Recursive Parameter Estimation using Polynomial Chaos Theory Applied to Vehicle Mass Estimation for Rough Terrain

by

Benjamin Lynn Pence

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Mechanical Engineering)
in the University of Michigan
2011

Doctoral Committee:

Professor Jeffrey L. Stein, Co-Chair
Assistant Professor Hosam K. Fathy, Pennsylvania State University, Co-Chair
Professor Dennis S. Bernstein
Assistant Professor Clayton D. Scott

Acknowledgments

I'd like to acknowledge and express gratitude to the people that have inspired and supported this dissertation work. My dissertation advisors, Prof. Jeffrey Stein and Prof. Hosam Fathy, encouraged, inspired, mentored, and supported me. They helped me stretch and grow beyond what I was capable of on my own. A significant portion of this work is a direct result of their feedback and ideas. I also thank my other dissertation chairmen, Prof. Dennis Bernstein and Prof. Clayton Scott, for their feedback on this work and their classroom instruction that provided a theoretical foundation for much of this dissertation.

I'd like to thank the people in my research group at the University of Michigan for their support, friendship, and encouragement. They provided valuable feedback and ideas in group meetings and in private discussions that contributed to this dissertation.

I thank my collaborators and friends at Virginia Tech for their contributions to this work. Samuel Shimp initiated my interest in polynomial chaos theory and provided valuable help to get me started. Fellow collaborators, Prof. Corina Sandu and Joseph Hays, contributed to both the polynomial chaos based estimators and the vehicle dynamics models.

Most importantly, I wish to thank my wife, children, and family for supporting me in my goals and dreams and making my life rich and enjoyable. I thank my Father in Heaven for His inspiration, love, and quiet but constant impression of assurance that sustained me through my graduate education.

Finally, the U.S. Army TARDEC funded this research through its center for excellence in automotive modeling and simulation. I am sincerely grateful for this support which made this work and my doctoral education possible.

Table of Contents

List of Figures

List of Tables

List of Appendices

Abstract

This dissertation uses polynomial chaos theory to address recursive parameter estimation in state space systems. It joins the recursive estimators with base excitation modeling concepts to determine the mass of off road vehicles, and successfully demonstrates the methods on actual vehicle data.

The recursive, polynomial chaos based estimators of this dissertation can be applied to linear and nonlinear state space systems having linear time invariant output equations. Unlike regressor model based estimators, this dissertation's estimators can be applied directly to state space systems, and in some situations, the proposed methods can be more easily tuned than state filtering methods. The new estimation techniques contribute to the solution of the vehicle mass estimation problem.

An accurate onboard estimate of vehicle mass is valuable to the optimal performance of safety systems, chassis controllers, and drivetrain controllers. These systems schedule gear shifts, actuate brakes, induce steer, schedule fuel injection, warn drivers of rollover susceptibility, etc. Since vehicle mass can vary significantly from one loading condition to the next, the estimate of vehicle mass must be updated online.

A significant number of mass estimation algorithms have been developed for *on road* conditions; however, the *rough terrain* real-time vehicle mass estimation problem remains relatively unexplored. Existing rough terrain solutions are difficult to apply in practice because they assume that the terrain profile is known, estimated, or measured, or they assume that the vehicle is equipped with an active or semi-active suspension.

Instead, this dissertation adopts a *base excitation* approach. This approach treats the vertical accelerations of the four unsprung masses as measured inputs to the dynamic equations governing the motion of the sprung mass; the estimator uses these sprung dynamics to calculate the most likely value of the vehicle mass.

This dissertation applies the polynomial chaos estimators and base excitation concepts to experimental data from an actual vehicle. When joined with a detection algorithm, the proposed approach had a success rate of 94%: 31 predicted successes with only 2 false positives. Without the detection algorithm, the proposed approach had a success rate of 78%: 31 total successes out of 40 total experiments.

Chapter 1

Introduction

A main objective of this dissertation is to address the problem of recursively estimating the sprung mass of a vehicle traversing rough terrain. An accurate onboard estimate of vehicle mass is valuable for optimizing the performance of safety systems, chassis controllers, and drivetrain controllers. These systems schedule gear shifts, actuate brakes, induce steer, schedule fuel injection, warn drivers of rollover susceptibility, etc. Since vehicle mass can vary significantly from one loading condition to the next, the estimate of vehicle mass needs to be updated online.

As an important part of the solution to this mass estimation problem, this dissertation introduces new recursive estimation techniques based on polynomial chaos theory. These estimation techniques can be applied to a variety of linear or nonlinear state-space systems, and hence their scope extends beyond the mass estimation problem. A review of the scientific literature covering parameter estimation for state space systems using polynomial chaos theory is provided in Chapter 2. Chapter 4 presents a recursive solution based on polynomial chaos theory for estimating the maximum likelihood values of unknown parameters of state-space systems. This maximum likelihood estimator is relatively easy to tune compared to other benchmark estimation approaches, and it is sufficiently numerically efficient for the real-time mass estimation problem of this dissertation. Chapter 5 presents Bayesian approaches to the recursive estimation problem. The recursive Bayesian maximum *a posteriori* estimator is similar in computational demand and ease of tuning to the maximum likelihood approach of Chapter 4. The Bayesian approach can also be used to estimate moments of the statistical distributions of the unknown parameters (in addition to the most likely values), but to do so requires more computational resources because of the requirement to evaluate an integral at each time iteration.

