# Vehicle Sprung Mass Parameter Estimation Using an Adaptive Polynomial-Chaos Method

**Samuel Kline Shimp III**

Thesis submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Master of Science

in

Mechanical Engineering

**Dr. Steve C. Southward, Chair**

**Dr. Mehdi Ahmadian**

**Dr. Corina Sandu**

18 April 2008

Blacksburg, Virginia

Keywords: Polynomial-Chaos, adaptive parameter estimation, adaptive control, uncertainty propagation, stochastic modeling.

# Vehicle Sprung Mass Parameter Estimation Using an Adaptive Polynomial-Chaos Method

**Samuel Kline Shimp III**

## Abstract

The polynomial-chaos expansion (PCE) approach to modeling provides an estimate of the probabilistic response of a dynamic system with uncertainty in the system parameters. A novel adaptive parameter estimation method exploiting the polynomial-chaos representation of a general quarter-car model is presented. Because the uncertainty was assumed to be concentrated in the sprung mass parameter, a novel pseudo mass matrix was developed for generating the state-space PCE model. In order to implement the PCE model in a real-time adaptation routine, a novel technique for representing PCE output equations was also developed. A simple parameter estimation law based on the output error between measured accelerations and PCE acceleration estimates was developed and evaluated through simulation and experiment. Simulation results of the novel adaptation algorithm demonstrate the estimation convergence properties as well as its limitations. The simulation results are further verified by a real-time experimental implementation on a quarter-car test rig. This work presents the first truly real-time implementation of a PCE model. The experimental real-time implementation of the novel adaptive PCE estimation method shows promising results by its ability to converge and maintain a stable estimate of the unknown parameter.

## Acknowledgements

I am deeply grateful to my wife Kela for supporting me so faithfully and patiently through my academic endeavors. Her personal sacrifices for our family have been a motivating example for me to complete this work. The patience she has shown me has been a source of calm and reason when I was too overwhelmed to think. Her philosophies on life and living have kept me sane by teaching me how to find balance in an imbalanced world. Her thoughtful approach to important decisions has taught me valuable discipline.

I am grateful for my son Kline's endless energy. He is the child embodiment of polynomial *chaos*! He gives me a joyous reason to go home each day. He is a constant reminder of why I must strive to make a difference in the world.

I am eternally indebted to my Lord and Savior. All success comes according to His Will and for His eternal purposes. I am grateful to have been able to grow and learn here at Virginia Tech, not only academically, but also personally through the numerous opportunities to serve others.

I am grateful to Dr. Steve Southward for his patient guidance and willingness to share his experience and expertise. I thank him for his patience with my decisions to try new career paths that were significantly different from my own (and his!) initial expectations.

I also thank all the members of my committee. They have each been mentors, colleagues and friends throughout all of our experiences working, communicating and traveling together.

# Contents

## Nomenclature

| | |
|---|---|
| $m_s$ | sprung mass (kg) |
| $m_{us}$ | unsprung mass (kg) |
| $b_0$ | linear bearing damping coefficient (N-s/m) |
| $k_s$ | suspension stiffness (N/m) |
| $k_{us}$ | tire stiffness (N/m) |
| $b_s$ | suspension damping coefficient (N-s/m) |
| $b_u$ | unsprung mass (tire) damping coefficient (N-s/m) |
| $\mathbf{u}(t)$ | state space model excitation vector |
| $\mathbf{w}(t)$ | quarter-car random displacement input (m) |
| $\dot{\mathbf{w}}(t)$ | quarter-car random velocity input (m/s) |
| $x_s(t)$ | sprung mass position as a function of time (m) |
| $x_{us}(t)$ | unsprung mass position as a function of time (m) |
| $X(\xi)$ | random process, a function of the random event $\xi$ |
| $\xi$ | uncertain parameter defined by a probability distribution along the interval $(-1,1)$ |
| $\phi_j(\xi)$ | $j^{th}$ orthogonal polynomial basis function of $\xi$ |
| $c_j$ | coefficient of the $j^{th}$ orthogonal polynomial basis function |

| | |
|---|---|
| $\dot{x}(t), x(t)$ | linear, deterministic state vectors |
| $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ | deterministic state-space equation matrices |
| $\dot{\hat{x}}(t,\xi), \hat{x}(t,\xi)$ | orthogonal series expansion of state vectors |
| $\mathbf{A}(\xi), \mathbf{B}(\xi)$ | series expansion of state and input matrices |
| $\mathbf{u}(t,\xi)$ | series expansion of state-space model excitation |
| $\dot{x}_s(t), \dot{x}_{us}(t)$ | sprung and unsprung mass velocity states (m/s) |
| PCE | polynomial chaos expansion |
| pdf | probability density function |
| $m_0$ | assumed mean value of the uncertain sprung mass (kg) |
| $m_1$ | assumed max variation of uncertain sprung mass (kg) |
| $m_s(\xi)$ | parameterized sprung mass (kg) |
| $m_j$ | coefficient for series expansion of sprung mass (kg) |
| $\dot{\hat{x}}_p(t,\xi), \hat{x}_p(t,\xi)$ | $p^{th}$ state of the series expansion of state vector |
| $\dot{\hat{x}}_{p,i}(t,\xi), \hat{x}_{p,i}(t,\xi)$ | $i^{th}$ PCE state trajectory for the $p^{th}$ state of the series expansion of state vector |
| $\mathbf{M}(\xi)$ | pseudo mass matrix parameterized by $\xi$ |
| $\hat{\mathbf{A}}, \hat{\mathbf{B}}$ | state matrices of deterministic model parameters (uncertain parameters factored out) |
| $\delta_{ni}$ | Galerkin projection of $\phi_i(\xi)$ onto $\phi_n(\xi)$ |

| | |
|---|---|
| $\delta_{nij}$ | Galerkin projection of $\phi_i(\xi)\phi_j(\xi)$ onto $\phi_n(\xi)$ |
| $[\delta_{ni}], [\delta_{nij}], [\beta_{ni}]$ | matrices of Galerkin projections |
| $\dot{\hat{\mathbf{X}}}(t), \hat{\mathbf{X}}(t)$ | vector of PCE state trajectories |
| $\mathbf{M}_{PCE}$ | PCE mass matrix |
| $\mathbf{A}_{PCE}, \mathbf{B}_{PCE}$ | PCE state matrices |
| $\delta_{input}$ | Galerkin projected input vector |
| $\mathbf{\Phi}(\xi)$ | vector of orthogonal polynomial basis functions |
| $\hat{\mathbf{y}}(t,\xi)$ | stochastic output equations |
| $\mathbf{M}_{PCE,output}(\xi)$ | PCE output equation mass matrix |
| $\mathbf{C}_{PCE}(\xi), \mathbf{D}_{PCE}(\xi)$ | PCE output equation matrices |
| $J(t,\xi)$ | adaptive algorithm cost function |
| $e_s(t,\xi)$ | sprung mass accelerometer and PCE output error |
| $e_{us}(t,\xi)$ | unsprung mass accelerometer and PCE output error |
| $\mu$ | adaptive algorithm gradient search step size |
| $\dot{\xi}(t)$ | gradient of the estimated parameter |
| $\frac{\partial J(t,\xi)}{\partial \xi}$ | derivative of cost function with respect to $\xi$ |
| $\frac{\partial e_s(t,\xi)}{\partial \xi}, \frac{\partial e_{us}(t,\xi)}{\partial \xi}$ | derivative of errors with respect to $\xi$ |

| | |
|---|---|
| $\mathbf{C}_1(\xi), \mathbf{C}_2(\xi)$ | row vector representation of the output matrix |
| $\mathbf{D}_1(\xi), \mathbf{D}_2(\xi)$ | row vector representation of the direct feed through matrix |
| $\frac{d\mathbf{C}_1(\xi)}{d\xi}, \frac{d\mathbf{C}_2(\xi)}{d\xi}$ | derivatives of row vector representation of output matrix |
| $\frac{d\mathbf{D}_1(\xi)}{d\xi}, \frac{d\mathbf{D}_2(\xi)}{d\xi}$ | derivatives of row vector representation of direct feed through matrix |
| $L_n(\xi)$ | nth order Legendre polynomial function of $\xi$ |
| $\mathbf{q}(\xi)$ | vector of $\xi^v$ for $v = 0,1,\ldots,S$ |
| $\mathbf{P}_n$ | column vector of $n^{th}$ order polynomial function coefficients |
| $\mathbf{P}$ | polynomial function coefficient matrix |
| $\frac{d\Phi(\xi)}{d\xi}$ | derivative with respect to $\xi$ of the vector of polynomial functions |
| $\xi_k$ | $k^{th}$ discrete time estimate of the uncertain parameter |
| $\xi_{k+1}$ | next discrete time estimate of uncertain parameter |
| $\nabla_k$ | gradient for $\xi_{k+1}$ at the $k^{th}$ time step |
| $e_{s,k}(\xi_k), e_{us,k}(\xi_k)$ | sample of sprung and unsprung mass error at $k^{th}$ time step |

| | |
|---|---|
| $\hat{\mathbf{X}}_k$ | PCE state trajectory at kth time step |
| $\hat{m}_0$ | initial estimate of sprung mass (kg) |
| $m_{true}$ | true value of sprung mass (kg) |
| $\mu_1, \mu_2$ | adaptive algorithm gradient step sizes for sprung and unsprung mass acceleration channels |
| $\zeta$ | damping coefficient of general equation of motion for a dynamic system (unitless) |
| $\omega_n$ | natural frequency of general equation of motion for a dynamic system (rad/s) |

# List of Tables



# List of Figures

# 1 Introduction

It is a truth universally acknowledged that all dynamic systems are subject to uncertainty. Uncertainty is present in every system and originates from various sources. Examples include applied forces characterized in stochastic terms; initial conditions represented stochastically, and very commonly, system parameters that are not accurately known. These are examples of the sources of uncertainty that may influence dynamic system response in non-linear, counter-intuitive ways.

## *1.1 Motivation*

The general need for more efficient tools to predict the effects of uncertainty in dynamic control systems is readily seen in the realms of land and aerospace vehicle design and control. Improvements to vehicle modeling techniques and vehicle controller synthesis benefit design and development in aerospace, automotive and racing industries. A growing interest in Polynomial Chaos Expansion (PCE) methods has led to development of new approaches and results in vehicle dynamics research [1-4]. The long term goal of this research was to look beyond predicting the effects of uncertainty in a model and to exploit stochastic modeling techniques for control purposes. Specifically, the application of PCE methods to controller design and synthesis is a novel technique that has seen very little published results.

Solving the problem of estimating an uncertain parameter using PCEs in a dynamic system serves as a preliminary step in the development of stochastic modeling based control algorithms. The problem of estimating one unknown parameter in a 2-degree of freedom system using PCE stochastic modeling techniques is treated in this work.

## 1.2 Research Objectives

This work has two main purposes. First is to unite the stochastic modeling technique known as Polynomial-chaos expansion (PCE) modeling to the real-time adaptive control world. In the stochastic dynamic system modeling community the method has proven to be an efficient tool. To bring its use to the adaptive control world the PCE methodology has been developed and implemented for a linear, discrete-time, 2 degree-of-freedom dynamic system with only one uncertain model parameter. The PCE method of modeling is well suited for non-linear, multiple uncertain parameter dynamic systems, [5-10] but there is need for clear presentation of the method if it is to be exploited by disciplines beyond theoretical modeling.

Second, as a first step in extending the PCE methodology to adaptive control, an adaptive parameter estimation algorithm exploiting the PCE algorithm will be developed and implemented in a real-time environment. This work is a proof of concept study to show that polynomial chaos expansion methods can be used for adaptive parameter estimation. Comparison of the proposed real-time adaptive PCE estimation method to generally accepted, real-time parameter estimation methods will occur in future work. Once the PCE method has been sufficiently proven to work for adaptive estimation, future work can be devoted to real-time controllers based on the method.

## 1.3 Two Degree of Freedom Quarter-Car Model

To present novel parameter estimation methods of a 2-DoF system to the motor sports and aerospace industry, the quarter-car model and corresponding quarter-car test rig provide fantastic testbeds for simulation and testing.

The quarter-car model, in addition to the simplicity it adds to the dynamic response problem, also eliminates non-linearity of some full vehicle models [11]. Quarter-car models are used in the motor sports industry for proof of concept component and control strategy testing [12]. The linear quarter-car model very accurately represents the system that, in real life, is a non-linear system [13, 14].

## *1.4 Modeling Stochastic Dynamic Systems*

There are many tools to be used for propagating uncertainty through a model of a dynamic system in order to predict a response in probabilistic terms. Polynomial-chaos is an approach gaining popularity. In this approach the governing equations of the model are expanded in terms of orthogonal polynomials weighted by probability distribution functions [8-10, 15-17]. The polynomial-chaos expansion (PCE) method is also known as stochastic finite elements [6].

The PCE method has been most frequently used to predict stochastic response of complex, non-linear systems in an offline or batch manner [6, 9, 10]. Under these conditions the method is effective and considerably less computationally demanding than other common methods like Monte Carlo simulations [9, 10]. One strength of the method lies in it ability to model non-linear systems but some attempts have also been made to exploit the method for simpler, linear systems [18, 19]. The PCE approach is especially effective at modeling systems with relatively few uncertainties while allowing the magnitude of the uncertainty to be great. Thorough comparisons of the PCE method against other uncertainty propagation tools can be readily found in the literature [6, 9, 10, 19].

In addition to modeling uncertainty, the PCE approach has also been successfully used for measurement estimation as in [18]. A linear observer was built using the PCE approach. The observer exploited the PCE method to represent the estimated states in probabilistic terms. While the method is intended for real-time probabilistic representation of measurements, only simulation studies were completed and a true real-time algorithm was never published.

## *1.5 Review of Literature*

According to [20], identification is the process of developing a mathematical model of a dynamic system based on observations and prior knowledge. There is an expansive body of literature available on the myriad of ways of estimating system parameters. The typical approach is to define a performance surface or cost function. The performance surface describes how well the estimated parameters match those of the actual system and is some function of the error between the predicted response and the measured response [21, 22]. Each parameter estimation algorithm differs in how the error or cost function is minimized.

Two basic offline methods of error minimization found in the literature are the “Iterative Generalized Least-Squares” [20, 23] and the “Maximum-Likelihood Algorithm” [20, 24]. These methods require complete knowledge of the time response in order to estimate the parameters that minimize the cost function in an iterative fashion.

There are Matlab functions for offline identification of system parameters. One approach of interest for vehicle dynamics research is a method that uses the FMINCON function of the Matlab optimization toolbox. The optimization tool minimizes a cost function

defined by the sum of the errors. This approach is most recently described for identification of a quarter-car rig in [11, 13].

With regards to online (real-time) adaptive parameter estimation, Astrom and Wittenmark discuss “Recursive least-squares estimation” (RLS) and “Extended least squares” (ELS) [21]. These two methods apply Gauss’s principle of least squares. As recursive algorithms they can be applied as a real-time method. Gradient search rules are also discussed by Astrom and Wittenmark and the most basic presented was the MIT rule. This adaptive gradient search approach uses a cost function defined as one-half the error squared. Further development of their gradient search algorithms lead to discussion of more complicated gradient search algorithms based on stability criterion.

Another gradient search method developed by Widrow presents a method of estimating a gradient to advance down the cost function. His method is known as the “least-mean-square (LMS) algorithm” [22]. In this simple method the gradient is the product of the error and the input. It does not require computation of a square, or an average, or differentiation, which makes it advantageous in real-time applications.

## *1.6 Parameter Estimation Using Polynomial Chaos*

PCE methods have also been used to estimate system parameters. Ghanem, et al. united polynomial-chaos and system identification techniques for stochastic dynamic systems. In reference [5], the restoring force for a stochastic dynamic system was identified using offline regression techniques. Subsequently the probabilistic response of the complete system for assumed uncertain spring stiffness was computed using polynomial-chaos expansions. In this case polynomial-chaos methods were not used for identification, but

the stochastic identification technique used facilitated the use of the polynomial-chaos method.

The polynomial-chaos method was utilized by Blanchard, et. al for offline parameter estimation [2]. In this approach, uncertainty in the system is modeled using the PCE method. Parameters are estimated by using the PCE equations for the system in a Bayesian cost function. Minimization of the cost function results in the maximum likelihood estimate for the uncertain parameters. The polynomial chaos based Bayesian approach requires recorded time series data and estimates the parameters offline.

Further parameter estimation was investigated by Blanchard, Sandu and Sandu by applying PCE equations to the state and parameter estimates in the equations for an extended Kalman filter (EKF) [1]. The PCE based EKF method was tested in simulation and compared directly to the Bayesian approach mentioned above.

Pursuing an adaptive approach, an algorithm presented by Southward [19] estimates the system parameter of a first order, 1-DoF system by updating a PCE model of the system until the error is driven to zero. The adaptive parameter estimation routine was tested in a series of simulation studies to illustrate the viability of the method for future real-time implementation.

The available literature provides few examples of truly real-time PCE models in an experimental setting. All PCE based modeling and parameter estimation has been limited to offline methods or simulated results. Additionally, few experimental results exist to validate the parameter estimation simulations. Further work is needed to exploit the strengths of the PCE method in real-time parameter estimation. Until now, no one has

been able to implement a truly real-time parameter estimation algorithm for parameter estimation of a dynamic system in an experimental setup.

