heave ( $z$ ) and pitch ( $\theta$ ) degrees of freedom. The Runge-Kutta fourth-order numerical integration method is used to solve the ordinary differential equation of the dynamic model. Computer codes are written in MATLAB. Geometric and inertial parameters of the motion base system are shown in Appendix B.

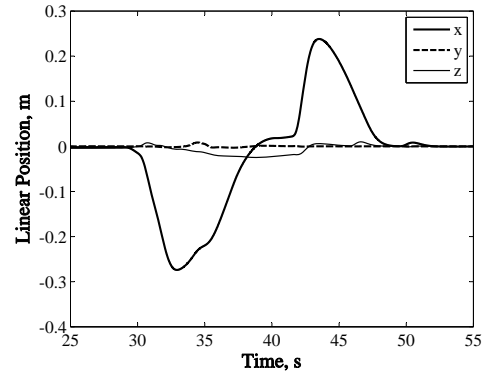
In order to yield a low-order controller it was considered that: the penalized output signals ( $\mathbf{y}$  vector in Fig. 8) only correspond to the position error, and, the weighting functions are constants, i.e.  $\mathbf{y}$  is a 6x1 vector and  $\mathbf{W}_e$  is a 6x6 matrix. Therefore, the weighting functions are given as:



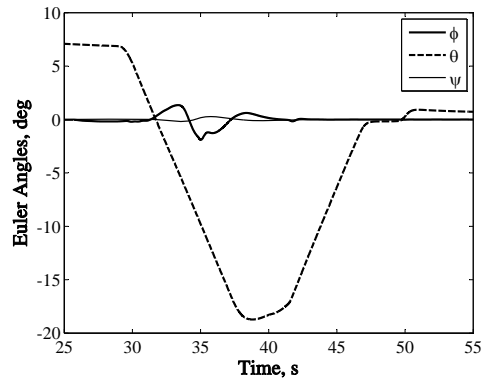
(a) Acceleration Components



(b) Euler Angles Rates



(c) Linear Position Components



(d) Euler Angles

Figure 10. Desired motion simulator

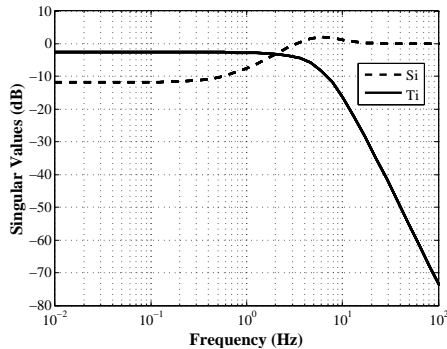
$$\mathbf{W}_e = \begin{bmatrix} 1/A_e & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1/A_e \end{bmatrix}; \mathbf{W}_d = \begin{bmatrix} A_d & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & A_d \end{bmatrix} \quad (30)$$

$$\mathbf{W}_u = \begin{bmatrix} 1/A_u & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1/A_u \end{bmatrix}$$

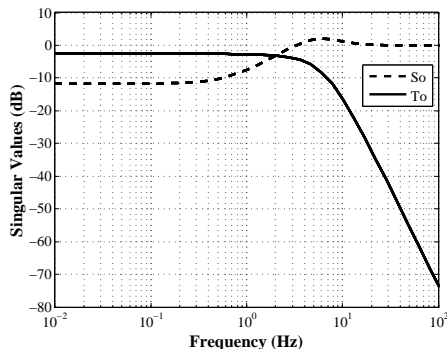
where  $A_e$ ,  $A_d$  and  $A_u$  were chosen as 0.008, 2 and 2.4, respectively.

With relation to the controller gains in Eq. (15), Koekebakker (2001) states the frequency  $\omega_i$  should not exceed human sensory thresholds and that it should ideally be sufficiently smooth and only require limited bandwidth (well below 1 Hz). In this paper, a bandwidth,  $\omega_i$ , of 2 Hz and a damping coefficient,  $\zeta_i$ , of 1, were chosen necessarily to get the desired performance. Running through the  $\gamma$  iteration technique (Skogestad and Postlethwaite, 2005) resulted in a suboptimal solution when  $\gamma = 0.7415$ .