The model of the vehicle dynamics is, of course, also an essential component to the solution of the vehicle mass estimation problem. Chapter 3 reviews the different modeling approaches for vehicle mass estimation. A significant number of mass estimation algorithms have been developed for *on road* conditions, and for the most part, these techniques use models of the longitudinal vehicle dynamics for mass estimation. The *rough terrain* real-time vehicle mass estimation problem, however, remains relatively unexplored, and existing solutions to this problem remain difficult to apply in practice. One of the main challenges is the fact that the motions introduced by rough terrain are significant enough to make estimation based on longitudinal vehicle dynamics infeasible; this rough terrain, however, makes mass estimation based on vertical vehicle dynamics more viable due to significant terrain-induced excitations.

The scientific literature has explored the use of vertical vehicle dynamics for mass estimation. However, existing approaches assume that the terrain profile is known, estimated, or measured, or they assume that the suspension is equipped with an active or semi-active force actuator which provides a known suspension force. Instead, Chapter 6 of this dissertation derives and experimentally validates base excitation models of vehicle ride dynamics. These base excitation models treat the four vertical unsprung mass accelerations, instead of the terrain profile, as the input to the state-space equations that govern the dynamic motion of the sprung body of the vehicle. The measured vertical acceleration of the sprung mass is the output of these equations.

In Chapter 7, the estimation approaches of Chapters 4 and 5 are combined with the base excitation models of Chapter 6 to formulate the complete solution to the sprung mass estimation problem. The techniques of this dissertation are validated in Chapter 8 using experimental data of an actual vehicle traversing a variety of terrains. The experimental results demonstrate the viability of the proposed solution and explore the limitations of the modeling assumptions. In the experimental study, the polynomial chaos based estimators of this dissertation are compared with traditional estimators including least squares algorithms and filtering algorithms. Both the approaches of this dissertation and the filtering methods perform more accurately than the least squares algorithms. In the experimental results, the polynomial chaos based algorithms have a higher success rate than the benchmark estimators. The polynomial chaos based algorithms are easier to tune that the state-filtering methods, but they are also more computationally expensive than the extended Kalman filter.

Chapter 2

Literature Review of Estimation Algorithms Based on Polynomial Chaos Theory

This chapter reviews the scientific literature that uses polynomial chaos theory as a framework for estimating unknown parameters of dynamic state space systems. Polynomial chaos theory provides a tool for evaluating dynamic systems with static/fixed stochastic uncertainties such as unknown parameters or initial conditions. This theory can be joined with estimation algorithms to calculate the most likely values of these stochastic uncertainties given only the system's known input and output signals. The scientific literature has generated a variety of polynomial chaos based estimators. This literature review groups the various estimators into two categories: (a) non-recursive or batch estimators and (b) recursive estimators – the methods of this dissertation fit into this second category. Before reviewing the various polynomial chaos based estimators, the following section provides a brief background on polynomial chaos theory. Then Section 2.2 reviews the non-recursive estimators and Section 2.3 reviews the recursive estimators. Finally, Section 2.4 provides conclusions.

2.1 Introduction to Generalized Polynomial Chaos Theory

The Generalized Polynomial Chaos (gPC) framework was developed by Xiu and Karniadakis (Xiu and Karniadakis, 2002) building off groundbreaking work by Ghanem and Spanos (Ghanem and Spanos, 1991) and the conceptualization by Wiener (Weiner, 1938). When gPC theory is applied to state space equations, the state and output equations are expressed as expansions of orthogonal polynomials of random variables. Galerkin projection (Xiu and Karniadakis, 2002) or collocation (Xiu, 2007) techniques solve for the coefficients of these polynomial expansions. Given the expansion coefficients, the output equation describes an *output process* or *family of output trajectories*. Given any realization of the random variables, the output process collapses to a single output trajectory.

The goal of estimation theory is to calculate the realization of the random variables that causes the output process to collapse to the output trajectory that is most like (in some sense) the trajectory of the measured system output. The following section reviews algorithms that perform this estimation in a batch manner, *e.g.*, after all the data have been collected.

2.2 Batch Polynomial Chaos Estimators

Batch estimators calculate estimates of unknown parameters by evaluating an entire set or batch of data as opposed to updating the estimates iteratively as data arrive. Blanchard *et al*. proposed a batch Bayesian parameter estimator for state space systems that selects estimates based on the maximum *a posteriori* estimate (Blanchard et al., 2008). Marzouk and Xiu (Marzouk and Xiu, 2009) proposed a Bayesian approach to estimate parameters of systems governed by partial differential equations; they provided a valuable study on the convergence of the polynomial chaos based estimators. Their work used the stochastic collocation approach and extended earlier but similar work (Marzouk et al., 2007) which used the Galerkin method.

Another batch estimator proposed by (Blanchard et al., 2010b) called the “whole-set-of-data-at-once” approach, combines polynomial chaos theory with the extended Kalman filter. Blanchard *et al*. report that the “whole-set-of-data-at-once” approach yields better results than the recursive or so-called “one-time-step-at-a-time” approach also proposed by (Blanchard et al., 2010b).

2.3 Recursive Estimators

The polynomial chaos based algorithms of this section estimate system parameters iteratively as data arrive. The polynomial chaos estimators of this section can be categorized into three groups: (a) observer/Kalman filter based estimators, (b) estimators based on evaluating gradients of instantaneous cost functions, and (c) recursive Bayesian and maximum likelihood estimators. The estimators proposed in this dissertation fit in the third category.

2.3.1 Observer/Kalman Filter Based gPC Estimators

Polynomial chaos theory can be combined with observer theory to predict estimates of the system states and then update the state predictions when measurements of the output signal become available. System parameters can also be estimated by these observers if the unknown parameters are explicitly treated as dynamic system states.