## *1.7 Brief Overview of This Work*

This work expands previous work presented in [19] by developing an adaptive parameter estimation algorithm for higher order systems. The deterministic 2 ½-DoF quarter-car model will be transformed into a probabilistic model by use of polynomial-chaos expansions. Theoretical development of the proposed PCE based adaptive algorithm will then be presented. Gradient search criterion and cost functions will be defined and explained. Simulation results of the theoretical method will be presented and discussed. Finally, experimental validation of the proposed algorithm will be shown. Description of the experimental equipment will precede presentation of the data. Comparison of the simulation studies against the data will give insight to the results and a lead-in to future work.

# 2 Background for Theoretical Development

For complete understanding of the theoretical development and analysis of the PCE quarter-car model and the parameter estimation algorithm for the uncertain model parameter, additional background information must be presented. This chapter will present and explain the general 2 ½-DoF quarter-car. For further information on the quarter-car model, see the literature [11-14, 25]. The chapter will also discuss the basic polynomial-chaos theory as found in the literature.

## *2.1 General 2 ½-DoF Quarter-Car Model*

The general 2 ½ -DoF quarter-car model used in this study is represented in the diagram shown in Figure 2.1. The 2 ½ DoF model was chosen over the 2-DoF system in order to excite the system with a velocity input. The quarter-car model used for this study assumes linear tire damping which requires a velocity input. The sprung mass $m_s$ is a lumped parameter representing one-quarter of the vehicle mass supported by the vehicle suspension system. The linear, viscous damping element between the sprung mass and the ground, $b_0$, represents the damping experienced by the rollers that guide the sprung mass along its vertical track on the experimental test rig used in this study. The unsprung mass, $m_{us}$, is also a lumped parameter that represents the wheel-tire assembly. The suspension stiffness is modeled by the linear stiffness element $k_s$. The suspension shock absorber is modeled by the linear viscous damping coefficient $b_s$. The tire stiffness is modeled linearly by $k_{us}$ attached in the diagram beneath the sprung mass. The effects of tire damping are represented in the unsprung mass by the linear viscous damping

coefficient $b_{us}$. All spring and damper elements are assumed to be linear and quasi-static (very slowly changing in time). Excitation of a quarter-vehicle model comes from road inputs to the tire and is represented by the random displacement $w(t)$. The vertical positions of $m_s$ and $m_{us}$ are expressed by $x_s(t)$ and $x_{us}(t)$, respectively.

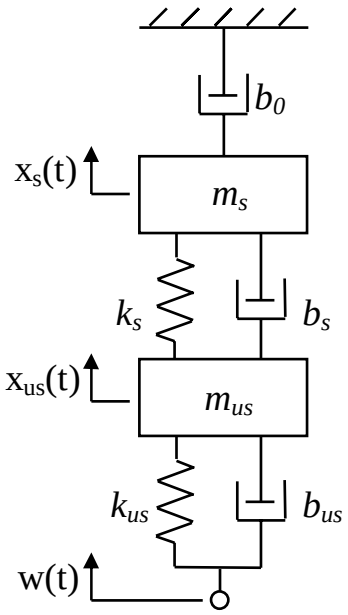


***Figure 2.1 Diagram of a general 2 ½-DoF quarter-car model.***

The equations of motion for a quarter-car represented as differential equations are:

$$\begin{aligned} m_s\ddot{x}_s(t)+(b_s-b_0)\dot{x}_s(t)+k_s x_s(t)-b_s\dot{x}_{us}(t)-k_s x_{us}(t)=0 \\ m_{us}\ddot{x}_{us}(t)+(b_s-b_{us})\dot{x}_{us}(t)+(k_s+k_{us})x_{us}(t)-b_s\dot{x}_s(t)-k_s x_s(t)=b_{us}\dot{w}(t)+k_{us}w(t) \end{aligned} \quad (2.1)$$

Accurate experimental testing of quarter-vehicle models requires a quarter-car test rig. Examples of such rigs can be found in [12]. The experimental quarter-car rigs are problems in and of themselves in that they require characterization or measurement of the system components. Often those components are impossible to measure accurately offline. A quarter-car characterization routine was developed an presented by [11]. In

that work, the dynamic system was characterized and proven to be accurately modeled by the linear model.

## *2.2 Modeling Stochastic Systems with Polynomial Chaos*

The Polynomial Chaos method is derived from Norbert Wiener's Homogenous Chaos presented in 1938 [16]. The method approximates a stochastic process by expanding the process equations with orthogonal polynomial basis functions [17].

A random process can be represented by $X(\xi)$, a function of the random event $\xi$. That process may then be expanded in terms of orthogonal basis functions [6, 9, 10, 19] called polynomial chaoses such that

$$X(\xi) = \sum_{j=0}^{S} c_j \phi_j(\xi) \tag{2.2}$$

This work will use post-character subscripts for indices and reserve post-character superscripts for exponential powers. The summation in (2.2) is a summation truncated at the polynomial of order S, as is represented by the index of the summation.

Each orthogonal polynomial $\phi_j(\xi)$ has its own associated weighting function that is also orthogonal to the polynomial. These weighting functions are easily recognizable as probability density functions. For example, the Hermite polynomials are orthogonally weighted to the Gaussian distribution, the Jacobi polynomials to the beta distribution, the Legendre polynomials to the uniform distribution, etc. [9, 17] The implication is that if a random process is characterized by a random variable with a known probability density

function, the random process can be expanded in terms of the polynomial chaoses and the associated weighting function.

The PCE method is readily extended to a state-space model with uncertain parameters. A linear-time-invariant, deterministic model represented in general state-space terms is

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \tag{2.3}$$

Using (2.2) the state-space equation of (2.3) would be represented by the following equation:

$$\dot{\hat{\mathbf{x}}}(t,\xi) = \mathbf{A}(\xi)\hat{\mathbf{x}}(t,\xi) + \mathbf{B}(\xi)\mathbf{u}(t) \tag{2.4}$$

Using polynomial chaos expansions on a stochastic system in actuality is discretely modeling the system along the stochastic dimension by parameterizing the uncertainty distribution [3, 4, 9, 10, 15, 19]. The 1-DoF dynamic system with uncertainty in the system parameter studied in [19] derives a simplified form of the equation in (2.4).

In order to solve the set of probabilistic state equations created by (2.4) the Galerkin method for computing the projection onto the $n^{th}$ state trajectory can be followed as shown in [6, 9, 10, 19]. The Galerkin projection of the general equation (2.4) results in a set of general PCE state-space equations

$$\dot{\hat{\mathbf{x}}}(t) = \mathbf{A}_{PCE}\hat{\mathbf{x}}(t) + \mathbf{B}_{PCE}\mathbf{u}(t) \tag{2.5}$$

$$\hat{\mathbf{y}}(t,\xi) = \mathbf{C}_{PCE}(\xi)\hat{\mathbf{x}}(t) + \mathbf{D}_{PCE}(\xi)\mathbf{u}(t) \tag{2.6}$$

This is the dynamic polynomial chaos model in state-space format. The $\mathbf{A}_{PCE}$ matrix and the $\mathbf{B}_{PCE}$ matrices are deterministic. The output equation is the only parameterized result

with $\hat{x}(t;\xi)$ being a function in both the time and stochastic dimensions. The succeeding development of the PCE equations for the 2 ½-DoF quarter-car model will further illustrate the method.

# 3 Quarter Vehicle Model with Uncertain Parameter

For this work the 2 ½-DoF model representing a quarter-car system was transformed into a probabilistic system using the polynomial chaos expansion approach outlined in chapter 2. This chapter will develop a set of PCE state equations based on the quarter-car model diagramed in Figure 2.1.

## *3.1 Deterministic Quarter-Vehicle Model in State-Space Format*

For a 2 ½-DoF quarter-car model that includes tire damping the most common state-space realization includes five states. The vector of states in the state-space equation is

$$\begin{aligned}\mathbf{x}(t) &= \begin{bmatrix} x_s(t) & \dot{x}_s(t) & x_{us}(t) & \dot{x}_{us}(t) & w(t) \end{bmatrix}^T \\ &= \begin{bmatrix} x_1(t) & x_2(t) & x_3(t) & x_4(t) & x_5(t) \end{bmatrix}^T \end{aligned} \tag{3.1}$$

In the state vector $x_s(t)$ and $\dot{x}_s(t)$ are the displacement and velocity states of the sprung mass, respectively. The unsprung mass position and velocity states are $x_{us}(t)$ and $\dot{x}_{us}(t)$, respectively. The displacement input to the model is $w(t)$. With this definition of the deterministic state vector, the state-space equations for a fully deterministic 2 ½-DoF quarter-car model are

$$\underbrace{\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \\ \dot{x}_4(t) \\ \dot{x}_5(t) \end{bmatrix}}_{\dot{\mathbf{x}}(t)} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ \frac{-k_s}{m_s} & \frac{-(b_s+b_0)}{m_s} & \frac{k_s}{m_s} & \frac{b_s}{m_s} & 0 \\ 0 & 0 & 0 & 1 & 0 \\ \frac{k_s}{m_u} & \frac{b_s}{m_u} & \frac{-(k_u+k_s)}{m_u} & \frac{-(b_u+b_s)}{m_u} & \frac{k_u}{m_u} \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \\ x_5(t) \end{bmatrix}}_{\mathbf{x}(t)} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{b_u}{m_u} \\ 1 \end{bmatrix}}_{\mathbf{B}} \dot{w}(t) \tag{3.2}$$

Consistency in the dynamic model requires that the actual excitation of the model be the time derivative of the displacement input $\dot{w}(t)$.

## *3.2 Quarter-Vehicle Model in the Polynomial-Chaos Framework*

By using the deterministic state-space model of (3.2) the PCE state-space model is derived. First, identify the probability distributions of the uncertain parameters. Once a probability density function (pdf) is determined for each uncertain parameter, the proper basis functions may be selected. The basis functions are used to create a series expansion of the uncertain parameters and of the system states [5, 19].

The sprung mass has been selected to be a uniformly distributed uncertain parameter. The sprung mass will be defined here as having a mean value of $m_0$ within the uniform distribution and a maximum variation of $m_1$. Legendre polynomials are the appropriate polynomial basis function for a uniformly distributed parameter. Refer to [26] for Legendre polynomial generating formulas.

The sprung mass is parameterized by a finite series expansion of Legendre polynomials. Only the first two orders of polynomials are needed for the sprung mass parameterization expansion to exactly match the assumed uniform pdf.

$$m_s(\xi) = \sum_{j=0}^{1} m_j \phi_j(\xi) \tag{3.3}$$

By creating a series expansion of the sprung mass the model's states must now be represented by a series expansion [5, 19]. To do this, each $p^{th}$ state of the deterministic model is expanded by the finite summation of the product of the orthogonal basis

functions and the polynomial chaos state trajectories associated with that p[th] state. Thus, the p[th] state and its derivative may be written generally as

$$\hat{x}_p(t,\xi) = \sum_{i=0}^{S} \hat{x}_{p,i}(t)\phi_i(\xi) \tag{3.4}$$

$$\dot{\hat{x}}_p(t,\xi) = \sum_{i=0}^{S} \dot{\hat{x}}_{p,i}(t)\phi_i(\xi) \tag{3.5}$$

As in (2.2), the summation is truncated after S+1 terms so that only the first S orders of the polynomial basis functions are included. In (3.4) and (3.5) $\dot{\hat{x}}_p(t,\xi)$ and $\hat{x}_p(t,\xi)$ represent the estimate for the p[th] state of a state vector with states

$$\begin{bmatrix} x_1(t) & x_2(t) & \dots & x_p(t) & \dots & x_P(t) \end{bmatrix}^T \tag{3.6}$$

In the same equations, $\dot{\hat{x}}_{p,i}(t)$ and $\hat{x}_{p,i}(t)$ are the i[th] PCE state trajectory for the p[th] state of the state vector defined by (3.6).

### 3.2.1 PCE Expansion of the State-Space Model

Substituting the sprung mass parameterization expansion (3.3) and the state vector expansions (3.4) and (3.5) into the state-space model results in a quarter-car model parameterized by $\xi$.

$$\underbrace{\begin{bmatrix} \dot{\hat{x}}_1(t,\xi) \\ \dot{\hat{x}}_2(t,\xi) \\ \dot{\hat{x}}_3(t,\xi) \\ \dot{\hat{x}}_4(t,\xi) \\ \dot{\hat{x}}_5(t,\xi) \end{bmatrix}}_{\dot{\hat{\mathbf{x}}}(t,\xi)} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ \dfrac{-k_s}{m_s(\xi)} & \dfrac{-(b_s+b_0)}{m_s(\xi)} & \dfrac{k_s}{m_s(\xi)} & \dfrac{b_s}{m_s(\xi)} & 0 \\ 0 & 0 & 0 & 1 & 0 \\ \dfrac{k_s}{m_u} & \dfrac{b_s}{m_u} & \dfrac{-(k_u+k_s)}{m_u} & \dfrac{-(b_u+b_s)}{m_u} & \dfrac{k_u}{m_u} \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} \hat{x}_1(t,\xi) \\ \hat{x}_2(t,\xi) \\ \hat{x}_3(t,\xi) \\ \hat{x}_4(t,\xi) \\ \hat{x}_5(t,\xi) \end{bmatrix}}_{\hat{\mathbf{x}}(t,\xi)} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ \dfrac{b_u}{m_u} \\ 1 \end{bmatrix}}_{\mathbf{B}} \dot{w}(t) \tag{3.7}$$

The presence of the uncertain parameter $m_s(\xi)$ in the denominator of the coefficients of the state-matrix prevents the standard procedure for developing the PCE equations. An alternative approach was developed in which (3.7) was rewritten. This was done by pre-multiplying (3.7) by a pseudo mass matrix

$$\mathbf{M}(\xi)=\begin{bmatrix}1 & 0 & 0 & 0 & 0\\ 0 & \sum_{j=0}^{1} m_j\phi_j(\xi) & 0 & 0 & 0\\ 0 & 0 & 1 & 0 & 0\\ 0 & 0 & 0 & m_u & 0\\ 0 & 0 & 0 & 0 & 1\end{bmatrix} \tag{3.8}$$

Modified state and input matrices were defined as

$$\hat{\mathbf{A}}=\begin{bmatrix}0 & 1 & 0 & 0 & 0\\ -k_s & -(b_s+b_{sf}) & k_s & b_s & 0\\ 0 & 0 & 0 & 1 & 0\\ k_s & b_s & -(k_u+k_s) & -(b_u+b_s) & k_u\\ 0 & 0 & 0 & 0 & 0\end{bmatrix} \tag{3.9}$$

$$\hat{\mathbf{B}}=\begin{bmatrix}0 & 0 & 0 & b_u & 1\end{bmatrix}^T \tag{3.10}$$

By performing the pre-multiplication of (3.7) by the pseudo mass matrix in (3.8) and using the definitions of (3.9) and (3.10) the series expansion of the state-space model can be rewritten as

$$\mathbf{M}(\xi)\dot{\hat{\mathbf{x}}}(t,\xi)=\hat{\mathbf{A}}\hat{\mathbf{x}}(t,\xi)+\hat{\mathbf{B}}\mathbf{u}(t) \tag{3.11}$$

The important result of (3.11) is that the uncertain sprung mass parameter is no longer in the divisor. Removing the sprung mass parameter from the divisor makes it possible to evaluate the necessary inner-products required for the Galerkin projection. It was stated

previously that the Galerkin projection approach to finding PCEs is more suitable to real-time applications. With a real-time model being the final goal of this research, the result from (3.11) is therefore very important.

An additional result is that the uncertain parameter is separated from the deterministic model parameters. This is shown in (3.11) by the $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$ matrices which are made up of fixed, deterministic system parameters while the $\hat{\mathbf{M}}(\xi)$ matrix is a function of $\xi$. Separation of the uncertain parameters from the deterministic parameters is useful in simplifying the development of the final PCE model.

### 3.2.2 Projection Definitions for Galerkin Method

Following the Galerkin method as shown in its application to a state-space model in [19] the projection onto the $n^{th}$ basis function can be found. This makes the series expansion of the state-space model solvable by determining the coefficients that weight each PCE state trajectory. For convenience, the necessary inner-product definitions from [19] are provided.

$$\langle \phi_n(\xi), \phi_i(\xi) \rangle = \begin{cases} 0, & n \neq i \\ \delta_{ni}, & n = i \end{cases} \tag{3.12}$$

$$\langle \phi_n(\xi), \phi_i(\xi)\phi_j(\xi) \rangle = \delta_{nij} \tag{3.13}$$

For the case of a time varying input, the projection was defined as

$$\langle \phi_n(\xi), \mathbf{u}(t) \rangle = \begin{cases} 0 & n \neq 0 \\ \delta_{00}\mathbf{u}(t) & n = 0 \end{cases} \tag{3.14}$$

Each projection results in a deterministic value. The set of projections are also written as matrices.

$$[\delta_{ni}] = \underbrace{\begin{bmatrix} \delta_{00} & 0 & \dots & 0 \\ 0 & \delta_{11} & \dots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & \delta_{SS} \end{bmatrix}}_{(S+1)\times(S+1)} \tag{3.15}$$

$$[\delta_{nij}] = \underbrace{\begin{bmatrix} \delta_{00j} & \delta_{01j} & \dots & \delta_{0Sj} \\ \delta_{10j} & \delta_{11j} & \dots & \delta_{1Sj} \\ \vdots & & \ddots & \vdots \\ \delta_{S0j} & \delta_{S1j} & \dots & \delta_{SSj} \end{bmatrix}}_{(S+1)\times(S+1)} \tag{3.16}$$

The size of the matrices shown in (3.15) and (3.16) are expressed in terms of S which is the highest-order polynomial term of the truncated series expansion of the expanded state vectors defined in (3.4) and (3.5). The dimension of (3.15) and (3.16) show that just as there are S+1 polynomial basis functions in each state vector expansions there will be corresponding S+1 rows and columns inner product matrices. To use the matrix in (3.16) a new matrix is defined

$$[\beta_{ni}] = m_0[\delta_{ni0}] + m_1[\delta_{ni1}] \tag{3.17}$$

With the projections now defined as matrices in (3.15), (3.16), and (3.17), the general PCE state-space equations for a 2 ½-DoF quarter-car model with 1 uncertain parameter can now be completed.