The output ( $\mathbf{S}_o$ ) and input ( $\mathbf{S}_i$ ) sensitivity functions, and the output ( $\mathbf{T}_o$ ) and input ( $\mathbf{T}_i$ ) complementary sensitivity functions, defined from Figure 8, are shown in Figure 11.



(a) Input sensitivity functions



(b) Output sensitivity functions

Figure 11. Sensitivity functions.

Figure 11 proves the low-pass filter behaviour of  $\mathbf{S}_i$  (input disturbance rejection) and  $\mathbf{S}_o$  (output disturbance rejection), and the high-pass filter behaviour of  $\mathbf{T}_o$  (noise attenuation control in measuring output) and  $\mathbf{T}_i$  (noise attenuation control

in measuring input).  $\mathbf{S}_o$  is less important in flight simulator motion bases considering no external forces acting on the motion base.

Acceleration step responses are shown in Figure 12. One can observe a damped response with a small time constant.

The heave, surge and pitch describing functions in Fig. 13 are very similar and present a flat response and bandwidth frequencies of approximately 20 Hz (typically bandwidth frequency in high performance flight simulators).

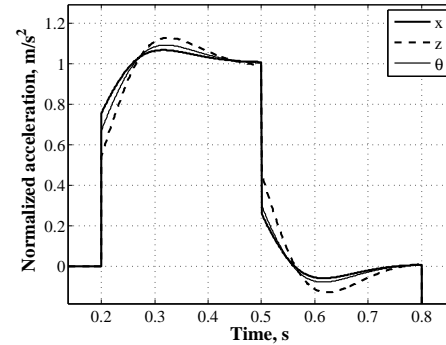


Figure 12. Acceleration step responses.

Crosstalks describing functions with the base driven in the surge direction ( $x$ ) are shown in Figures 14. A rise occurs on parasitic pitch direction ( $\theta$ ) as the frequency increases, due possibly to the nonlinear coupling among the legs at high frequencies. Similar results were presented in the other directions, i.e. with the base driven in the pitch direction, a rise occurs on parasitic surge direction as the frequency increases.

The acceleration response to the rejected take-off maneuver is shown in Figure 15. The driving torques supplied by the servo-motors of the servo-electromechanical actuators are presented in Figure 16. The motor angular velocities are shown in Figure 17. Similarly, the driving forces supplied by the actuators are shown in Figure 18.

One can observe from Figures 15, 16 and 18 short time peak accelerations, torques and forces (in 30 and 42 seconds approx), respectively. These peaks are produced due to the angular rate limiting that is applied in the washout filter (angular rate limiting can be observed in Figure 10-b). Angular rate limit leads to a very large angular jerk (derivate of acceleration), therefore large translation jerk is produced. This is important due to two reasons. One is a large jerk can lead to false cues, and the other is that if the magnitude of the jerk required is not considered, the motor may be undersized and the system won't perform as required. A predictive motion cueing algorithm could avoid these false cues.

In order to compare the robust inverse dynamics controller with the inverse dynamics controller, the control robust term  $\mathbf{u}$  in Eq. (17) is eliminated and the step response in the surge direction ( $x$ ) of both controllers are shown in Figure 19. Noise was added to the controller inputs to test the robustness of them. One can observe that robust controller achieved a better noise reduction than the inverse dynamics controller alone, and higher controller gains are required (without robust term) to get a good response. In fact, and as pointed before, high controller gain can produce a rough correction in the position error and therefore can produce false cues.