Blanchard *et al*. combined polynomial chaos theory with the extended Kalman filter for state and parameter estimation (Blanchard et al., 2010b). Li and Xiu proposed a gPC ensemble Kalman filter for improved estimation accuracy and computational efficiency (Li and Xiu, 2009). Saad *et al*. proposed a gPC-based ensemble Kalman filter for system identification and monitoring (Saad et al., 2007), and Smith *et al*. (Smith et al., 2006) combined gPC with the Luenberger observer for state estimation.

2.3.2 gPC Estimation via Gradients of Instantaneous Cost Functions

Southward developed a unique framework for recursive parameter estimators based on gPC theory (Southward, 2008). Southward's method calculates parameter estimates by searching in the direction of gradients of instantaneous quadratic cost functions. Shimp (Shimp, 2008) and the author (Pence et al., 2010) applied Southward's method to the problem of real-time vehicle mass estimation. Because of (Shimp, 2008), the algorithm by (Southward, 2008) inspired and initiated the author's interest in polynomial chaos based estimators.

2.3.3 gPC Bayesian and Maximum Likelihood Estimators

The algorithms of this section - and the proposed algorithms of this dissertation - recursively seek parameter estimates that are optimal in the Bayesian or maximum likelihood sense. Therefore, these estimators are similar to the estimators of Section 2.3.1; they are different, however, because they do not use state observers such as Kalman filters. Instead, polynomial chaos theory propagates parametric uncertainty through the dynamic system, and the Bayesian or maximum likelihood estimation theories are applied directly to the stochastic system output to calculate parameter estimates. This is the method taken by the proposed estimators of this dissertation. Dutta and Bhattacharya also proposed a Bayesian estimator based on polynomial chaos theory (Dutta and Bhattacharya, 2010). Unlike the methods of this dissertation, their estimator requires all of the inner products in the estimation algorithm to be recomputed at each time step.

2.4 Conclusions

This chapter has reviewed estimators for state space systems that are founded on polynomial chaos theory. This chapter reviewed non-recursive or batch estimators as well as recursive estimators. The proposed estimators of this dissertation recursively seek parameter estimates that are optimal in the Bayesian or maximum likelihood sense. However, they do not rely on observers nor require computing inner products at each time step.

Although they can be applied to a broad range of estimation problems, the estimators of this dissertation were motivated by the vehicle mass estimation problem of this dissertation. In this problem, the computation is performed onboard the vehicle where computational memory and resources are typically limited. Recursive algorithms with low computation requirements (like the methods of this dissertation) are attractive for this problem.

Chapter 3

Vehicle Mass Estimation Literature Review

This chapter reviews the literature covering online vehicle mass estimation. To date, the scientific literature has explored more than 50 different ways to estimate in real-time the mass of a vehicle. The large number of different approaches hints to the importance and difficulty of the mass estimation problem. While all of the approaches have differences, this dissertation categorizes the various methods into groups according to the approach's underlying physical model. These underlying models are based on (a) the longitudinal vehicle dynamics, (b) vertical ride (or suspension) physics, and (c) combined axes dynamics and (d) the drivetrain shuffle dynamics. First, Section 3.1 reviews vehicle mass estimation techniques based on the longitudinal vehicle dynamics. Section 3.2 reviews approaches that use combined axes dynamics, and Section 3.3 reviews mass estimation via drivetrain shuffle dynamics. Finally, Section 3.4 reviews the methods based on the vertical vehicle physics – the approach of this dissertation fits into this group.

3.1 Longitudinal Dynamics Mass Estimation

The longitudinal axis is in line with the vehicle's forward and reverse directions of travel. The vehicle's acceleration and deceleration in the longitudinal direction is proportionally related to the vehicle's net longitudinal forces; the proportionality constant, by Newton's second law, is the mass of the vehicle. The various estimation approaches that use the longitudinal dynamics differ based on how they estimate or measure the three parts – force, acceleration, and mass – of the longitudinal equation.

3.1.1 Longitudinal Acceleration/Deceleration

Longitudinal vehicle acceleration can potentially be measured directly using an accelerometer, and some techniques suggest this approach (Massel et al., 2004), (Lingman and Schmidtbauer, 2003). Typically, however, longitudinal acceleration is inferred from vehicle velocity using, for example, measurements of wheel speed (*e.g.* (Vahidi et al., 2003b), (Eriksson, 2009), (Ritzen, 1998)), angular velocity of the engine drive shaft (*e.g.* (Eriksson)), and/or a Global Positioning System (GPS) (Bae et al., 2001). The motivation to avoid measuring acceleration directly may be to reduce cost by using equipment and signals already available onboard the vehicle or to reduce disturbances caused be the pitching motion of the vehicle. Filtering, smoothing, and/or other techniques have been employed to reduce noise in acceleration estimates (Vahidi et al., 2005), (Bae et al., 2001), (Eriksson), and (Ritzen, 1998). Also some techniques fuse information from two or more sensors to potentially improve the estimates of the longitudinal acceleration (Lingman and Schmidtbauer, 2003), (Bae et al., 2001), and (Eriksson).