### 3.2.3 General PCE State-space Model with One Uncertain Parameter

To find the PCE state equations for the quarter-car system, we first find the Galerkin projection of (3.11). This leads to a representation in a form general to most n-DoF linear systems with only one uncertain parameter.

$$\mathbf{M}_{PCE}\dot{\hat{\mathbf{X}}}(t) = \mathbf{A}_{PCE}\hat{\mathbf{X}}(t) + \mathbf{B}_{PCE}\mathbf{u}(t) \tag{3.18}$$

The PCE state vectors of (3.18) are defined as follows:

$$\dot{\hat{\mathbf{X}}}(t) = \begin{bmatrix}[\dot{\hat{x}}_{1,i}(t)] & [\dot{\hat{x}}_{2,i}(t)] & \dots & [\dot{\hat{x}}_{5,i}(t)]\end{bmatrix}^T \tag{3.19}$$

$$\hat{\mathbf{X}}(t) = \begin{bmatrix}[\hat{x}_{1,i}(t)] & [\hat{x}_{2,i}(t)] & \dots & [\hat{x}_{5,i}(t)]\end{bmatrix}^T \tag{3.20}$$

The following vectors are defined for (3.19) and (3.20):

$$\begin{aligned}
[\hat{x}_{1,i}(t)] &= [\hat{x}_{1,0}(t) \quad \hat{x}_{1,1}(t) \quad \dots \quad \hat{x}_{1,S}(t)]^T \\
[\hat{x}_{2,i}(t)] &= [\hat{x}_{2,0}(t) \quad \hat{x}_{2,1}(t) \quad \dots \quad \hat{x}_{2,S}(t)]^T \\
&\vdots \\
[\hat{x}_{5,i}(t)] &= [\hat{x}_{5,0}(t) \quad \hat{x}_{5,1}(t) \quad \dots \quad \hat{x}_{5,S}(t)]^T
\end{aligned} \tag{3.21}$$

The PCE state trajectories $\dot{\hat{x}}_{p,i}(t)$ and $\hat{x}_{p,i}(t)$ were previously defined in section 3.2 in conjunction with equations (3.4) and (3.5). A new mass matrix, called the PCE mass matrix is written as

$$\mathbf{M}_{PCE} = \begin{bmatrix}
[\delta_{ni}] & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & [\beta_{ni}] & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & [\delta_{ni}] & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & m_u[\delta_{ni}] & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & [\delta_{ni}]
\end{bmatrix} \tag{3.22}$$

The PCE state matrix for the quarter-car model is

$$\mathbf{A}_{PCE} = \begin{bmatrix} \mathbf{0} & [\delta_{ni}] & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -k_s[\delta_{ni}] & -(b_s + b_{sf})[\delta_{ni}] & k_s[\delta_{ni}] & b_s[\delta_{ni}] & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & [\delta_{ni}] & \mathbf{0} \\ k_s[\delta_{ni}] & b_s[\delta_{ni}] & -(k_u + k_s)[\delta_{ni}] & -(b_u + b_s)[\delta_{ni}] & k_u[\delta_{ni}] \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \quad (3.23)$$

The input matrix of the quarter-car model written as a vector is

$$\mathbf{B}_{PCE} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & b_u\delta_{input} & \delta_{input} \end{bmatrix}^T \quad (3.24)$$

Using the projection of the input in (3.14) the vector in (3.24) is defined.

$$\delta_{input} = [\delta_{00} \quad 0 \quad \ldots \quad 0]^T \quad (3.25)$$

As a final step, the $\mathbf{M}_{PCE}$ matrix can be removed from the left side of (3.18) by pre-multiplying by $\mathbf{M}_{PCE}^{-1}$. This is possible only if an inverse exists. By definition of the projection in (3.12) the $[\delta_{ni}]$ matrix is always full rank. A general study of the rank of $[\beta_{ni}]$ was not explored in this study, but for the dimensionalities examined $[\beta_{ni}]$ was also full rank. Based on the full rank nature of $[\delta_{ni}]$ it is very likely that a general proof can be found to show that $[\beta_{ni}]$ is also always full rank. With all the elements of $\mathbf{M}_{PCE}$ full rank, the PCE mass matrix is invertible. The deterministic set of five state equations has now become a set of $5(S+1)$ state equations.

$$\dot{\hat{\mathbf{X}}}(t) = \mathbf{M}_{PCE}^{-1}\mathbf{A}_{PCE}\hat{\mathbf{X}}(t) + \mathbf{M}_{PCE}^{-1}\mathbf{B}_{PCE}\mathbf{u}(t) \quad (3.26)$$

One of the important results of the Galerkin projection is to create a completely deterministic set of state equations that are no longer dependent on the uncertain parameter $\xi$. The result of the inner-products as defined in (3.12) and (3.13) for this realization is a set of fixed values that do not vary for the specified range of uncertain parameter and the truncated series of polynomial basis functions. Therefore, the state and input matrices of the PCE state-space model is made up of constants and can be used in a linear state-space solver. Solving the PCE state equations for all the PCE state trajectories (all $(S+1)$ state trajectories for each state!) can be easily accomplished in a linear state-space solver. The computed PCE state trajectories may then be transformed into the physical domain to produce any of the physical states of interest.

### 3.2.4 PCE Output Equations

The complete set of computed PCE state trajectories is mapped back to the five states representative of the physical domain by expressing the output equations in terms of the orthogonal basis functions. Referring back to (3.4), an output equation of each of the 5 states may be obtained.

$$\hat{\mathbf{x}}(t,\xi) = \underbrace{\begin{bmatrix} \Phi(\xi) & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \Phi(\xi) & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \Phi(\xi) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \Phi(\xi) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \Phi(\xi) \end{bmatrix}}_{5\times 5(S+1)} \hat{\mathbf{X}}(t) \tag{3.27}$$

A vector of polynomial basis functions used in (3.27) can be defined.

$$\Phi(\xi) = \begin{bmatrix} \phi_0(\xi) & \phi_1(\xi) & \dots & \phi_S(\xi) \end{bmatrix} \tag{3.28}$$

The output equation of (3.27) is another way of writing (3.4). The dimension of the vector of polynomial basis functions in (3.28) is $1\times(S+1)$.

For the experimental quarter-car rig used here, the outputs are accelerations. The PCE expression of quarter-car accelerations is

$$\hat{\mathbf{y}}(t,\xi)=\begin{bmatrix}\dot{\hat{x}}_2(t,\xi)\\ \dot{\hat{x}}_4(t,\xi)\end{bmatrix} \tag{3.29}$$

The PCE output equation of (3.29) can be rewritten.

$$\begin{aligned}&\underbrace{\begin{bmatrix}m_0+m_1\xi & 0\\ 0 & m_u\end{bmatrix}}_{\mathbf{M}_{PCE,output}(\xi)}\begin{bmatrix}\dot{\hat{x}}_2(t,\xi)\\ \dot{\hat{x}}_4(t,\xi)\end{bmatrix}=\\ &\underbrace{\begin{bmatrix}-k_s\Phi(\xi) & -(b_s+b_0)\Phi(\xi) & k_s\Phi(\xi) & b_s\Phi(\xi) & 0\\ k_s\Phi(\xi) & b_s\Phi(\xi) & -(k_u+k_s)\Phi(\xi) & -(b_u+b_s)\Phi(\xi) & k_u\Phi(\xi)\end{bmatrix}}_{\mathbf{C}_{PCE}(\xi)}\hat{\mathbf{X}}(t)\\ &+\underbrace{\begin{bmatrix}0 & b_u\end{bmatrix}^T}_{\mathbf{D}_{PCE}^T(\xi)}\mathbf{u}(t)\end{aligned} \tag{3.30}$$

Remember that $\Phi(\xi)$ is defined in (3.28) as a row vector. The $\mathbf{M}_{PCE,output}$ matrix can be removed from the left side of the equation similar to the derivation for (3.26). In this case, the inverse of $\mathbf{M}_{PCE,output}(\xi)$ is proven to exist by inspection.

$$\hat{\mathbf{y}}(t,\xi)=\mathbf{M}_{PCE,output}^{-1}(\xi)\mathbf{C}_{PCE}(\xi)\hat{\mathbf{X}}(t)+\mathbf{M}_{PCE,output}^{-1}(\xi)\mathbf{D}_{PCE}(\xi)\mathbf{u}(t) \tag{3.31}$$

The $\mathbf{C}_{PCE}(\xi)$ can alternatively be defined as the product of a matrix of constants and a matrix of $\Phi(\xi)$ vectors. This is an important definition for later real-time development.

$$\mathbf{C}_{PCE}(\xi)=\underbrace{\begin{bmatrix} -k_s & -(b_s+b_{sf}) & k_s & b_s & 0 \\ k_s & b_s & -(k_u+k_s) & -(b_u+b_s) & k_u \end{bmatrix}}_{2\times 5}\underbrace{\begin{bmatrix} \Phi(\xi) & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \Phi(\xi) & & \vdots \\ \vdots & & \ddots & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{0} & \Phi(\xi) \end{bmatrix}}_{5\times 5(S+1)} \quad (3.32)$$

In summary, (3.26) is a continuous time deterministic state-space model formulated in the PCE framework. A stochastic continuous time output equation for the sprung and unsprung mass accelerations in given by equation (3.31).

# 4 Adaptive PCE Formulation

As discussed in the literature review, there are many methods and algorithms for estimating parameters on a dynamic system. Matlab has several tools readily available for offline system identification. The scope of this proof of concept study does not include comparing the proposed method to existing parameter estimation methods.

This chapter will present development of an adaptive parameter estimation algorithm that follows earlier work presented by Southward [19]. The preceding derivations of the PCE quarter-car model replace the uncertain mass parameter with the stochastic parameter $\xi$. The goal of the adaptive parameter algorithm is to update $\xi$ such that the output errors of the plant and the PCE model are minimized. A block diagram of the proposed adaptive method is shown in Figure 4.1.

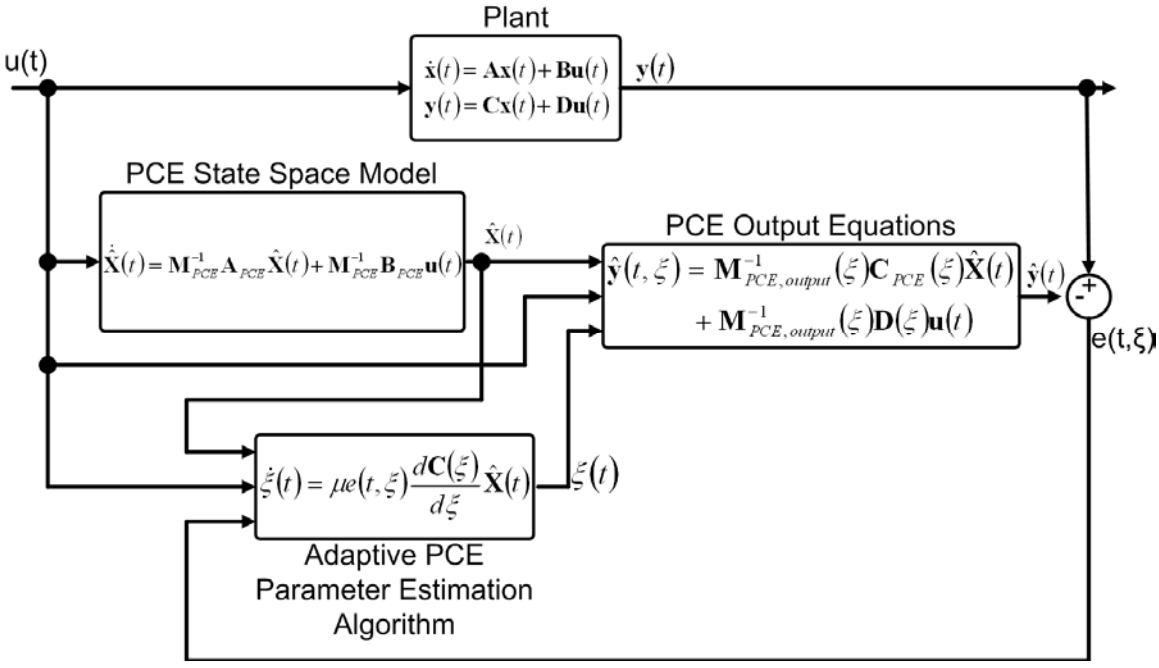


***Figure 4.1 Block diagram of adaptive PCE parameter estimation method.***

The PCE state-space model dynamically estimates the PCE state trajectories for the quarter-car system. The state trajectories are wired to the PCE output equation block and the adaptive parameter estimation block. The PCE output block receives the PCE state trajectories, excitation, and current parameter estimate in order to estimate the current output of the plant. The adaptive parameter estimation block receives the PCE state trajectories, excitation, and error signals and creates a new estimate by minimizing a cost function. That cost function will be discussed in the following section.

## *4.1 PCE Adaptation Law*

The measured signals of the experimental quarter-car rig used for this study are the accelerations of the sprung mass and the unsprung mass. The PCE quarter-car model has also been formulated with an output equation that predicts the acceleration of both the sprung mass and the unsprung mass. With the accelerometer channels and output channels two output errors were constructed. The objective is to actively update $\xi$ so that the output errors are minimized. That minimization is done with a positive-definite cost function which is the weighted sum of the squared errors.

$$J(t,\xi) = ae_s^2(t,\xi) + be_{us}^2(t,\xi) \tag{4.1}$$

The error signal $e_s(t,\xi)$ is the error from the output sprung mass acceleration and the acceleration measured by the sprung mass accelerometer. The error signal $e_{us}(t,\xi)$ is the error of the predicted unsprung mass acceleration and the measured acceleration from the unsprung mass accelerometer channel.

While there are many adaptation algorithms for adaptive parameter estimation, in following the work by Southward [19], a simple parameter update law, the MIT rule [21], was chosen. The MIT rule defines the gradient of the adjustable parameter as a function of the derivative of the cost function with respect to the parameter $\xi$ [21]. This is the adaptation law.

$$\dot{\xi}(t) = -\mu \frac{\partial J(t,\xi)}{\partial \xi} \tag{4.2}$$

Analytically, the derivative of the cost function with respect to the uncertain parameter $\xi(t)$ is dependent on several derivatives with respect to $\xi$.

$$\frac{\partial J(t,\xi)}{\partial \xi} = 2ae_s(t,\xi)\frac{\partial e_s(t,\xi)}{\partial \xi} + 2be_{us}(t,\xi)\frac{\partial e_{us}(t,\xi)}{\partial \xi} \tag{4.3}$$

The derivatives of the error signals are dependent on the PCE output equation matrices. Referring back to (3.31), the output equation can be rewritten.

$$\hat{\mathbf{y}}(t,\xi) = \underbrace{\begin{bmatrix}\mathbf{C}_1(\xi)\\ \mathbf{C}_2(\xi)\end{bmatrix}}_{\mathbf{M}_{PCE,output}^{-1}(\xi)\mathbf{C}_{PCE}(\xi)} \hat{\mathbf{X}}(t) + \underbrace{\begin{bmatrix}\mathbf{D}_1(\xi)\\ \mathbf{D}_2(\xi)\end{bmatrix}}_{\mathbf{M}_{PCE,output}^{-1}(\xi)\mathbf{D}_{PCE}(\xi)} \mathbf{u}(t) \tag{4.4}$$

The equation (4.4) is equivalent to the output equation of (3.31). The partial derivative of the error signals can now be written in terms of $\mathbf{C}_1(\xi)$ and $\mathbf{C}_2(\xi)$.

$$\frac{\partial e_s(t,\xi)}{\partial \xi} = -\frac{d\mathbf{C}_1(\xi)}{d\xi}\hat{\mathbf{X}}(t) - \frac{d\mathbf{D}_1(\xi)}{d\xi}\mathbf{u}(t) \tag{4.5}$$

$$\frac{\partial e_{us}(t,\xi)}{\partial \xi} = -\frac{d\mathbf{C}_2(\xi)}{d\xi}\hat{\mathbf{X}}(t) - \frac{d\mathbf{D}_2(\xi)}{d\xi}\mathbf{u}(t) \tag{4.6}$$

It is clearly seen that the partial derivative of the error signals is dependent on the derivative of the PCE output matrices with respect to $\xi$. Since the PCE output matrices are known and defined in (3.30) and (3.31) the analytical derivatives can be found. For each row of the PCE output equation matrices the derivatives are now defined.