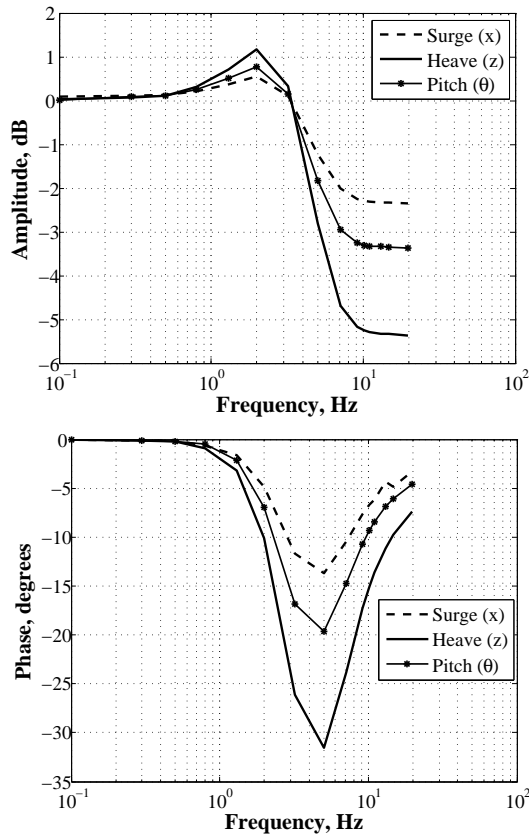


Figure 13. Describing function.

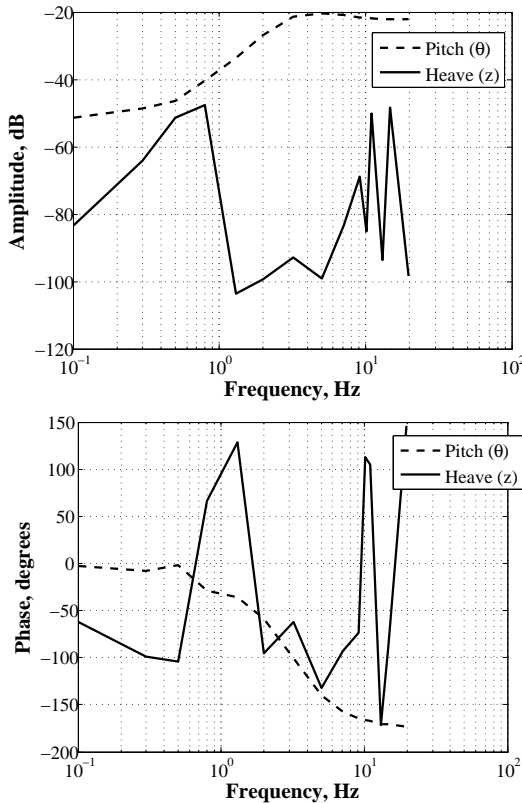
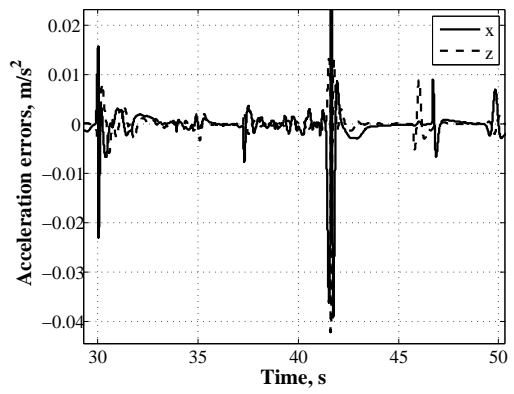
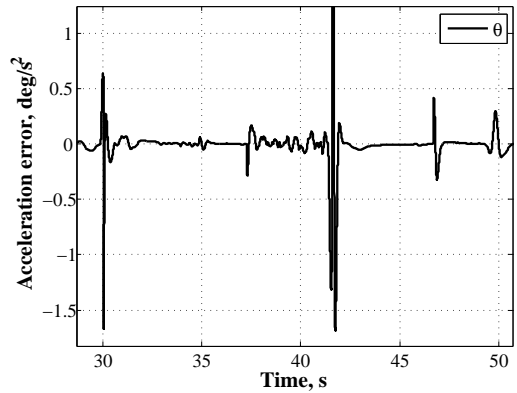


Figure 14. Crosstalks describing function - Motion base driven in the surge direction (x).



(a) Surge (x) and heave (z) directions



(b) Pitch direction ( $\theta$ )

Figure 15. Rejected take-off maneuver - Acceleration errors.