3.1.2 Longitudinal Forces

The net longitudinal force acting on a vehicle at any time is a sum of a number of components: engine induced forces, braking forces, gravitational forces, rolling resistance, aerodynamic drag, longitudinal force components due to uneven terrain, dynamic inertial forces from rotating components, parasitic losses such as air conditioning and alternator forces, torque converter losses, and other mechanical resistive forces (see Chapter 4 of (Rajamani, 2006)). Estimating the actual longitudinal forces acting on the vehicle may be the most difficult part of the longitudinal mass estimation problem (Eriksson). The various mass estimation techniques differ by how they measure, estimate, or creatively neglect the different force components.

An estimate of engine torque may be calculated from an engine map ((Vahidi et al., 2003b), (Bae et al., 2001), (Eriksson, 2009), (Winstead and Kolmanovsky, 2005), (Ritzen, 1998)) based on engine speed, throttle position, engine temperature and/or other available sensors (see for example (Eriksson, 2009), (Ritzen, 1998)). In general, the longitudinal mass estimation research assumes that the engine torque estimate is known and reliable (notable exceptions include (Druzhinina et al., 2002) and (Winstead and Kolmanovsky, 2005) where model predictive control is used to enhance parameter estimation under uncertain engine torque conditions). Resistive, inertial, and parasitic losses are subtracted from the estimate of engine torque and converted to wheel torque based on the current transmission gear ratio (and possibly a gain from a torque converter) (see Sections 4.2 and 5.5.1 of (Rajamani, 2006) and (Eriksson, 2009)). The resultant longitudinal force at the wheel/terrain surface is a function of the estimated wheel torque, the effective wheel radius (which can vary), wheel inertia, and other tire properties (see Chapter 4 of (Rajamani, 2006)).

Environmental disturbance forces include rolling resistance, gravitational forces, and aerodynamic drag. In most proposed methods, the rolling resistance and aerodynamic drag (assuming negligible headwind) are calculated based on the vehicle velocity and subtracted from the estimated wheel force. In (Bae et al., 2001) the road grade is estimated based on GPS measurements. In (Lingman and Schmidtbauer, 2003) the road grade is measured using a specific force sensor. In (Klatt, 1985), (Reiner et al., 1990), (Genise, 1996), (Phillips and Richardson, 1997), (Leimbach et al., 2001), (Leimbach et al., 2002), and (Eriksson, 2009) the environmental forces were estimated during driveline disengagement or assumed to be constant between measurements so that taking the difference between the force equation at the two time instances removed the effect of environmental forces. In (Vahidi et al., 2003a), (Vahidi et al., 2005), (Bae et al., 2001), (Massel et al., 2004), (Druzhinina et al., 2002), (Winstead and Kolmanovsky, 2005), (Lingman and Schmidtbauer, 2003) and (Ritzen, 1998), unknown environmental forces are estimated simultaneously as an explicit part of the mass estimation routine. In (Fathy et al., 2008), (Hayakawa et al., 2002), and (Yamada et al., 2006) filtering techniques extract high frequency force/acceleration information; since environmental forces generally occur at lower frequencies, this filtering approach allows the mass estimation algorithm to neglect these disturbance forces. In (Massel et al., 2004), (Lee et al., 2009), (Breen, 1996), (Genise, 1996), and (Bellinger et al., 2003), the mass was estimated only when specific conditions were met (*e.g.* transmission in a specified gear, longitudinal force and its derivative were above a threshold, acceleration above a threshold, steady state braking (Breen)); in some cases, environmental forces were neglected in these estimation techniques. In (Yanase), the forces for driving at a constant speed on level ground were found prior to runtime and used to correct the force estimates during runtime. The method by (Grieser) also accounted for longitudinal forces due to steering angle.

3.1.3 Longitudinal Mass Estimation Techniques

Even with measurements or estimates of the longitudinal forces and acceleration, there are many different approaches for estimating the vehicle mass. Least squares and filtering approaches iteratively update their prediction of the vehicle mass as they repeatedly evaluate the longitudinal dynamic equation. Conversely, other algorithms seek high acceleration/deceleration events in which inertia dominates the equation, and update the mass estimate only during these events.

The approaches taken by (Winstead and Kolmanovsky, 2005), (Vahidi et al., 2003a), (Eriksson, 2009), and (Lingman and Schmidtbauer, 2003) use Kalman filtering techniques to estimate the mass. The methods proposed by (Fathy et al., 2008), (Bae et al., 2001), (Vahidi et al., 2003a), (Vahidi et al., 2005), (Ritzen, 1998), (Zhu et al., 2000), (Zhu et al., 2002), (Rieker et al., 2002), (Yanase, 2005), and (Germann and Isermann, 1994) use least squares or other regression approaches. As part of a model reference adaptive control strategy, (Druzhinina et al., 2000) and (Druzhinina et al., 2002) estimate mass, road grade, and rolling resistance using a speed gradient algorithm. The various approaches of (Massel et al., 2004), (Lee et al., 2009), (Klatt, 1985), (Reiner et al., 1990), (Phillips and Richardson, 1997), (Leimbach et al., 2002), and (Bellinger et al., 2003) seek events in which the acceleration signal is large enough that the equation $mass = Force/Acceleration$ is sufficiently well conditioned. Then the mass estimate can be simply calculated by evaluating that equation. Sometimes when this last approach is used, filtering, averaging, or another scheme is also used to reduce noise. In (Bellinger and Shutty, 2000), the estimates of mass and road grade were taken from a lookup table with at least vehicle speed and fueling command as inputs.

3.2 Combined Axes Methods

The mass of a vehicle can be estimated based on evaluating Newton's second law for two or more axes of the vehicle. The estimation scheme may combine multiple axes into one coupled set of equations as in (Wenzel et al., 2006) and (Huang and Lin, 2009) and then estimate the mass based on the entire set of equations. Alternatively, it may calculate multiple estimates of the vehicle mass by evaluating the dynamics of each axis individually and then determine the best estimate using supervisory logic as in (Huh et al., 2007) and (Han et al., 2009).