$$\frac{d\mathbf{C}_1(\xi)}{d\xi} = \left[ -\frac{k_s}{m(\xi)}\left(\frac{d\Phi(\xi)}{d\xi} - m_1\frac{\Phi(\xi)}{m(\xi)}\right) \quad \cdots \quad \frac{b_s}{m(\xi)}\left(\frac{d\Phi(\xi)}{d\xi} - m_1\frac{\Phi(\xi)}{m(\xi)}\right) \quad \mathbf{0} \right] \tag{4.7}$$

$$\frac{d\mathbf{C}_2(\xi)}{d\xi} = \left[ \frac{k_s}{m_u}\frac{d\Phi(\xi)}{d\xi} \quad \cdots \quad \frac{k_u}{m_u}\frac{d\Phi(\xi)}{d\xi} \right] \tag{4.8}$$

Similar to (3.32) an alternative representation of $\frac{d\mathbf{C}_1(\xi)}{d\xi}$ and $\frac{d\mathbf{C}_2(\xi)}{d\xi}$ was found which will be later used to facilitate real-time implementation.

$$\begin{aligned}\frac{d\mathbf{C}_1(\xi)}{d\xi} = {} & \tfrac{1}{m_s(\xi)}\begin{bmatrix} -k_s & -(b_s+b_{sf}) & k_s & b_s & 0 \end{bmatrix}\begin{bmatrix} \frac{d\Phi(\xi)}{d\xi} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \frac{d\Phi(\xi)}{d\xi} & & \vdots \\ \vdots & & \ddots & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \frac{d\Phi(\xi)}{d\xi} \end{bmatrix} \\ & -\tfrac{m_1}{m_s^2(\xi)}\begin{bmatrix} -k_s & -(b_s+b_{sf}) & k_s & b_s & 0 \end{bmatrix}\begin{bmatrix} \Phi(\xi) & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \Phi(\xi) & & \vdots \\ \vdots & & \ddots & \\ \mathbf{0} & \cdots & \mathbf{0} & \Phi(\xi) \end{bmatrix}\end{aligned} \tag{4.9}$$

$$\frac{d\mathbf{C}_2(\xi)}{d\xi} = \tfrac{1}{m_{us}}\begin{bmatrix} k_s & b_s & -(k_u+k_s) & -(b_u+b_s) & k_u \end{bmatrix}\begin{bmatrix} \frac{d\Phi(\xi)}{d\xi} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \frac{d\Phi(\xi)}{d\xi} & & \vdots \\ \vdots & & \ddots & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \frac{d\Phi(\xi)}{d\xi} \end{bmatrix} \tag{4.10}$$

Looking closer at the $\mathbf{D}_{PCE}(\xi)$ matrix of (3.31), sometimes called the direct feed through matrix, it can be seen that in this case it is actually constructed of constants parameters and is not a function of $\xi$. This independence from the uncertain parameter $\xi$ stems

from the separation of the uncertain parameters as was explained in section 3.2.1. The derivatives for each row of the feed through matrix for this case are simply zero.

$$\frac{d\mathbf{D}_1(\xi)}{d\xi} = 0 \tag{4.11}$$

$$\frac{d\mathbf{D}_2(\xi)}{d\xi} = 0 \tag{4.12}$$

Substituting the results from (4.3), (4.7), (4.8), (4.11), and (4.12) into the MIT rule defined in (4.2) will result in the analytical form for the gradient of the uncertain parameter $\xi$. In addition to applying the MIT rule, a limiting factor is placed on the algorithm so that the estimate $\xi$ never exceeds its defined range of ±1. For clarity $\xi$ will be represented in the parameter update law explicitly dependent on time as $\xi(t)$.

$$\dot{\xi}(t) = \begin{cases} \mu_1 e_s(t,\xi(t)) \dfrac{d\mathbf{C}_1(\xi(t))}{d\xi(t)} \hat{\mathbf{X}}(t) + \mu_2 e_{us}(t,\xi(t)) \dfrac{d\mathbf{C}_2(\xi(t))}{d\xi(t)} \hat{\mathbf{X}}(t) & |\xi(t)| < 1 \\ 0 & |\xi(t)| = 1 \end{cases} \tag{4.13}$$

The single gradient step size in (4.2) becomes two gradient step sizes as an effect of the linear combination of the error signals in the cost function.

$$\mu_1 = 2a\mu \tag{4.14}$$

$$\mu_2 = 2b\mu \tag{4.15}$$

Integrating $\dot{\xi}(t)$ from (4.13) will produce a trajectory estimate for the uncertain parameter $\xi(t)$. Notice in (4.13) that the parameter update law does not actually require knowledge of the excitation as was shown in Figure 4.1. This is because the derivative of the feed through matrix is zero and is not in the equation. If the unsprung mass were

assumed to be unknown, the derivative of the feed through matrix would no longer be zero and knowledge of the excitation would be required. Thus Figure 4.1 is of a more general case.

## *4.2 Real-Time Output Equation Update Using Matrix Products*

Looking back at the PCE output equations defined in (3.31) we see that for each estimate of $\xi$ the output equation needs to be completely updated. Referring to the PCE parameter adaptation law defined in (4.13) it is seen that it requires matrix derivatives be evaluated for each estimate of $\xi$. This is a problem for fast implementation of the algorithm because each output matrix or derivative updated at the current value of $\xi$ will be computationally costly. A fast and efficient method for computing both the output matrix and the derivative of the output matrix has been developed and is presented now.

### 4.2.1 Legendre Polynomial Coefficient Matrix

A novel way to efficiently evaluate the set of Legendre polynomials in a real-time application was developed. The output equations were rearranged to pre-compute the coefficients of the Legendre polynomials independent of the parameter $\xi$. Using the pre-computed matrices, each update to the output and feed through matrices is obtained by simple matrix multiplication and addition operations.

To explain, the n+1 order Legendre polynomial function, $L_{n+1}(\xi)$, is computed numerically using a rearranged form of the recursion relationship found in [26].

$$L_{n+1}(\xi) = \frac{(2n+1)\xi L_n(\xi) - nL_{n-1}(\xi)}{n+1} \tag{4.16}$$

To achieve a format conducive to matrix-vector products, the recursion relationship of (4.16) was modified. This was done by defining a row vector $\mathbf{q}$ which consists of $\xi^{\nu}$ for $\nu = 0, 1, \ldots, S$ and by defining a column vector of polynomial coefficients.

$$\mathbf{q}(\xi) = \begin{bmatrix} 1 & \xi & \ldots & \xi^{S} \end{bmatrix} \tag{4.17}$$

$$\mathbf{P}_n = \begin{bmatrix} p_{0,n} & p_{1,n} & \ldots & p_{S,n} \end{bmatrix}^T \tag{4.18}$$

The elements of $\mathbf{P}_n$ in (4.18), $p_{i,n}$, are the i[th] polynomial coefficient corresponding to $\xi^i$ of the n[th] order polynomial function. As an example, the first two polynomials of the Legendre set, $L_0(\xi)$ and $L_0(\xi)$ can be easily written as a vector matrix product.

$$L_0(\xi) = \mathbf{q}(\xi) \cdot \mathbf{P}_0 = 1 \tag{4.19}$$

$$L_1(\xi) = \mathbf{q}(\xi) \cdot \mathbf{P}_1 = \xi \tag{4.20}$$

The polynomial coefficient vectors of (4.19) and (4.20) are

$$\mathbf{P}_0 = \begin{bmatrix} 1 & 0 & \ldots & 0 \end{bmatrix}^T \tag{4.21}$$

$$\mathbf{P}_1 = \begin{bmatrix} 0 & 1 & 0 & \ldots & 0 \end{bmatrix}^T \tag{4.22}$$

Extending the vector notation to the terms in (4.16) the $L_{n-1}(\xi)$ term can also be written as a product of vectors $\mathbf{q}(\xi)$ and $\mathbf{P}_n$.

$$L_{n-1}(\xi) = \mathbf{q}(\xi) \cdot \mathbf{P}_n \tag{4.23}$$

The $\xi L_n(\xi)$ term in (4.16) was defined in a similar fashion by using a shifted $\mathbf{P}_n$ vector to account for the $\xi L_n(\xi)$ product.

$$L_n(\xi) = \mathbf{q}(\xi) \cdot \mathbf{P}'_n \tag{4.24}$$

$$\mathbf{P}'_n = \begin{bmatrix} 0 & p_{0,n} & p_{1,n} & \cdots & p_{(S-1),n} \end{bmatrix}^T \tag{4.25}$$

By substituting (4.17), (4.18), and (4.23) into (4.16) the n[th] order Legendre polynomial can be computed as follows:

$$L_{n+1}(\xi) = \mathbf{q}(\xi) \cdot \mathbf{P}_{n+1} = \frac{(2n+1)\mathbf{q}(\xi)\mathbf{P}'_n - n\mathbf{q}(\xi)\mathbf{P}_{n-1}}{n+1} \tag{4.26}$$

By inspection of (4.26) a formula for computing the matrix of Legendre polynomial coefficients was devised. The matrix $\mathbf{P}$ is a matrix of vectors $\mathbf{P}_n$ computed by the modified formula.

$$\mathbf{P}_{n+1} = \frac{(2n+1)\mathbf{P}'_n - n\mathbf{P}_{n-1}}{n+1} \tag{4.27}$$

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_0 & \mathbf{P}_1 & \cdots & \mathbf{P}_S \end{bmatrix} \tag{4.28}$$

As an example, a set of 4[th] order Legendre polynomial coefficients would look as follows:

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & -\frac{1}{2} & 0 & \frac{3}{8} \\ 0 & 1 & 0 & -\frac{3}{2} & 0 \\ 0 & 0 & \frac{3}{2} & 0 & -\frac{15}{4} \\ 0 & 0 & 0 & \frac{5}{2} & 0 \\ 0 & 0 & 0 & 0 & \frac{35}{8} \end{bmatrix} \tag{4.29}$$

The product of vector $\mathbf{q}(\xi)$ and the $\mathbf{P}$ matrix can now be used to express the same $\Phi(\xi)$ presented in (3.28) as a product of a data vector and a matrix of constant polynomial basis function coefficients.

$$\Phi(\xi) = \begin{bmatrix} 1 & \xi & \dots & \xi^S \end{bmatrix} \begin{bmatrix} P_0 & P_1 & \dots & P_S \end{bmatrix} = \begin{bmatrix} \phi_0(\xi) & \phi_1(\xi) & \dots & \phi_S(\xi) \end{bmatrix} \quad (4.30)$$

By pre-assembling a matrix of Legendre polynomial coefficients **P** and computing in real-time the vector $\mathbf{q}(\xi)$ at the current estimate of $\xi$, the $\Phi(\xi)$ vector of (4.30) can be efficiently computed and substituted into the output equations defined by (3.30), (3.31), and (3.32). This is a crucial step to making a real-time implementation of the PCE model more efficient.

In addition to having real-time computable output equations, the PCE adaptation law from (4.13) requires updates of the derivatives of $\mathbf{C}_1(\xi)$ and $\mathbf{C}_2(\xi)$. To compute these derivatives we need the derivative of (4.30). To take the derivative $\frac{d\Phi(\xi)}{d\xi}$, a similar process to that described in the development of (4.30) is followed.

$$\frac{d\Phi(\xi)}{d\xi} = \begin{bmatrix} 0 & 1 & 2\xi & \dots & S\xi^{S-1} \end{bmatrix} \begin{bmatrix} P_0 & P_1 & \dots & P_S \end{bmatrix} \quad (4.31)$$

The resulting equations (4.30) and (4.31) show that only the $\mathbf{q}(\xi)$ vector need be updated while the **P** matrix for the set of Legendre functions up to order S remains unchanged for all defined values of $\xi$. To further illustrate the ability to implement (4.30) and (4.31) in a real-time environment, Simulink implementations of this method are presented in Figure 4.2 and Figure 4.3.

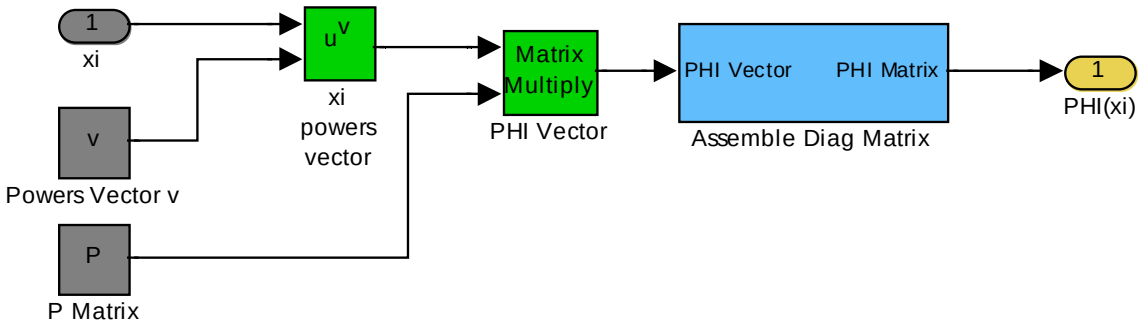


***Figure 4.2 Simulink implementation of simplified*** $\Phi(\xi)$***.***

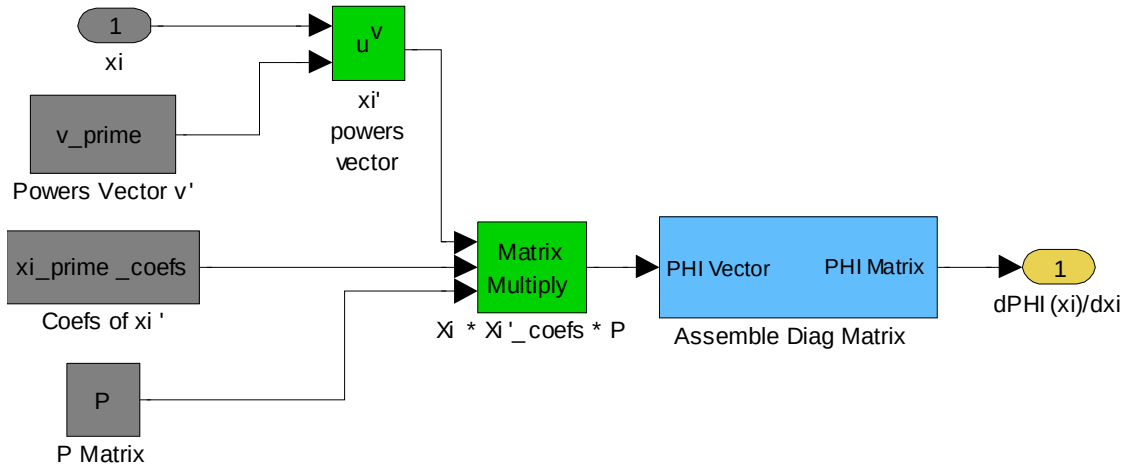


***Figure 4.3 Simulink implementation of simplified*** $\frac{d\Phi(\xi)}{d\xi}$***.***

To summarize, real-time implementation of the PCE output equations is made possible by the real-time updatable equations (4.30) and (4.31). Because of the construction of the output equations (3.31) they do not need to be explicitly put into discrete time.

## *4.3 Discretization of PCE Equations and Adaptation Law*

The quarter-car model is completely represented as a continuous time PCE state equation model described by (3.26). In order to implement the continuous time PCE quarter-car model in a real-time environment the model needs to be transformed to discrete-time. For this study a numerical approach was taken by using the C2D function in Matlab to

transform the continuous time PCE state-space model into discrete time using the Tustin (bilinear) method. The numerical approach was possible because the PCE state equations are deterministic. The Tustin method transforms a transfer function in the Laplace "s" domain, which corresponds to a continuous-time state-space model, into a discrete time "z" domain model. That transformation is done by redefining the "s" term in the transfer function in terms of "z" and the sample time, $T_s$ [27].

$$s = \left(\frac{2}{T_s}\right)\left(\frac{z-1}{z+1}\right) \tag{4.32}$$

By using the Matlab C2D function the state-space model described by (3.26) is numerically transformed from its continuous-time form to a discrete-time state-space form using the Tustin approximation as described in (4.32).

The first order PCE parameter adaptation law (4.13) also needs to be discretized. At each time step the next estimate of the parameter $\xi$ is determined by the current estimate $\xi_k$ adjusted by the current time step gradient $\nabla_k$.

$$\xi_{k+1} = \xi_k + \nabla_k \tag{4.33}$$

The current time step gradient is defined similar to the (4.13).

$$\nabla_k = \begin{cases} \mu_1 e_{s,k}(\xi_k)\dfrac{d\mathbf{C}_1(\xi_k)}{d\xi}\hat{\mathbf{X}}_k + \mu_2 e_{us,k}(\xi_k)\dfrac{d\mathbf{C}_2(\xi_k)}{d\xi}\hat{\mathbf{X}}_k & |\xi_k| < 1 \\ 0 & |\xi_k| = 1 \end{cases} \tag{4.34}$$

The current sample of the error signals are $e_{s,k}(\xi_k)$ and $e_{us,k}(\xi_k)$ for the sprung and unsprung mass, respectively. The PCE state trajectory at sample k is $\hat{\mathbf{X}}_k$. The

derivatives $\frac{d\mathbf{C}_1(\xi_k)}{d\xi}$ and $\frac{d\mathbf{C}_2(\xi_k)}{d\xi}$ are not explicitly discrete but are nevertheless evaluated at the current estimate, $\xi_k$. A Simulink representation of the discrete time parameter estimation routine including the PCE state estimates and output estimates is shown in Figure 3.2.