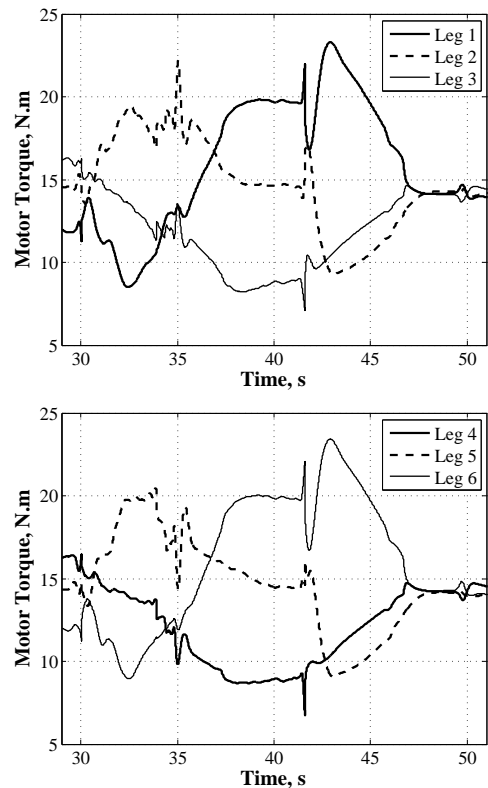


Figure 16. Driving torques supplied by the servo-motors.

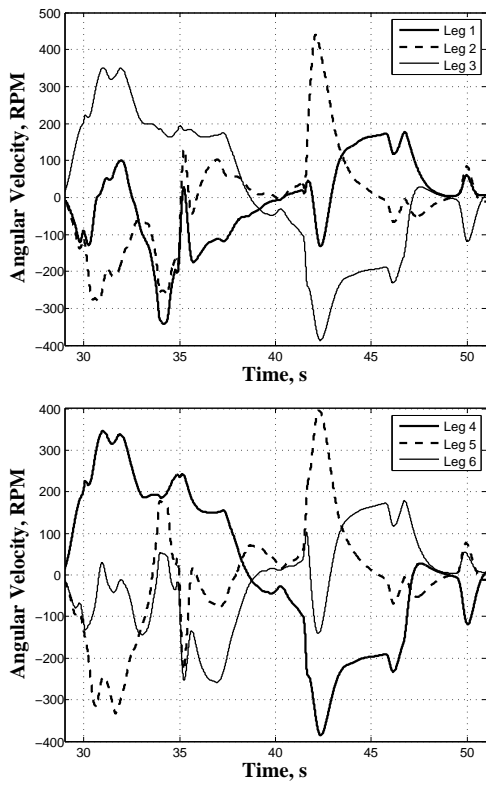


Figure 17. Motor angular velocities.

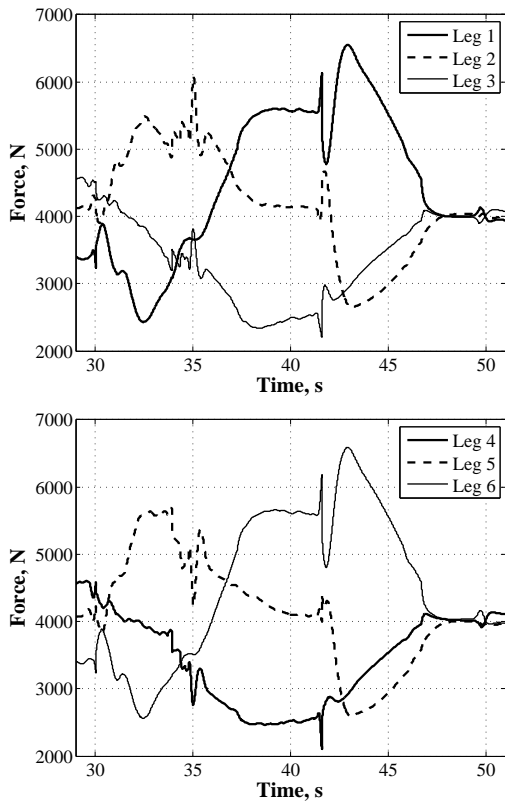
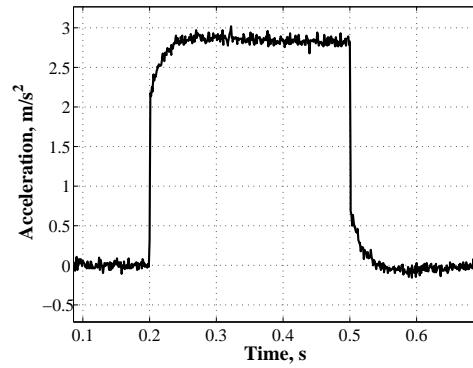
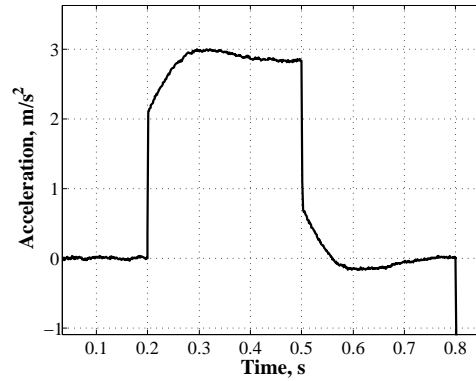