The method of (Huang and Lin, 2009) uses a combination of the lateral, yaw, and roll axes of the vehicle. The estimator of (Wenzel et al., 2006) combines the longitudinal, lateral, roll, and yaw axes. And the technique of (Huh et al., 2007) and (Han et al., 2009) evaluate the lateral, longitudinal, and vertical ((Huh et al., 2007) only) axes individually and combine the estimates using the supervisory logic.

3.3 Drivetrain Shuffle Dynamics Method

Fremd (Fremd) exploited the relationship between the vehicle's drivetrain shuffle frequencies and its mass for a creative approach to vehicle mass estimation.

3.4 Vertical Axis Methods

The majority, if not all, of the algorithms in Sections 3.1 through 3.3 have been developed for *on road* conditions, and for the most part, these techniques use models of the longitudinal vehicle dynamics for mass estimation. The *rough terrain* real-time vehicle mass estimation problem, however, remains relatively unexplored, and existing solutions to this problem remain difficult to apply in practice. As stated in the introduction, one of the main challenges is the fact that the motions introduced by rough terrain are significant enough to make estimation based on longitudinal vehicle dynamics infeasible; this rough terrain, however, makes mass estimation based on vertical vehicle dynamics much more viable due to significant terrain-induced excitations. This section reviews the different vehicle mass estimation techniques that use a vertical vehicle model.

The scientific literature has explored the use of vertical vehicle dynamics for mass estimation. Many existing approaches assume that the terrain profile is known, estimated, or measured, (Blanchard et al., 2010a), (Best and Gordon, 1998), (Huh et al., 2007), (Kim and Ro, 2000), (Lin and Kortum, 1991), and (Shimp, 2008). Others assume that the suspension is equipped with an active or semi-active force actuator which provides a known suspension force ((Rajamani and Hedrick, 1995), (Ohsaku and Nakai, 2000), (Song et al., 2005), (Du et al., 2008), (Du and Zhang, 2010), and (Priyandoko et al., 2009)). A third group uses modal analysis or frequency domain techniques ((Rozyn and Zhang, 2010) and (Tal and Elad, 1999)). Finally, (Hac, 2009) investigated using dynamic tire pressure measurements to estimate mass. Instead, Chapter 6 of this dissertation derives base excitation models of vehicle ride dynamics. These base excitation models treat the four vertical unsprung mass accelerations, instead of the terrain profile, as the input to the state-space equations that govern the dynamic motion of the sprung body of the vehicle. The measured vertical acceleration of the sprung mass is the measured output of these equations. This base excitation approach furnishes estimation schemes that do not require active suspensions, wheel pressure sensors, or *a priori* knowledge, measurement, or estimation of the terrain.

Chapter 4

Recursive Maximum Likelihood Parameter Estimation for State Space Systems using Polynomial Chaos Theory

This chapter describes a novel method for recursively estimating the unknown static parameters of linear and nonlinear state space systems. This method combines two established theories: generalized polynomial chaos (gPC) theory (Xiu and Karniadakis, 2002) and maximum likelihood estimation theory (see Chapter 12 of (Moon and Stirling, 2000)).

Unlike many traditional methods such as recursive least squares (Chapter 4 of (Ioannou and Sun, 1996)) and total least squares (Section 7.7 of (Moon and Stirling, 2000)), the proposed method does not require the underlying model to be formatted into a regressor model form but can be applied directly to state space models. Other state-space estimation methods, such as Kalman filtering approaches (Simon, 2006) and sequential Monte Carlo (or particle filtering (Simon, 2006), (Ristic et al., 2004), (Arulampalam et al., 2002)) approaches treat unknown parameters as dynamic states and formally include them in the state vector, thus differing from the proposed approach. This chapter will use a numerical simulation to study the benefits of the proposed approach compared with the filtering methods.

Many researchers in the estimation community have recognized the benefit of using polynomial chaos theory for parameter estimation of dynamic systems. Chapter 2 of this dissertation reviews the scientific literature on polynomial chaos based estimators.

This chapter combines polynomial chaos theory with maximum likelihood estimation to recursively estimate the static unknown parameters of state space systems. Similar to the approaches discussed in Chapter 2, this chapter applies polynomial chaos theory to solve the stochastic differential equations that govern the underlying system dynamics. However, unlike any of the methods of the existing literature, this chapter *recursively* calculates the *maximum likelihood values* of the unknown parameters *based on all of the past system observations*. To the best of the author's knowledge, this is the first proposed approach to address recursive maximum likelihood parameter estimation for state space systems using polynomial chaos theory.

4.1 Generalized Polynomial Chaos Theory

The generalized polynomial chaos (gPC) framework is essential to the methods of this chapter. The gPC framework was developed by Xiu and Karniadakis (Xiu and Karniadakis, 2002) building off groundbreaking work by Ghanem and Spanos (Ghanem and Spanos, 1991) and the conceptualization by Wiener (Weiner, 1938).