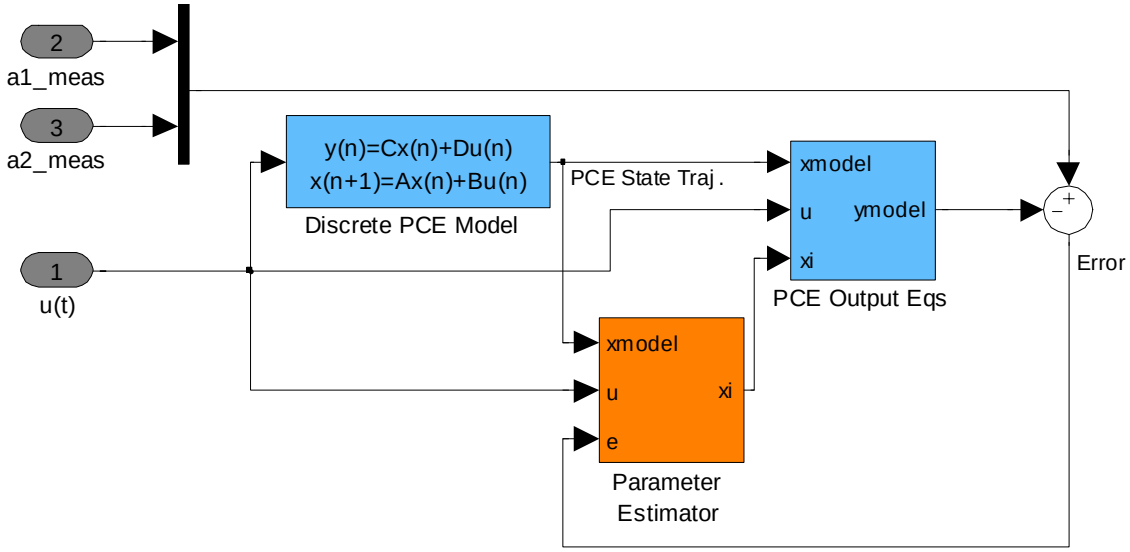


***Figure 4.4 Simulink implementation of adaptive PCE algorithm for real-time.***

At this point the quarter-car has been modeled with a discrete time PCE state-space model. The PCE output equations have been presented, including a novel approach to creating a real-time update of both the output equations and their derivative with respect to $\xi$. A discrete time PCE adaptation law was introduced and Simulink implementations of the adaptive PCE parameter estimation algorithm shown. Now the work will turn an eye toward simulation and experimentation of the novel adaptive PCE algorithm.

# 5 Experimental Test Bed Parameterization

The quarter-car rig developed for the Performance Engineering and Research Lab described in [11, 12] was used to validate the proposed parameter estimation algorithm. The rig shown in Figure 5.1 and consists of a large steel plate attached to a large frame. Steel guides with roller bearings limit relative motion to the vertical direction. The steel plate has steel bars attached to the rear. The steel bars may be removed or additional bars added in order to change the sprung mass. The steel plate and attached masses are the sprung mass represented in the model by $m_s$. The friction in the roller bearings between the plate-mass and the support frame of the rig is represented in the model by the linear damping element $b_0$.

Attached to the steel plate on the front of the rig is the rear McPherson strut suspension from a Porsche 996 racecar. In [13] it was shown that for the Porsche 996 McPherson strut suspension with a small displacement input, a linear model provides an acceptable response estimate. Since the linear model is valid, the spring and shock absorber of the suspension are represented in the model by the linear spring element $k_s$ and the linear viscous damping element $b_s$, respectively. The model's unsprung mass, $m_{us}$ represents the mass of the wheel and tire assembly. The spring and damping elements between the ground and the unsprung mass shown in the model as $k_{us}$ and $b_{us}$ represent tire stiffness and tire damping, respectively.

The quarter-car rig is excited by a hydraulic actuator placed underneath the tire. The vertical displacement imposed on the tire patch simulates road inputs to the tire.

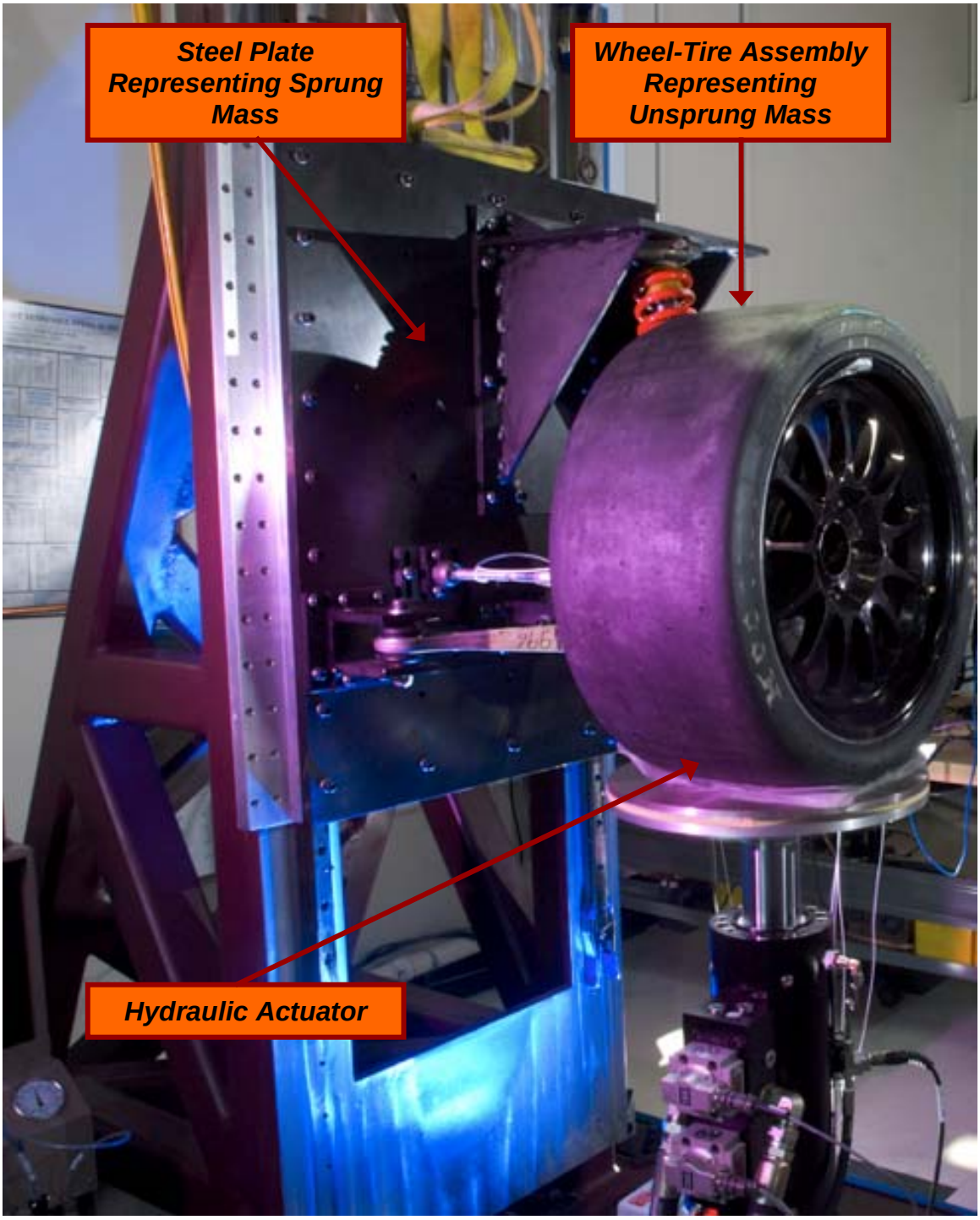


***Figure 5.1 Quarter-car rig used for experimental validation of the proposed algorithm.***

The hydraulic actuator is an MTS 5.5 Kip Linear Hydraulic Actuator. It is controlled by a FlexTest SE Controller which receives a position waveform from a dSpace unit. In

dSpace a random velocity waveform is created and subsequently integrated using a 2-complex-pole integrator [11] to create the displacement waveform sent to the hydraulic controller. For further information on the rig and its characteristics see [11, 12].

Two PCB accelerometers were collocated on the rig, one on the steel plate (sprung mass) and the other on the wheel-tire assembly (unsprung mass). The two channels are wired to a IOTECH Wavebook 516E signal conditioner where the signal is first filtered by a low-pass filter to prevent aliasing and also by a high-pass filter to eliminate any DC offset. The filtered accelerometer signal data is then input to the dSpace Autobox.

When identifying the system parameters, the processed accelerometer data is only captured by the Autobox for offline analysis and identification. When using this system for the experimental validation, the Autobox samples the accelerometer data in real-time to run the adaptive PCE parameter estimation.

## *5.1 Experimentally Determined Quarter-Car Parameters*

The PCE model developed for the quarter-car has been built on the assumption that all the parameters except for the sprung mass are known. In reality, none of the system parameters are known and must be estimated experimentally before hand. For estimation of those parameters the linear state-space model parameters were optimized such that the predicted response matches the actual response for a random excitation.

To measure the response of the rig to a known input, the experimental rig was excited by random wheel displacements with a maximum displacement of ±1inch. In Simulink the random excitation waveform was generated and shaped by a $2^{nd}$ order Butterworth filter

with a 5 Hz cutoff frequency. That waveform was then used by the MTS controller to excite the hydraulic actuator at the wheel-pan as was described previously. Ninety seconds of sprung mass and unsprung mass acceleration data was captured. The data was used by an iterative, offline routine that found the best set of model parameters to match the measured response. The FMINCON function in the Matlab optimization toolbox was used for finding the optimal model parameters. See the works of Andersen and Ziegenmeyer for more information on system identification of quarter-car rigs using this method [11, 13].

***Table 5.1 System parameters estimated by offline system ID.***

| $m_s$ | $k_s$ | $b_s$ | $m_{us}$ | $k_{us}$ | $b_{us}$ | $b_0$ |
|---|---|---|---|---|---|---|
| *257 kg* | *217525 N/m* | *3912 N-s/m* | *30 kg* | *558083 N/m* | *465 N-s/m* | *2031 N-s/m* |

The estimated system parameters from the offline system identification are listed in Table 5.1. To assess the accuracy of the estimated parameters a comparison of the modeled response using the estimated parameters was made against the measured response.

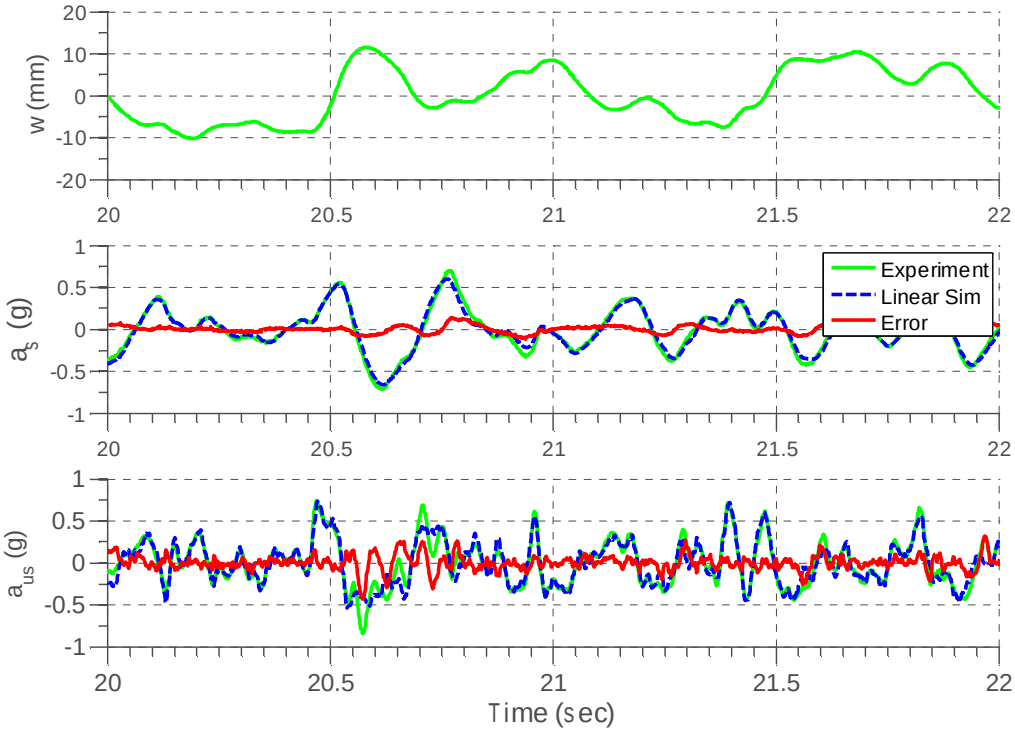


***Figure 5.2 Experimentally measured quarter-car sprung and unsprung accelerations compared to response predicted by a linear model using parameters estimated by the FMINCON system ID method.***

In Figure 5.2 the predicted sprung mass response appears to match reasonably well the measured data. There does appear to be error in the unsprung mass response. This error is most likely due to some non-linearity of the system. By looking closely at the input from just after 20.5 seconds it can be seen that there is a large stroke in the displacement. Immediately following this large stroke more noticeable error is present in the response of the unsprung mass. In light of this, the error can be explained as being excitation dependent. Nevertheless, the error is not unreasonable and the large displacement excitation is more representative of actual road inputs on a car suspension than a more modest displacement would be.

# 6 Simulation Studies of the Adaptive PCE Algorithm

Preliminary evaluation of the performance of the adaptive PCE parameter estimation routine was by simulating the algorithm under various conditions. The ability of the algorithm to initialize at an incorrect value of the sprung-mass and converge to the neighborhood of the correct value was tested. In each simulation all the known parameters of stiffness, damping and unsprung mass were considered constant. The parameters of the quarter-car experimental rig as estimated by the system ID were used in simulation.

Simulations were done using models of the plant and adaptive PCE method created in Simulink. The simulations were setup to create mimic the future experimental test environment. To do this the Simulink models were setup to run with the discrete time solver (no continuous states). Simulations ran with a 1000 Hz sample rate. The model was excited by a random velocity corresponding to a maximum total displacement of ±1 inch. The excitation was filtered by a $2^{nd}$ order Butterworth filter with a 5 Hz cutoff frequency. Measurement noise from the simulated plant was approximated by adding to the simulated plant output a band-limited white-noise signal. The measurement noise generator had a noise power proportional to the random excitation input.

The adaptive algorithm gradient step sizes, $\mu_1$ and $\mu_2$ were normalized with respect to the random excitation noise power. The nominal mass $m_0$ was selected to be 300 kg with a variance, $m_1$ of 100 kg. The PCE expansion was expanded to the seventh order polynomial term. In terms of (3.4) and (3.5) $S = 7$. This produces eight PCE state trajectories for each of the five physical states for a total of 40 PCE states. In each

simulation, the adaptive parameter estimation routine is enabled 5 seconds after the simulation begins in order to show that it is the method that is truly converging and not a result of system dynamics.

***Table 6.1 Simulation conditions for each evaluation case.***

| | Initial Sprung Mass Estimate | True Sprung Mass | $\mu_1$ | $\mu_2$ |
|---|---|---|---|---|
| ***Case 1*** | *220 kg* | *380 kg* | *1e-5* | *0* |
| ***Case 2*** | *380 kg* | *220 kg* | *1e-5* | *0* |
| ***Case 3*** | *300 kg* | *257 kg* | *1e-5* | *0* |
| ***Case 4*** | *300 kg* | *257 kg* | *1e-5* | *1e-5* |
| ***Case 5*** | *300 kg* | *257 kg* | *2e-5* | *0* |
| ***Case 6*** | *300 kg* | *257 kg* | *2e-5* | *1e-5* |
| ***Case 7*** | *300 kg* | *283 kg* | *2e-5* | *0* |
| ***Case 8*** | *300 kg* | *309 kg* | *2e-5* | *0* |

The simulation study sought to determine the effects of different testing conditions. First was the convergence ability of the algorithm with a large initial error of the estimate and verify that once converged, the algorithm is stable. This was done by exciting a the quarter-car plant with all the deterministic parameters as specified by the system ID while the sprung mass parameter was set to be either far above or below the nominal value. After five seconds of simulation the adaptation algorithm was turned on with an initial sprung mass estimate exactly opposite the true value with respect to the nominal sprung mass. For example, simulation case 1 had the true sprung mass value for the plant set 80 kg above the nominal sprung mass at 380kg while the initial estimate was set 80 kg below the nominal at 220 kg.

The second goal was to gauge the effect of adaptation step size. This was done by maintaining all the deterministic parameters at the specified values from the system ID and keeping the value of the simulated plant sprung mass set at 257 kg. The adaptation

step sizes $\mu_1, \mu_2$ were varied from one simulation run to another and the convergence of the estimate onto the true value was analyzed for each simulation.