Figure 18. Driving forces supplied by the actuators.



(a) Inverse dynamics control



(b) Robust inverse dynamics control

Figure 19. Step response in the surge (x) direction - Measured sensor noise.

#### 4. Conclusions

In this paper, a control approach for the motion control of a flight simulator motion base was presented. The controller was implemented in the outerloop of the inverse dynamic control scheme in order to counteract imperfect compensation. Imperfect compensation was included intentionally by defining the motion base nominal dynamic matrices as constants. The approach was designed via  $H_\infty$  theory. Controller performance evaluation was carried out through describing functions, step acceleration inputs and some standard manoeuvres. The controller presented robustness to bounded modelling error and measured sensor noise.

#### Acknowledgements

The author gratefully acknowledge the financial support provided by FAPESP (Sao Paulo State Foundation for Research Support - Brazil) through contract number 2005/25486-6.

#### References

Becerra-Vargas, M., E. Belo, and P. Grant (2009). Robust control of a flight simulator motion base. In *AIAA Modeling and Simulation Technologies Conference*, Chicago, IL, pp. 1–13. AIAA.

Dasgupta, B. and T. Mruthyunjaya (1998). Closed form dynamic equations of the general Stewart platform through the Newton-Euler approach. *Mechanism and Machine*

*Theory* 33(7), 993–1012.

- Graf, R., R. Vierling, and R. Dillman (1998). A flexible controller for a Stewart platform. In *Proceedings of the 1998 Second International Conference on Knowledge-Based Intelligent Electronic Systems*, Adelaide-SA-Australia, pp. 52–59. IEEE.
- Grant, P. (1986). Motion characteristics of the UTIAS flight research simulator motion-base. Technical report, University of Toronto - UTIAS, Toronto-Canada. Technical Note No. 261.
- Idan, M. and D. Sahar (1996). Robust controller for a dynamic six degree of freedom flight simulator. In *AIAA Flight Simulation Technologies Conference*, San Diego, CA, pp. 53–64. AIAA.
- Koekebakker, S. (2001). *Model based control of a flight simulator motion system*. Ph. D. thesis, Delft University of Technology, Netherlands.
- Lean, D. (1979). Dynamic characteristics of flight simulation motion systems. Technical report, France. AGARD Advisory Report No. 144.
- Nahon, M. and L. D. Reid (1990). Simulator motion drive algorithms - a designer's perspective. *Journal of Guidance, Control, and Dynamics* 13(2), 702–709.
- Nguyen, C., S. Antrazi, Z. Zhou, and C. Campbell (1993). Adaptive control of a Stewart platform-based manipulator. *Journal of Robotic systems* 10(5), 657–687.
- Salcudean, S., P. Drexel, D. Ben-Dov, A. Taylor, and P. Lawrence (1994). A six degree-of-freedom, hydraulic, one person motion simulator. In *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, San Diego-CA, pp. 2437–2443. IEEE.
- Sciavicco, L. and B. Siciliano (2005). *Modeling and Control of Robot Manipulators* (2 ed.). Londres: Springer-Verlag.
- Skogestad, S. and I. Postlethwaite (2005). *Multivariable Feedback Control* (2 ed.). England: John Wiley & Sons Ltd.
- Spong, M. and M. Vidyasagar (2006). *Robot Modeling and Control* (1 ed.). New York: John Wiley & Sons.