A set of continuous-time state equations, which are often nonlinear, are used to describe the dynamic behavior of a system.

$$\dot{x} = f(t, x, u; \theta) \tag{4.1}$$

$$x(0) = x_0 \tag{4.2}$$

The vector $x \in \mathbb{R}^{n_s}$ contains the system states which have known initial conditions x_0, and the vector $\theta = \left[\theta_1\ \theta_2 \cdots \ \theta_{n_p}\right]^T$ contains the unknown parameters. If any of the initial conditions x_0 is unknown, it can be treated as one of the unknown parameters. Or, if the state equation is linear, asymptotically stable, and time invariant, it may be possible to neglect the initial conditions; this is explored in Appendix A1. The input vector $u \in \mathbb{R}^{n_u}$ is known and time-varying. The "dot" notation signifies the derivative with respect to time t.

In general, observations on a system may be governed by a nonlinear, time-varying output model $y = h(t, x, u; \theta)$. However, the scope of this chapter is limited to systems having observations described by a linear, time-invariant, discrete-time output model:

$$y_k = C(\theta)x(t_k) + v_k. \tag{4.3}$$

The output vector $y_k \in \mathbb{R}^{n_y}$ contains the observations on the system at time t_k. The vector $v_k \in \mathbb{R}^{n_y}$ represents an additive Gaussian disturbance with known covariance $R_k \in \mathbb{R}^{n_y \times n_y}$.

The unknown parameters are viewed as being functions of random variables ξ_i, *i.e.*, $\theta_i = \theta_i(\xi_i)$ for $i = 1, \dots, n_p$. The random variables are independently identically distributed (IID), and the joint density function is $\bar{\rho}(\xi) = \prod_{i=1}^{n_p} \rho(\xi_i)$ where $\rho(\xi_i)$ is the distribution of the i^{th} random variable ξ_i, and $\xi = [\xi_1\ \xi_2\ \cdots\ \xi_{n_p}]$. Parametric uncertainty leads to uncertainty in the system states. Therefore, $x(t) = x(t, \xi)$ is also a function of the random variables ξ.

Following the gPC method, the unknown parameters $\theta(\xi)$ and system states $x(t, \xi)$ are expanded in terms of orthogonal polynomial basis functions $\Phi_\alpha(\xi)$:

$$\theta(\xi) \approx \sum_{|\alpha|=0}^{S} \theta_\alpha \Phi_\alpha(\xi), \tag{4.4}$$

$$\hat{x}(t, \xi) \approx \sum_{|\alpha|=0}^{S} x_\alpha(t) \Phi_\alpha(\xi). \tag{4.5}$$

Here, the vector $\alpha := \left[\alpha_1, \dots, \alpha_{n_p}\right]$ is an n_p-dimensional multi-index, and $|\alpha|$ is the sum of the vector elements, i.e. $|\alpha| := \alpha_1 + \cdots + \alpha_{n_p}$. Each element α_i of α can take on a non-negative integer value between 0 and S. Under certain assumptions (see (Xiu and Karniadakis, 2002)), Equations (4.4) and (4.5) become exact in the L^2 sense as $S \to \infty$. An infinite expansion is not computationally attainable, so truncation is necessary, and (4.4) and (4.5) are only approximations.

The expansion coefficients θ_α, $|\alpha| \leq \mathrm{S}$ are chosen such that (4.4) is distributed according to the known parameter prior distribution $\bar{\rho}(\theta)$, and hence θ_α is known for all α. Polynomial chaos theory then solves for the coefficients $x_\alpha(t)$ of the polynomial chaos state expansion (4.5) using either the Galerkin (Ghanem and Spanos, 1991) or collocation approach (Xiu, 2007). Some helpful examples that use the Galerkin and collocation approaches can be found in (Ghanem and Spanos, 1991), (Sandu et al., 2006), (Li and Xiu, 2009), and (Pence et al., 2010).

4.1.1 Galerkin Approach

The Galerkin approach solves for the expansion coefficients $x_\alpha(t)$ by projecting the state equations (4.1) and (4.2) onto the polynomial chaos basis functions $\Phi_\alpha(\xi)$ *i.e.*,

$$\begin{gathered}\langle \dot{\hat{x}}(t,\xi), \Phi_\alpha(\xi)\rangle = \langle f\big(t, \hat{x}(t,\xi), u; \theta(\xi)\big), \Phi_\alpha(\xi)\rangle, \\ \langle \hat{x}(0), \Phi_\alpha(\xi)\rangle = \langle x_0, \Phi_\alpha(\xi)\rangle, \qquad |\alpha| \leq S.\end{gathered} \tag{4.6}$$

This results in a set of deterministic state equations having the state-expansion coefficients $x_\alpha(t)$ as the new state variables. These new deterministic state equations can be solved using numerical integration. The number of states in the new set of deterministic state equations is equal to the total number of state-expansion coefficients multiplied by the number n_s of original states. The total number of state-expansion coefficients $x_\alpha(t)$ (and polynomial chaos basis functions $\Phi_\alpha(\xi)$) is (Xiu and Karniadakis, 2002)

$$r \coloneqq \frac{(S+n_p)!}{S!\,n_p!}. \tag{4.7}$$

This number grows rapidly as the polynomial order S and/or the number of unknown parameters n_p increases. The inner product $\langle F(\xi), G(\xi)\rangle$ is an integral of the product of $F(\xi)$ and $G(\xi)$, integrated over the event space of the random variables ξ:

$$\langle F(\xi), G(\xi)\rangle \coloneqq \int G(\xi)\, F(\xi) W(\xi) d\xi. \tag{4.8}$$

The weighting function $W(\xi)$ depends on the choice of polynomial basis functions, and is generally equal to the prior distribution $\bar{\rho}(\xi)$ of the random variables ξ (Xiu and Karniadakis, 2002).