The final aim was to test the ability of the algorithm to converge to any true value within the range of the probability distribution was verified. To accomplish this, the sprung mass of the simulated plant was set for each run to one of three values, 257 kg, 283 kg, or 309 kg. The adaptive estimation algorithm was turned on five seconds into the simulation and allowed to adapt to see if it converged to the correct mass. For this set of simulations the initial estimate was always 300 kg. For example, in simulation case 8 the sprung mass of the simulated plant was set to 309 kg and the adaptation algorithm was expected to converge to 309 kg and no other value. Table 6.2 has a break down of the test cases according to the effect they illustrate. To recap, the adaptation parameters that were varied between the simulations are

- Initial estimate of the uncertain sprung mass parameter, $\hat{m}_0$.
- True value of the uncertain sprung mass parameter, $m_{true}$
- Adaptive algorithm gradient step-sizes, $\mu_1, \mu_2$

***Table 6.2 Simulation categories and corresponding data and figures.***

| Simulation Category | Simulations Included in Category | Figures of Simulation Results |
|---|---|---|
| ***Variation of Initial Estimate*** | *1, 2* | *Figure 6.1* |
| ***Effect of step-size on estimate convergence*** | *3, 4, 5, 6* | *Figure 6.2* |
| ***Different "true" masses*** | *5, 7, 8* | *Figure 6.4* |

The simulation results show that even with added noise the estimate converges to the true value and maintains itself near the true value even in the presence of noise. The

following figures show results of the simulation studies. In each simulation shown the adaptive estimation algorithm was activated after 5 seconds of simulation in order to see clearly the initial estimate of the algorithm.

## *6.1 Simulated Variation of Initial Sprung Mass Estimate*

The first set of simulations kept the step sizes the same. The initial estimate of the sprung mass was at one extreme of the range of the distribution while the true sprung mass was at the other extreme.

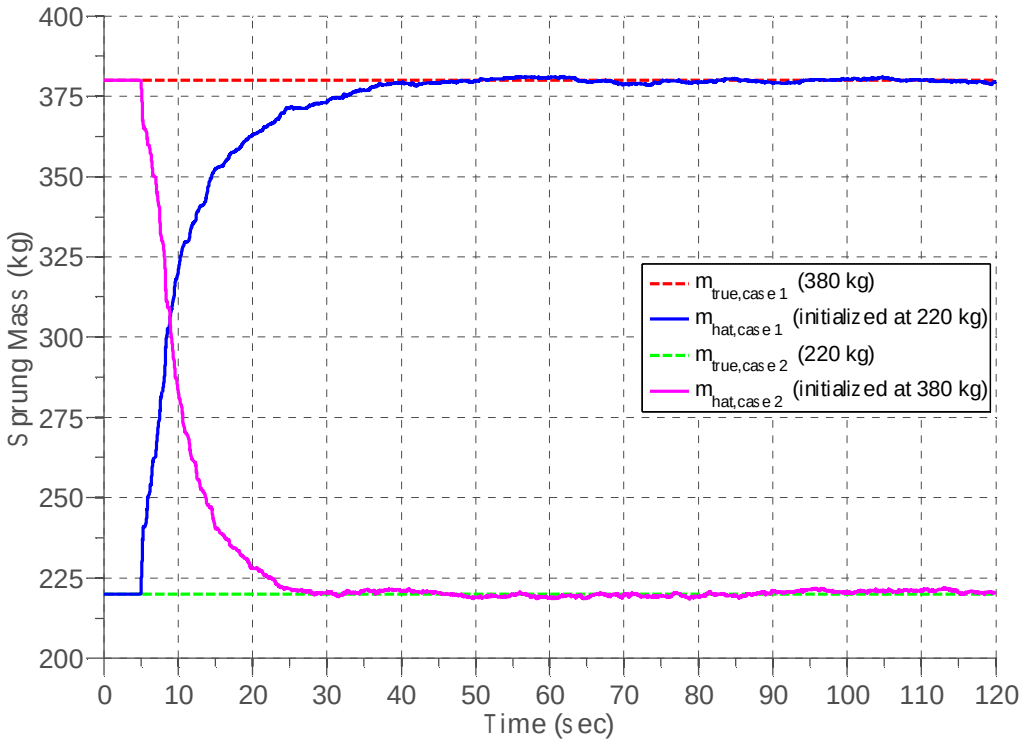


***Figure 6.1 Simulated convergence and stability of sprung mass estimate when initialized far from the true sprung mass value.***

Simulation cases 1 and 2 shown in Figure 6.1 illustrates that the method converges to the true value of the sprung mass parameter and maintains stability once that convergence is

reached. These 2 simulations initialized with an initial sprung mass estimate relatively far from the true value but still rapidly converged.

## *6.2 Simulated Effect of Step-Size on Estimate Convergence*

In this group of tests the true sprung mass and initial estimate of the sprung mass were kept the same while the adaptation step sizes were varied. The goal was to determine which step size was most likely to be effective for later experimental work.

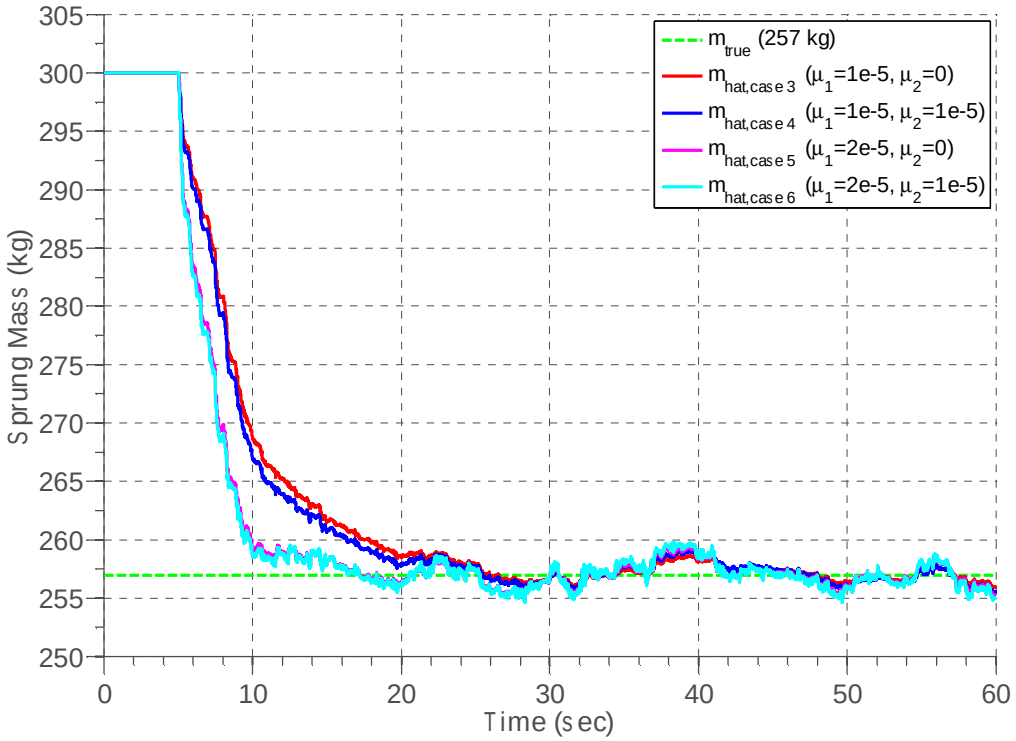


***Figure 6.2 Simulated effect of step-size on convergence when the true sprung mass value is 257 kg and the initial estimate was 300 kg.***

With the initial sprung mass estimate at the nominal sprung mass value, the response of Figure 6.2 shows that cases 3 and 4 converge slightly slower than cases 5 and 6. This is due to the lower adaptation step sizes $\mu_1$ and $\mu_2$. On the other hand, larger adaptation step sizes experience a slight increase in mis-adjustment of the estimated parameter.

Cases 3 and 5 have $\mu_2$ set to 0 while cases 4 and 6 have $\mu_2 = \mu_1$. Having a non-zero $\mu_2$ gives a very slight improvement to the convergence rate. Close inspection of the response shows that the cost of a non-zero $\mu_2$ is increased mis-adjustment of the estimated parameter.

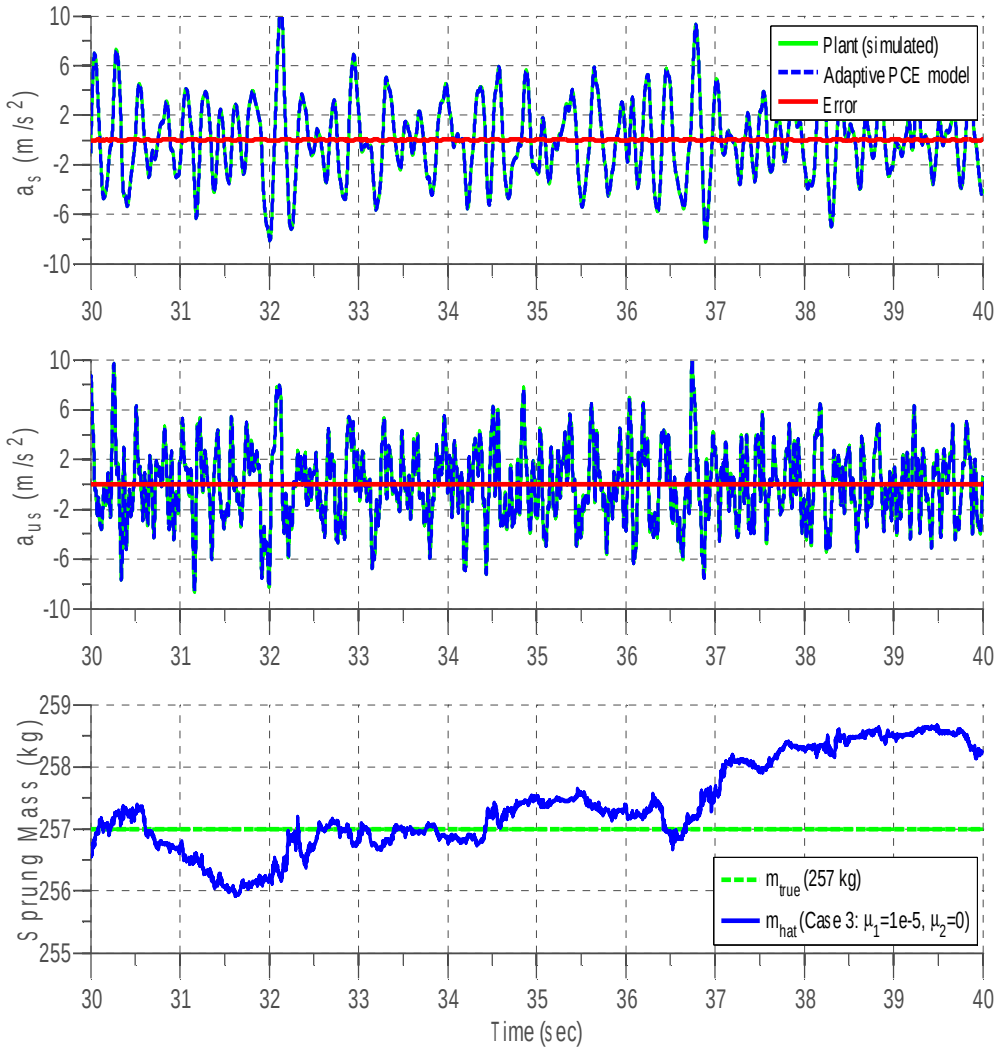


***Figure 6.3 Simulated convergence of sprung mass estimate and a comparison between plant response and the response predicted by the PCE model using the adaptive sprung mass estimate.***

In Figure 6.3 a close up view of 10 seconds of the simulated response of case 3 highlights the range of variation of the estimated sprung mass over time. In this figure the errors between the adaptive PCE model and the simulated linear plant are practically zero, signifying that the PCE model parameters match those of the simulated plant.

## *6.3 Simulated Effect on Convergence of Increased Sprung Mass of the Plant*

In this trio of tests each simulation had the same set of step-sizes and the same initial estimate of the sprung mass. It was the true mass that was varied between runs.

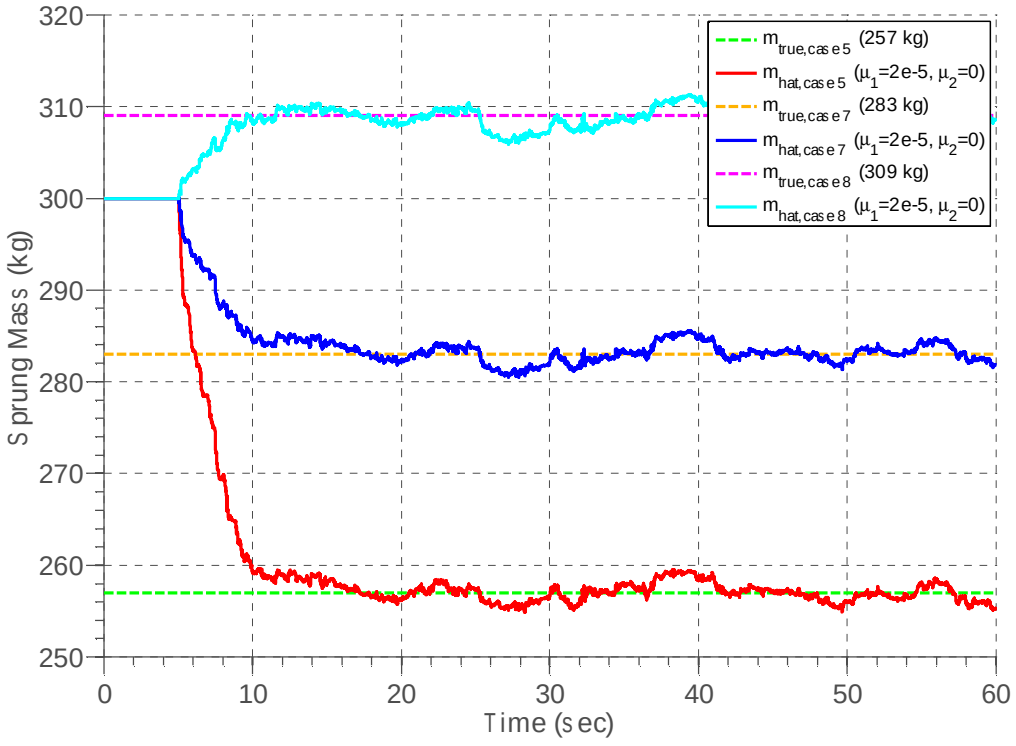


***Figure 6.4 Three cases showing the simulated effect on convergence by increasing the sprung mass of the plant from 257 kg to 283 kg and 309 kg, respectively.***

Cases 5, 7, and 8 show that the true value of the sprung mass can be accurately identified for any mass value within the predetermined variance,.±100 kg. In Figure 6.4 the proximity of the initial estimate to the true value does not appear to significantly affect the rate of convergence. Case 5 which has a large initial estimate error converges at approximately the same rate as case 8 which has the lowest initial estimate error of the three cases shown.

## *6.4 Sensitivity to Poor Estimates of Deterministic Parameters*

To understand the influence of poor measurement of the deterministic system parameters, an informal sensitivity study was performed to identify which system parameters would have the greatest influence on the sprung mass estimate. A total of 13 simulations were completed in this sensitivity study. One simulation kept all parameters as identified by the system ID and was used as a control. Each of the twelve remaining simulations altered the one of the deterministic parameters by either +10% or -10%. This way the sensitivity of the estimation routine to each of the deterministic parameters was tested separately.

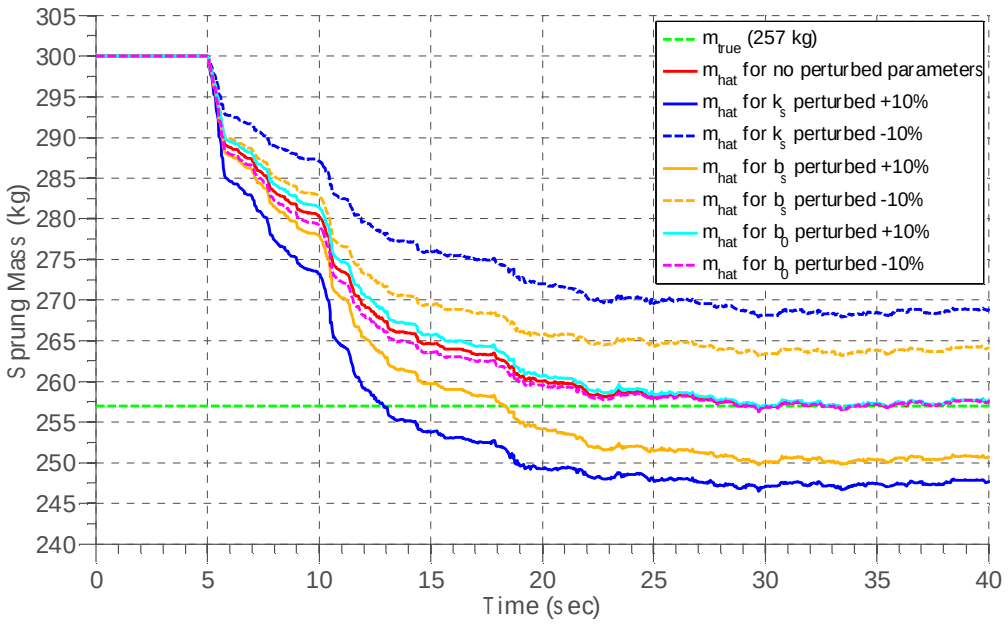


***Figure 6.5 Simulated influence of a 10% error of deterministic suspension parameter estimates on the accuracy of the adaptive PCE estimation algorithm.***

The spring stiffness and damping coefficient of the suspension greatly influence the accuracy of the adaptive PCE algorithm. A ±10% error in the $k_s$ parameter leads to approximately a 3.8% underestimate and overestimate of the sprung mass. Likewise, a ±10% error in the $b_s$ parameter leads to approximately a 1.9% underestimate and 2.7 % overestimate, respectively. The sensitivity of the mass estimate convergence to the spring and damper elements attached to the sprung mass is shown in Figure 6.5.