## Appendix A

The matrices of the motion base dynamic model are given by:

$$\mathbf{M}_p = \begin{bmatrix} M\mathbf{I} & -M\tilde{\mathbf{R}}\mathfrak{R}_\omega \\ M\tilde{\mathbf{R}} & (\mathbf{I}_p - M\tilde{\mathbf{R}}\tilde{\mathbf{R}})\mathfrak{R}_\omega \end{bmatrix};$$

$$\mathbf{M}_a = \begin{bmatrix} \sum_{i=1}^6 \mathbf{Q}_i & -\left(\sum_{i=1}^6 \mathbf{Q}_i(\tilde{\mathbf{q}}_p)_i\right)\mathfrak{R}_\omega \\ \sum_{i=1}^6 (\tilde{\mathbf{q}}_p)_i \mathbf{Q}_i & -\left(\sum_{i=1}^6 (\tilde{\mathbf{q}}_p)_i \mathbf{Q}_i(\tilde{\mathbf{q}}_p)_i\right)\mathfrak{R}_\omega \end{bmatrix} \quad (31)$$

$$\mathbf{C}_p = \begin{bmatrix} M\boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{R}) \\ \boldsymbol{\omega} \times \mathbf{I}_p \boldsymbol{\omega} + M\mathbf{R} \times (\boldsymbol{\omega} \times \mathbf{R}) \boldsymbol{\omega} \end{bmatrix} - \begin{bmatrix} M\tilde{\mathbf{R}} \\ M\tilde{\mathbf{R}}\tilde{\mathbf{R}} - \mathbf{I}_p \end{bmatrix} \mathfrak{R}_\omega \dot{\boldsymbol{\omega}} \quad (32)$$

$$\mathbf{C}_a = \begin{bmatrix} \sum_{i=1}^6 (\mathbf{V}_c)_i \\ \sum_{i=1}^6 ((\mathbf{q}_p)_i \times (\mathbf{V}_c)_i) \end{bmatrix} - \begin{bmatrix} \sum_{i=1}^6 \mathbf{Q}_i(\tilde{\mathbf{q}}_p)_i \\ \sum_{i=1}^6 (\tilde{\mathbf{q}}_p)_i \mathbf{Q}_i(\tilde{\mathbf{q}}_p)_i \end{bmatrix} \mathfrak{R}_\omega \dot{\boldsymbol{\omega}} \quad (33)$$

$$\mathbf{G}_p = - \begin{bmatrix} M\mathbf{g} \\ M\mathbf{R} \times \mathbf{g} \end{bmatrix}; \quad \mathbf{G}_a = \begin{bmatrix} \sum_{i=1}^6 (\mathbf{V}_g)_i \\ \sum_{i=1}^6 ((\mathbf{q}_p)_i \times (\mathbf{V}_g)_i) \end{bmatrix} \quad (34)$$

$$\mathbf{B}_t = \begin{bmatrix} \sum_{i=1}^6 (\mathbf{V}_f)_i \\ \sum_{i=1}^6 ((\mathbf{q}_p)_i \times (\mathbf{V}_g)_i) - \mathbf{f}_i \end{bmatrix}. \quad (35)$$

where  $\mathbf{Q}_i$  depends on the  $i$ th leg inertial properties and

$$(\mathbf{V})_i = (\mathbf{V}_c)_i + (\mathbf{V}_g)_i + (\mathbf{V}_f)_i, \quad (36)$$

where  $(\mathbf{V}_c)_i$  and  $(\mathbf{V}_g)_i$  depend on the  $i$ th leg dynamic properties, and  $(\mathbf{V}_f)_i$  is the viscous friction force vector at the  $i$ th leg joints, and where

$$\mathfrak{R}_\omega = \begin{bmatrix} C\psi C\theta & -S\psi & 0 \\ C\theta S\psi & C\psi & 0 \\ -S\psi & 0 & 1 \end{bmatrix} \quad (37)$$