4.1.2 Collocation Approach

The collocation approach (Xiu, 2007) can be more straightforward to implement than the Galerkin method, especially for nonlinear systems (Sandu et al., 2006). However, it is generally less accurate than the Galerkin method (Xiu, 2007). A set of collocation points (or nodes) $\mu^{(1)}, \dots, \mu^{(Q)}$, ($Q \geq r$ and $\mu^{(i)} \in \mathbb{R}^{n_p}$) are drawn from the parameter prior distribution $\bar{\rho}(\xi)$. These collocation points are substituted for the random variables ξ in (4.1) and (4.2), *i.e.*,

$$\dot{z}^{(i)} = f\big(t, z^{(i)}(t); \theta(\mu^{(i)})\big), \qquad i = 1, \dots, Q \tag{4.9}$$

$$z^{(i)}(0) = x_0 \tag{4.10}$$

Here, $z^{(i)} \approx \sum_{|\alpha|=0}^{S} x_\alpha(t) \Phi_\alpha\big(\mu^{(i)}\big)$ is the i^{th} deterministic state vector. The resulting Q uncoupled sets of state equations (each set having n_s states) can be solved using numerical integration. Stacking the new sets of states $z^{(i)}$ into a matrix $Z \in \mathbb{R}^{Q\times n_s}$ yields:

$$Z \coloneqq \begin{bmatrix} \left(z^{(1)}\right)^T \\ \vdots \\ \left(z^{(Q)}\right)^T \end{bmatrix}. \tag{4.11}$$

The transpose of the state-expansion of (4.5) can be written as the vector-matrix product:

$$(\hat{x}(t,\xi))^T = \left(P(\xi)\right)^T \chi(t), \qquad \chi(t) \coloneqq \begin{bmatrix} \left(x_{|\alpha|=0}(t)\right)^T \\ \vdots \\ \left(x_{|\alpha|=S}(t)\right)^T \end{bmatrix}. \tag{4.12}$$

In this equation, if $\Phi_\alpha(\xi)$ is the k^{th} element of the column vector $P(\xi) \in \mathbb{R}^r$ then $(x_\alpha(t))^T$ is the k^{th} row of $\chi(t) \in \mathbb{R}^{r\times n_s}$. The matrix Z from (4.11) can be written in terms of P and χ as follows:

$$Z = \begin{bmatrix} \left(P(\mu_1)\right)^T \\ \vdots \\ \left(P(\mu_Q)\right)^T \end{bmatrix} \chi(t). \tag{4.13}$$

The estimates of $\chi(t)$., *i.e.* the estimates of the state expansion coefficients $x_\alpha(t)$, are obtained by left-multiplying both sides of (4.13) by the pseudo-inverse of $\left[P(\mu_1)|\cdots|P(\mu_Q)\right]^T$.

Another way to write the state expansion (4.5) that will be useful for concise notation in the following sections is to stack the columns of χ, *i.e.* $\chi^{(1)}, \dots, \chi^{(n_s)}$, into a single column vector. Then Equation (4.5) can be written as follows:

$$\hat{x}(t,\xi) = \mathbb{P}(\xi)X(t). \tag{4.14}$$

Here, $\mathbb{P}(\xi) \in \mathbb{R}^{n_s\times r\cdot n_s}$ and $X(t) \in \mathbb{R}^{r\cdot n_s}$, are defined respectively as

$$\mathbb{P}(\xi) := \begin{bmatrix} P(\xi)^T & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & P(\xi)^T \end{bmatrix}, \tag{4.15}$$

$$X(t) := \begin{bmatrix} \chi^{(1)}(t) \\ \vdots \\ \chi^{(n_s)}(t) \end{bmatrix}. \tag{4.16}$$

Sandu et al. (Sandu et al., 2006) provide a note on the relationship between the stochastic collocation and stochastic Galerkin methods, and they also suggest methods for implementing the Galerkin method on nonlinear systems.

In summary, the deterministic part of the system is calculated using either the Galerkin or collocation method. Solving the deterministic dynamic equations results in known trajectories of the time dependent part $x_\alpha(t)$ which is then recombined using (4.5) with the random variable dependent part $\Phi_\alpha(\xi)$ to obtain the complete stochastic solution $\hat{x}(t,\xi)$.

4.2 Recursive Parameter Estimation

This section derives the recursive parameter update law for estimating the maximum likelihood values of the random variables ξ given the system output observations. The estimates of the unknown parameters $\theta(\xi)$ are then calculated using (4.4). The derivations and resulting parameter estimators of this section constitute the main contributions of this chapter.

This development assumes that the noise in the system output observations is zero mean and Gaussian with known covariance matrix $R_k \in \mathbb{R}^{n_y \times n_y}$. It also assumes that the system observations y_k are mutually independent for all k. It assumes the uncertainty in (4.1) and (4.2) is entirely due to the unknown parameters. Finally it assumes that the polynomial chaos approximations in (4.4) and (4.5) are exact. (As mentioned above, this last assumption is satisfied as the number of expansion terms goes to infinity (Xiu and Karniadakis, 2002). In practice, the expansion must be truncated after a finite number of terms, and thus the parameter estimates via this method will only approximately satisfy the maximum likelihood criterion.) Under these assumptions, the likelihood function becomes (see Chapter 12 of (Moon and Stirling, 2000)):

$$\begin{aligned} \mathcal{L}_k(\xi|y_{0:k}) &= \prod_{\tau=0}^{k} \rho(y_\tau|\xi) \\ &\propto exp\left\{-\frac{1}{2}\sum_{\tau=0}^{k}\big(y_\tau - \hat{y}_\tau(\xi)\big)^T R_\tau^{-1}\big(y_\tau - \hat{y}_\tau(\xi)\big)\right\}. \end{aligned} \tag{4.17}$$