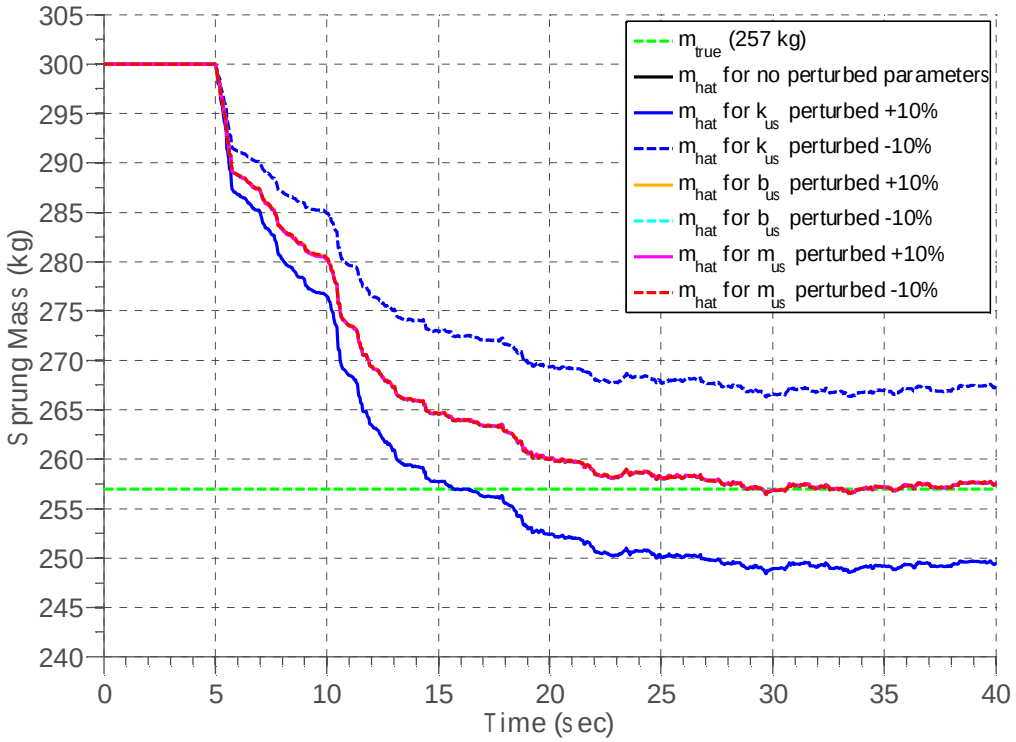


***Figure 6.6 Simulated influence of a 10% error of deterministic tire parameter estimates on the accuracy of the adaptive PCE estimation algorithm.***

Figure 6.6 shows the sensitivity of the convergence to the parameter values for elements attached to the unsprung mass. In the figure it is clear that only the tire stiffness, $k_{us}$ plays a significant role in causing an estimation error.

The results of the sensitivity study show that the adaptive estimation algorithm is sensitive to 3 out of the 6 deterministic linear quarter-car parameters, namely suspension stiffness $k_s$, suspension damping $b_s$ and tire stiffness $k_{us}$. Of these three parameters the suspension stiffness has the greatest impact. All other deterministic parameter, $b_0$, $b_{us}$, $m_{us}$, were found to not significantly impact the convergence of the adaptive estimation routine.

The impact of the stiffnesses on the estimation of the mass might be explained by looking at the fundamental, non-dimensional equations of motion for a dynamic system [25].

$$\ddot{x}(t) + 2\zeta\omega_n \dot{x}(t) + \omega_n^2 x(t) = F(t) \tag{6.1}$$

The response of this system is most heavily affected by the natural frequency, $\omega_n$ which appears twice in the equations of motion and one of those times is squared. Since $\omega_n$ is a function of the stiffness it can be inferred that small changes to the stiffness of the system cause great changes in system response. A change in the damping of the model can also have an effect on the response but would be less dramatic than stiffness.

Considering the implications of (6.1) with regards to the adaptive PCE estimation algorithm, the PCE model which is constructed on the same framework as (6.1) will also be most sensitive to stiffness parameters. Thus, it should be expected that any error present in the quarter-car parameters, as determined by the system ID, will influence the estimate of the sprung mass in the experimental system with the tire and suspension stiffnesses being the most influential.

# 7 Experimental Validation

The simulations of the adaptive PCE parameter estimation method were validated experimentally in a series of tests performed on the quarter-car rig described in section 4.3. In the experimental work all the "known" parameters of stiffness, damping, and unsprung mass were considered constant and the values determined by the system ID of the experimental rig were used by the adaptive PCE algorithm.

The adaptive PCE parameter estimation algorithm was compiled and uploaded to a dSpace Autobox. Just as in the simulation studies and system identification, the experimental rig was excited by a random displacement with a maximum displacement of ± 1 inch. The waveform used by the actuator was generated and filtered by a $2^{nd}$ order Butterworth filter with a 5 Hz cutoff frequency in real-time by a random noise generator in the Simulink code that was uploaded to dSpace. The accelerometer channels were low-pass and high-pass filtered by the Wavebook signal conditioner. The adaptive algorithm gradient step sizes, $\mu_1$ and $\mu_2$ were normalized with respect to the random excitation noise power.

The mean or nominal mass of the distribution, $m_0$ was chosen to be 300 kg with a maximum variation, $m_1$ of 100 kg. The adaptation variables as described in the simulation study were again varied for the experimental work. A table of all test runs and the variables for each test run are shown in Table 7.1.

***Table 7.1 Parameter estimation variables for each test run***

| | Initial Sprung Mass Estimate | "True" Sprung Mass | $\mu_1$ | $\mu_2$ |
|---|---|---|---|---|
| ***Test 1*** | *300 kg* | *257 kg* | *1e-5* | *0* |
| ***Test 2*** | *300 kg* | *257 kg* | *1e-5* | *0* |
| ***Test 3*** | *300 kg* | *257 kg* | *2e-5* | *0* |
| ***Test 4*** | *300 kg* | *257 kg* | *5e-5* | *0* |
| ***Test 5*** | *300 kg* | *257 kg* | *2e-5* | *1e-5* |
| ***Test 6*** | *300 kg* | *283 kg* | *1e-5* | *0* |
| ***Test 7*** | *300 kg* | *283 kg* | *2e-5* | *0* |
| ***Test 8*** | *300 kg* | *283 kg* | *2e-5* | *1e-5* |
| ***Test 9*** | *300 kg* | *309 kg* | *1e-5* | *0* |
| ***Test 10*** | *300 kg* | *309 kg* | *2e-5* | *0* |
| ***Test 11*** | *300 kg* | *309 kg* | *2e-5* | *1e-5* |
| ***Test 12*** | *210 kg* | *309 kg* | *2e-5* | *0* |
| ***Test 13*** | *390 kg* | *309 kg* | *2e-5* | *0* |

The experimental work set out to answer four questions about the algorithm. First, if all parameters are the same, does the algorithm repeatably converge to the same estimate? Second, what is the effect of the step-size on the convergence rate? Third, can the method converge to any "true" value within the range of the probability distribution? Finally, what is the effect of the initial estimate on the convergence? The tests and corresponding figures are shown in Table 7.2

***Table 7.2 Testing categories and corresponding test data and figures.***

| Testing Category | Tests Included in Category | Figures of Test Results |
|---|---|---|
| ***Repeatability of adaptive PCE algorithm*** | *1, 2* | *Figure 7.1* |
| ***Effect of step-size on estimate convergence*** | *1, 3, 4, 5* | *Figure 7.2* |
| ***Different "true" masses*** | *6, 7, 8, 9, 10 , 11* | *Figure 7.3, Figure 7.4* |
| ***Variation of Initial Estimate*** | *10, 12, 13* | *Figure 7.5* |

## *7.1 Experimental Repeatability of Adaptive PCE Algorithm.*

The question of repeatability was pursued in tests 1 and 2. The initial mass estimate, the "true" mass, and the step sizes were all kept the same. In each case the excitation was still ±1 inch maximum displacement, but was still random.

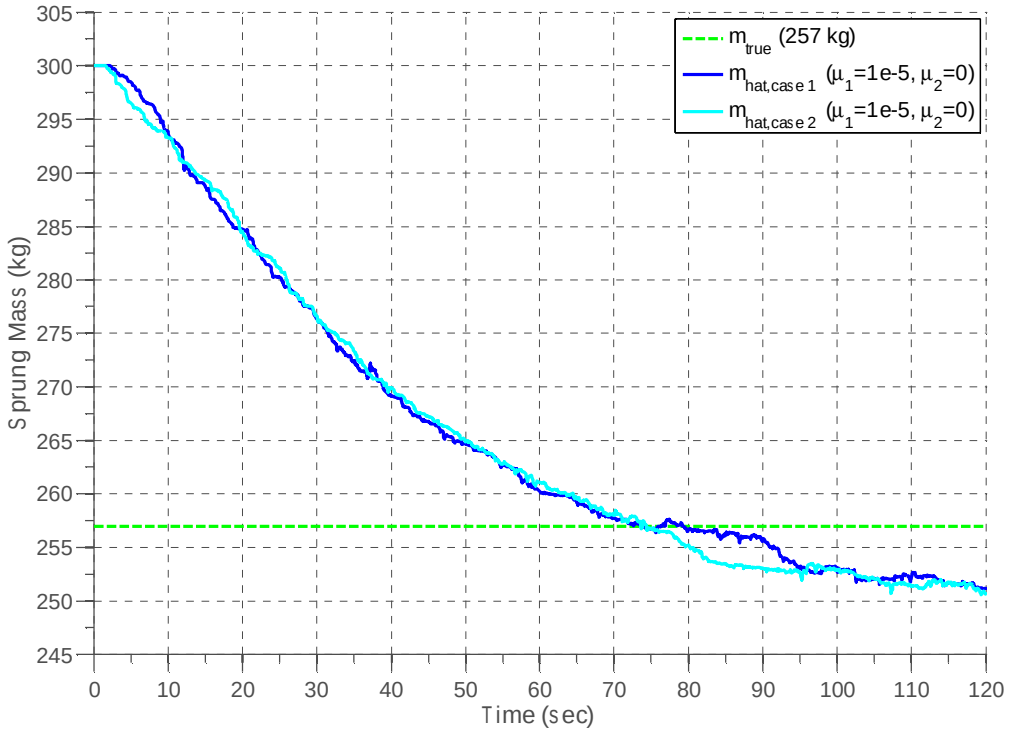


***Figure 7.1 Experimental repeatability of parameter estimate convergence for two test cases with step-sizes, initial estimate, and "true" sprung mass repeated.***

The convergence of the sprung mass estimate as shown Figure 7.1 establishes that in 2 different test runs, with different excitations, and with all variables held constant, the adaptive PCE method is repeatable. The method converges to an estimate within 120 seconds.

The results also show that the estimate is an underestimate. This is due to estimation error of the deterministic parameters by the system ID that was explained in section 4. Shown in the sensitivity study, the estimate of the sprung mass is most influenced by the deterministic parameters of suspension stiffnesses and suspension damping. If the estimate of these parameters is poor the adaptive PCE estimate of the sprung mass will also be poor.

Equivalent adaptive algorithm step-sizes, $\mu$, were used in simulation. The simulation convergence rate appears to predict the convergence rate only slightly faster than the experimental convergence rate.

## *7.2 Experimental Effect of Step-Size on Estimate Convergence*

The effect of step-size on the adaptive PCE algorithm was studied by test cases 1, 3-5. The effect of adaptive step size on the experimental tests is shown in Figure 7.2. The initial estimate and the "true" mass were kept the same for each test. Only the step-sizes were varied.

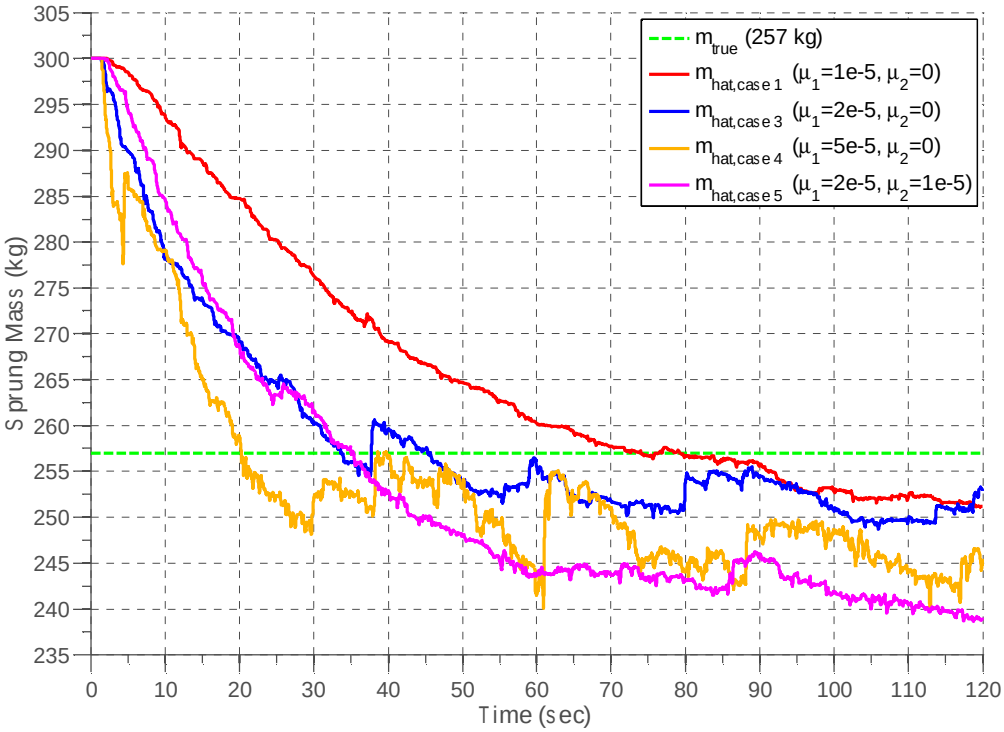


***Figure 7.2 Experimental effect of step-size on the estimate convergence for tests 1, 3, 4, and 5.***

Test case 3 represents the best choice for the adaptive step sizes to deliver quick convergence of the estimate and steady maintenance of the estimate over time. Test case 1 has very little mis-adjustment but takes nearly 120 seconds to converge. Both test case 1 and 3 had $\mu_2$ set to zero which neglects the unsprung mass acceleration. By neglecting the unsprung mass acceleration the algorithm had the least error in its estimate of the sprung mass.

Test cases 4 also neglected the unsprung mass but a larger adaptation step-size caused quicker convergence at the cost of mis-adjustment and a lower estimate of the sprung mass. Test case 5 did not neglect the unsprung mass channel and had a lower estimate for the sprung mass than did test cases 1 and 3. This low estimate is also due to the error

in the deterministic parameter estimates from the sprung mass. It would appear that the greatest error of the system ID was in the tire stiffness and damping parameters.

## *7.3 Experimental Effect on Convergence of Increased Mass of Quarter-car Rig*

The adaptive PCE parameter estimation algorithm was tested to show its capability to detect changes in the "true" mass. Mass was added to the experimental rig and several trials were made to measure the convergence rate for different adaptive step sizes.