The motion base Jacobian is given as

$$\mathbf{J}_{1,\omega} = \begin{bmatrix} \mathbf{s}_1 & \mathbf{s}_2 & \mathbf{s}_3 & \mathbf{s}_4 & \mathbf{s}_5 & \mathbf{s}_6 \\ \mathbf{q}_1 \times \mathbf{s}_1 & \mathbf{q}_2 \times \mathbf{s}_2 & \mathbf{q}_3 \times \mathbf{s}_3 & \mathbf{q}_4 \times \mathbf{s}_4 & \mathbf{q}_5 \times \mathbf{s}_5 & \mathbf{q}_6 \times \mathbf{s}_6 \end{bmatrix}^T, \quad (38)$$

and the Jacobian that maps the cartesian coordinates into joint coordinates is given as

$$\mathbf{J}_{l,q} = \mathbf{J}_{l,\omega} \mathbf{J}_{\omega,q} \quad (39)$$

where

$$\mathbf{J}_{\omega,q} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathfrak{R}_\omega \end{bmatrix}$$

The matrices of the equation of motion of the electromechanical actuator are given as

$$\mathbf{K}_a = \begin{bmatrix} K_a & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & K_a \end{bmatrix}; \quad \mathbf{D}_a = \begin{bmatrix} D_a & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & D_a \end{bmatrix};$$

$$\mathbf{B}_a = \begin{bmatrix} B_a & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & B_a \end{bmatrix} \quad (40)$$

where

$$K_a = \frac{2\pi\eta}{p}; \quad M_a = \frac{J_t 4\pi^2 \eta}{p^2}; \quad B_a = \frac{B_t 4\pi^2 \eta}{p^2}$$

and where  $p$  and  $\eta$  are the lead and efficiency of the ballscrew, and  $J_t$  and  $B_t$  are the moment of inertia and viscous damping of the rotor and ballscrew, respectively.

### Appendix B

A symmetrical distribution of joints on the payload are arranged as shown in Figure 20, where the angles  $\theta_B$  and  $\theta_P$  and the upper and base platform radius values,  $r_P$  and  $r_B$ , respectively, are given in Table 1. The neutral position is chosen at half stroke of all the actuators, i.e.

$$\mathbf{q}_n = [ 0 \ 0 \ -2.154\text{m} \ 0 \ 0 \ 0 ]^T \quad (41)$$

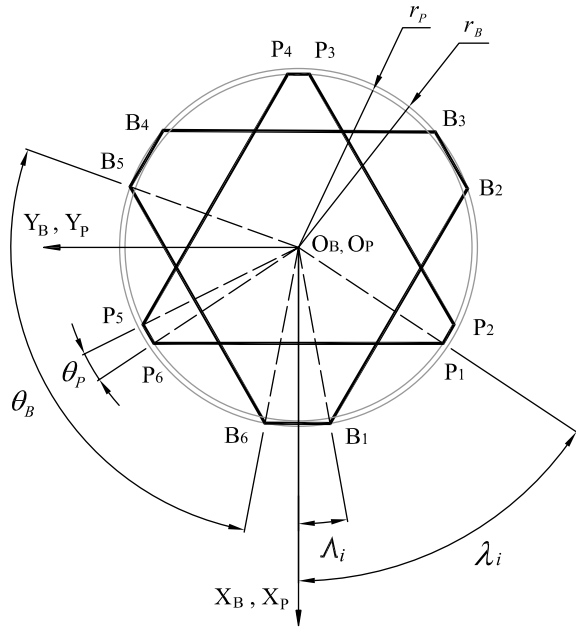


Figure 20. Platform and base leg points distribution.

Table 1. Geometric and inertial parameters.

Parameter	Value	Parameter	Value
$\theta_P$	$100^\circ$	$\theta_B$	$20^\circ$
$r_P$	1.60 m	$r_B$	1.65 m
$M$	2500 Kg	$M_{act}$	100 Kg