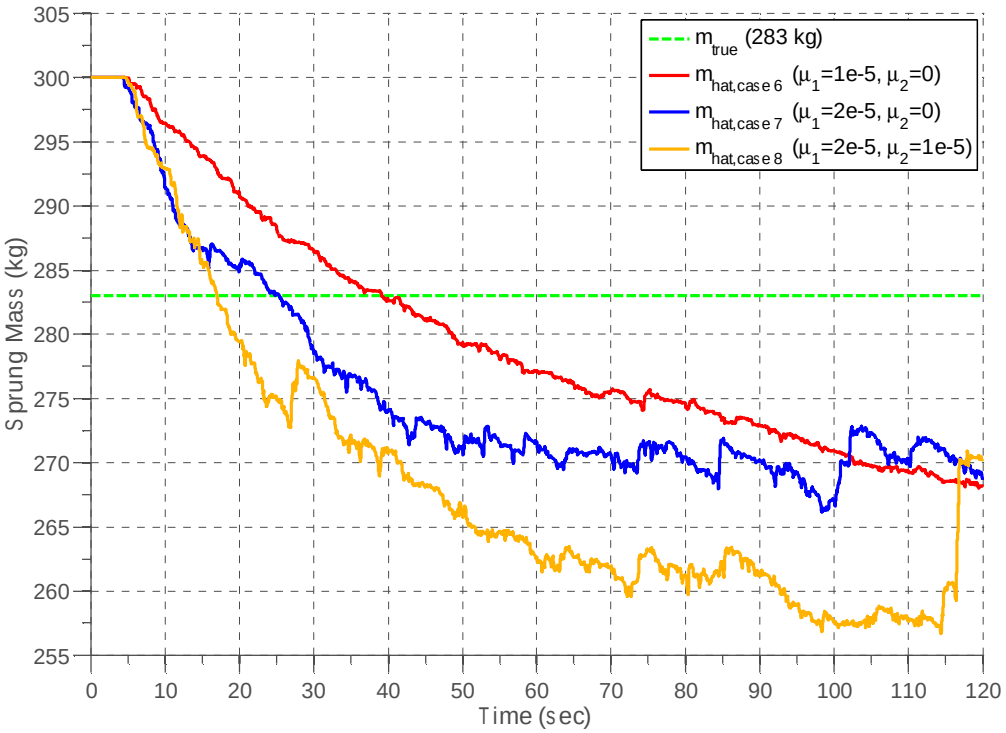


***Figure 7.3 Experimental effect on convergence of the estimate after 26 kg of mass was added to the quarter-car sprung mass for a new "true" mass of 283 kg.***

In Figure 7.3 the convergence rate of the adaptive algorithm is shown for the case of a true mass of 283 kg with an initial estimate of 300 kg. Again, more aggressive adaptation

step sizes gave faster convergence. Neglecting the unsprung acceleration channel also provided a more accurate estimate, though all estimates were under the true value. Test cases 6 and 7 provided a 4.6% underestimate of the true sprung mass. Test case 8 converged to an estimate about 8.1% below the true sprung mass value.

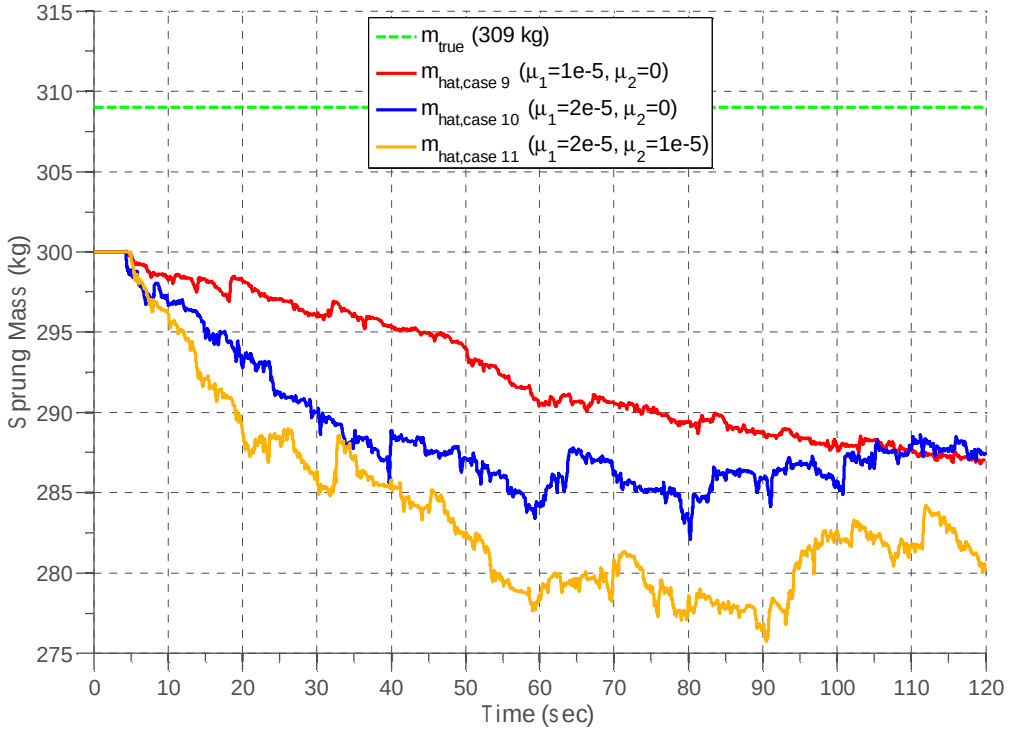


***Figure 7.4 Experimental effect on convergence after 52 kg was added to the quarter-car sprung mass for a new "true" sprung mass of 309 kg.***

In Figure 7.4 we see further time response of the adaptive PCE estimate of the sprung mass. Results are similar to those above with the PCE method under estimating the mass in all cases. Just like the previous results, the unsprung mass channel used in test case 11 causes to the method to underestimate more than 9 and 10 which neglected the unsprung mass channel. Test case 9 and 10 underestimate the sprung mass by approximately 7.11% while test case 11 underestimated by approximately 9.4%.

## 7.4 Experimental Variation of Initial Sprung Mass Estimate

The goal in this set of tests was to determine how the initial sprung mass estimate affected the convergence of the algorithm. In each test the step-sizes were kept the same and "true" sprung mass was maintained at 309 kg. The initial sprung mass estimate was all that was varied from test to test.

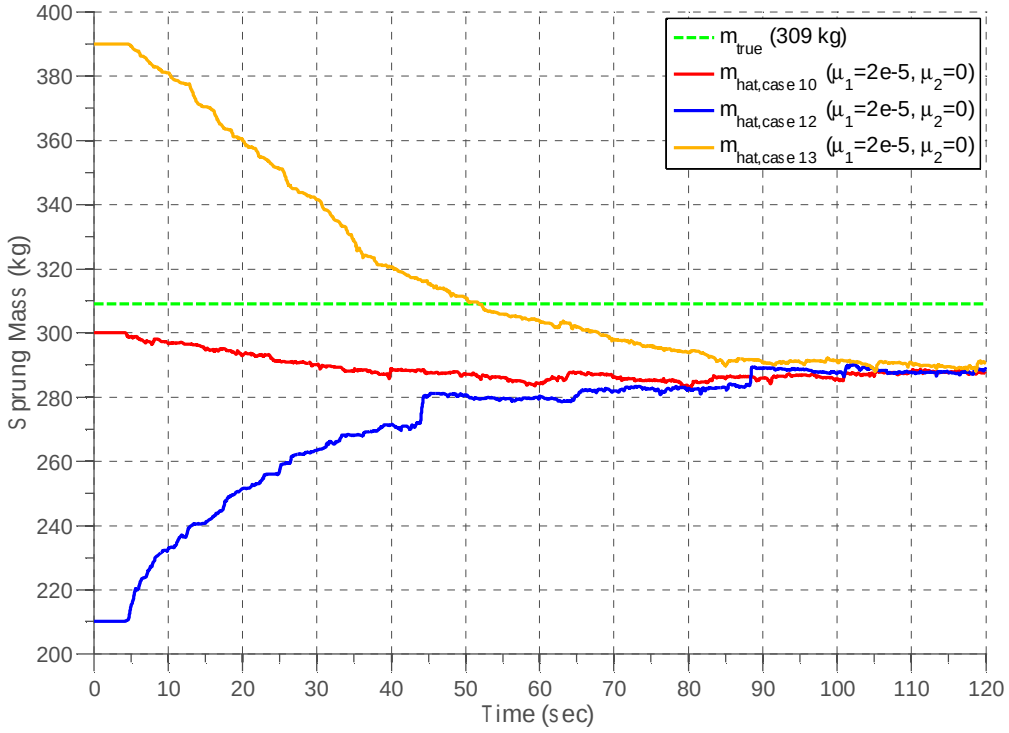


***Figure 7.5 Experimental effect on convergence by different initial estimates of the sprung mass.***

Testing the adaptive PCE estimate with different initial estimates showed consistent convergence to the same estimate in each case. This is shown in Figure 7.5. While the estimate is a low estimate it is nevertheless consistent for each case and only an underestimate by 8.7%.

# 8 Conclusion

A novel real-time adaptive parameter estimation method using a polynomial-chaos expansion was introduced. The literature shows that very little has been done to use polynomial-chaos for parameter estimation and that the PCE method has not previously been implemented in real-time.

General polynomial-chaos expansion equations for linear state-space models of multiple degrees of freedom and one uncertain parameter were presented and discussed. A new technique that separates the uncertain sprung mass parameter from the PCE state-space model facilitated the application of the Galerkin projection for the solution of the PCE state-space coefficients. The PCE state-space model was discretized numerically in Matlab for simulation and real-time implementation.

An adaptive parameter estimation law using the MIT rule was created for this work and was presented and discussed. The adaptation law minimized a positive definite cost function of the sprung and unsprung mass estimate errors by adaptively updating an estimate for the uncertain parameter at each time step. Successful minimization of the cost function lead to an accurate estimate of the uncertain sprung mass parameter. Implementation of the adaptive PCE method in real-time was facilitated by separating the PCE output equations and the cost function into matrices of fixed components and time-varying components.

The successful implementation of the method was shown by a series of simulations in which the method accurately converged to the true value of the unknown parameter. Convergence for the conditions simulated was under 30 seconds. The simulations

showed that the convergence of the algorithm under the simulated conditions was stable. A sensitivity study showed that accurate measurement of the sprung mass spring stiffness, the sprung mass damping coefficient, and the tire stiffness are necessary for accurate convergence of the adaptive PCE algorithm.

Validation of the simulated results was obtained through experimental testing of the method in real-time. The experiments showed that the adaptive PCE algorithm converges to an estimate that is stable and repeatable; however, the convergence trajectories are slightly dependent on the excitation. The initial estimate does not affect the accuracy of the adaptive estimate. The adaptive PCE method also predicts the correct trend for changes in the sprung mass. Bias errors present in the sprung mass estimate have been explained by errors in the model parameters that were assumed deterministic.

## *8.1 Suggested Future Work*

Improvement should be made to the adaptive PCE parameter estimation algorithm before thorough comparison to other estimation routines is undertaken. Future work can explore the use of two or more uncertain parameters in PCE model equations and reformulation of the adaptive estimation strategy to include estimation of the additional uncertain parameters. An additional step might be to model the quarter-car system as a higher degree of freedom system which can be done with relative ease by use of the simplified PCE equations described in (3.26).

In this study the simple MIT rule gradient search method was used, but more sophisticated adaptation laws can be implemented. These would potentially improve both convergence rate and robustness of the estimate. In addition, adaptive step-sizes

could be used in which an aggressive step-size is used for rapid convergence and then the step-size decreased in order to refine the accuracy.

Another potential avenue is the creation of a more realistic quarter-car model by taking a non-linear approach to modeling the quarter-car system. The PCE model for the non-linear system would have to be derived but the PCE method can be applied to non-linear models to great advantage. Investigating non-linear modeling approaches to the problem would also require careful attention to verify the viability of real-time implementation.

# References


[1] E. Blanchard, A. Sandu, and C. Sandu, "Parameter Estimation Method Using an Extended Kalman Filter," in *Joint North America, Asia-Pacific ISTVS Conference* Fairbanks, Alaska, 2007.

[2] E. Blanchard, C. Sandu, and A. Sandu, "A Polynomial-Chaos-Based Bayesian Approach for Estimating Uncertain Parameters of Mechanical Systems," in *2007 International Design Engineering Technical Conference* Las Vegas, Nevada: ASME, 2007.

[3] C. Sandu and L. Li, "On the impact of cargo weight, vehicle parameters, and terrain characteristics on the prediction of traction for off-road vehicles," *Journal of Terramechanics,* vol. 44, pp. 221-238, July 2007.

[4] C. Sandu, A. Sandu, and L. Li, "Stochastic Modeling of Terrain Profiles and Soil Parameters," *SAE Transactions,* vol. 114, pp. 211-220, 2005.

[5] R. Ghanem, S. Masri, M. Pellissetti, and R. Wolfe, "Identification and prediction of stochastic dynamical systems in a polynomial chaos basis," *Computer methods in applied mechanics and engineering,* pp. 1641-1654, 2005.

[6] R. G. Ghanem and P. D. Spanos, *Stochastic Finite Elements: A Spectral Approach,* Revised ed.: Dover Publications, 2003.

[7] D. Lucor and G. E. Karniadakis, "Adaptive Generalized Polynomial Chaos For Nonlinear Random Oscillators," *SIAM Journal of Scientific Computing,* vol. 26, pp. 720-735, March 2004.

[8] S. B. Mulani, "Uncertainty Quantification in Dynamic Problems with Large Uncertainties," in *Aerospace and Ocean Engineering*. vol. Doctor of Philosophy Blacksburg, VA: Virginia Polytechnic Institute and State University, 2006, p. 236.

[9] A. Sandu, C. Sandu, and M. Ahmadian, "Modeling Multibody Dynamic Systems With Uncertainties. Part I: Theoretical and Computational Aspects," *Multibody System Dynamics,* vol. 15, pp. 369-391, 2006.

[10] C. Sandu, A. Sandu, and M. Ahmadian, "Modeling Multibody Dynamic Systems with Uncertainties. Part II: Numerical Applications," *Multibody System Dynamics,* vol. 15, pp. 241-262, April 2006.

[11] J. D. Ziegenmeyer, "Estimation of Disturbance Inputs to a Tire Coupled Quarter-car Suspension Test Rig," in *Mechanical Engineering*. vol. Master of Science Blacksburg, VA: Virginia Polytechnic Institute and State University, 2007.

[12] J. D. Langdon, "Design and Adaptive Control of a Lab-based, Tire-coupled, Quarter-car Suspension Test Rig for the Accurate Re-creation of Vehicle Response," in *Mechanical Engineering*. vol. Master of Science Blacksburg, VA: Virginia Polytechnic Institute and State University, 2007.

[13] E. R. Andersen, C. Sandu, and S. Southward, "Multibody Dynamics Modeling and System Identification of a Quarter-Car Test Rig with McPherson Strut Suspension," in *SAE 2007 Commercial Vehicle Engineering Congress & Exhibition* Rosemont, IL: SAE, 2007.

[14] J. Y. Wong, *Theory of Ground Vehicles,* 3rd ed.: Wiley, 2001.

[15] C. Sandu, A. Sandu, B. J. Chan, and M. Ahmadian, "Treatment of Constrained Multibody Dynamic Systems with Uncertainties," in *SAE Congress 2005*, Detroit, MI, 2005.

[16] N. Wiener, "The Homogenous Chaos," *American Journal of Mathematics,* vol. 60, pp. 897-936, Oct 1938.

[17] D. Xiu and G. E. Karniadakis, "The Wiener-Askey Polynomial Chaos for Stochastic Differential Equations," *SIAM Journal of Scientific Computing,* vol. 24, pp. 619-644, 2002.

[18] A. H. C. Smith, A. Monti, and F. Ponci, "Indirect Measurements via a Polynomial Chaos Observer," *IEEE Transactions on Instrumentation and Measurement,* vol. 56, pp. 743 - 752, June 2007.

[19] S. Southward, "Real-Time Parameter ID Using Polynomial Chaos Expansions," in *2007 ASME International Mechanical Engineering Congress and Exposition (IMECE)* Seattle, Washington, USA: ASME, 2007.

[20] J. P. Norton, *An Introduction to Identification*, 1st ed. Orlando, Florida, USA: Academic Press Inc, 1986.

[21] K. J. Astrom and B. Wittenmark, *Adaptive Control*, 1st ed. Reading, Massachusetts, USA: Addison-Wesley Publishing Company, 1989.

[22] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1985.

[23] D. W. Clark, "Generalized-least-squares Estimation of the Parameters of a Dynamic Model," in *IFAC SYMP. Identification Autom. Control* Prague, Czechoslovakia, 1967.

[24] K. J. Astrom, "Computer control of a Paper Machine - An Application of Linear Stochastic Control Theory," *IBM J. Res. Dev.*, vol. 11, pp. 389-405, 1967.

[25] D. J. Inman, *Engineering Vibrations*, 3rd ed.: Prentice Hall, 2007.

[26] A. D. Poularikas, "The Handbook of Formulas and Tables for Signal Processing," Boca Raton: CRC Press LLC, 1999.

[27] J. Cartinhour, *Digital Signal Processing: An Overview of Basic Principles*, 1st Edition Upper Saddle River, NJ: Prentice Hall, 1999.

## Appendix A – get_Pmatrix.m

```
% This function initializes a matrix of legendre polynomial
coefficients.

% Created by Samuel K. Shimp (23 Jan 2008)

Function [P v v_prime xi_prime_coefs] = get_Pmatrix(S);

% --Set conditions for S <= 1

if S == 0;

P = 1;

end

if S == 1;

P = eye(2);

end

% --For the case of S > 1

if S > 1;

% --Define first two vectors of coefficients

P = zeros(S+1,S+1);

P(1,1) = 1;

P(2,2) = 1;

% --Assemble P matrix

for i = 2:S

%Shift Ln down by one element

Ln_shift = zeros(S+1,1);

Ln_shift(2:S+1,1) = P(1:S,i);

%Find Coefficients
```

```
n = i-1;

P(:,i+1) = ((2*n+1)/(n+1)).*Ln_shift - (n/(n+1)).*P(:,i-1);

end

end

% --Build vectors for legendre function assembly in simulink

if nargout > 1;

v = 0:S; % Vector of exponents for xi in descending order

end;

if nargout > 2;

v_prime = [0,v(1,1:S)]; %Vector of derivative exponents in
descending order

end;

if nargout > 3;

xi_prime_coefs = diag(0:S); %Vector of derivative coefficients

end;

return
```