Here, $\mathcal{L}_k(\xi|y_{0:k})\colon \mathbb{R}^{n_p} \to \mathbb{R}$ is the scalar likelihood at time t_k of the unknown parameters ξ conditioned on a matrix $y_{0:k}$ which contains all of the output observations up to the current time t_k. The function $\rho(y_k|\xi)$ is the conditional probability of the observation y_k at time t_k given ξ. In general, $y_k \in \mathbb{R}^{n_y}$ is the argument of $\rho(y_k|\xi)$, but in calculating the likelihood (4.17), ξ is the argument of $\rho(y_k|\xi)\colon \mathbb{R}^{n_p} \to \mathbb{R}$ and y_k is assumed to be given. The vector $\hat{y}_k(\xi) \in \mathbb{R}^{n_y}$ is the output of the stochastic model, which, using (4.14) can be written as

$$\hat{y}_k = C\big(\theta(\xi)\big)\mathbb{P}(\xi)X(t_k). \tag{4.18}$$

The maximum likelihood estimate $\hat{\xi}$ is the realization of ξ that maximizes the likelihood function (4.17). Equation (4.17) is maximized when the magnitude of the negative term in the exponent, *i.e.*, the negative log-likelihood, is minimized:

$$J_k(\xi) \coloneqq \frac{1}{2}\sum_{\tau=0}^{k}\big(y_\tau - \hat{y}_\tau(\xi)\big)^T R_\tau^{-1}\big(y_\tau - \hat{y}_\tau(\xi)\big). \tag{4.19}$$

Thus the most likely value of ξ at time t_k is $\hat{\xi}_k = argmin\, J_k(\xi)$. The ability to update $J_k(\xi): \mathbb{R}^{n_p} \to \mathbb{R}$ iteratively is critical to making the approach of this chapter recursive. This chapter leverages the linearity of the output model (4.18) and the ability of polynomial chaos to separate the time and unknown parameter parts of the equation to make this recursion possible. By substituting (4.13) and (4.18) into (4.19) and performing a few algebraic manipulations, the objective function $J_k(\xi)$ can be written as:

$$J_k(\xi) := \frac{1}{2}\sum_{i=1}^{n_y}\sum_{j=1}^{n_y}\left(D_k^{y^{(i)}y^{(j)}} - 2C^{(i)}\mathbb{P}D_k^{Xy^{(j)}} + C^{(i)}\mathbb{P}D_k^{XX^T}\left(C^{(j)}\mathbb{P}\right)^T\right). \tag{4.20}$$

In Equation (4.20), the term D_k^G is defined as $D_k^G := \sum_{\tau=0}^{k}[R_\tau^{-1}]^{(i,j)}G_\tau$ where $G_k \in \left\{y_k^{(i)}y_k^{(j)}, X(t_k)y_k^{(j)}, X(t_k)\big(X(t_k)\big)^T\right\}$. The scalar term $[R_k^{-1}]^{(i,j)}$ is the j^{th} element in the i^{th} row of the inverse covariance matrix R_k^{-1}. Also, the scalar $y^{(l)}$ is the l^{th} element of the observation vector y_k and $C^{(l)}$ is the l^{th} row of the output matrix C. Equation (4.20) can be updated recursively from time t_k to t_{k+1} since $D_{k+1}^G = D_k^G + G_{k+1}$, and G_k is not a function of ξ.

The forgoing discussion has outlined a procedure for determining recursively $J_k(\xi)$, *i.e.* the term in the exponent of the likelihood function (4.17). The remaining challenge is to determine the value of ξ that minimizes $J_k(\xi)$, thus maximizing the likelihood at each time step. This can be viewed as an optimization problem in which the objective function $J_k(\xi)$ is time-varying. The following sections offer potential solution approaches.

4.2.1 Solution via Gradient Descent

Proceeding in a manner inspired by Southward's estimator (Southward, 2008), this chapter proposes a gradient based parameter update law.

$$\hat{\xi}_{k+1} = \hat{\xi}_k - \Gamma_k \frac{\partial J_k(\xi)}{\partial \xi}\bigg|_{\xi=\hat{\xi}} \tag{4.21}$$

Here Γ_k is a user-specified gain matrix that may vary in time. In static optimization, if Γ_k is the identity matrix, (4.21) is a steepest descent method. If Γ_k is the inverse Hessian matrix (matrix of second derivatives), then (4.21) is Newton's method, and if Γ_k is proportional to the Hessian matrix, then (4.21) is a modified Newton's method. These static optimization concepts may be helpful for selecting Γ_k.

As stated above, this gradient based approach is similar to the approach taken by Southward (Southward, 2008) (and Shimp (Shimp, 2008)). There is, however, an important difference. This chapter addresses maximum likelihood parameter estimation, and hence uses gradients of an *integrated cost function*, *i.e.* the cost function (4.19) is a function of all the data up to time t_k. Southward's method uses gradients of an *instantaneous cost function* and does not propose to maximize a likelihood function.

Substituting (4.20) into (4.21) and using the fact that the time-dependent parts $D_k^{y^{(i)}y^{(j)}}$, $D_k^{Xy^{(j)}}$, and $D_k^{XX^T}$ can be moved outside of the partial derivatives since they do not depend on the unknown parameters gives the following update law:

$$\hat{\xi}_{k+1} = \hat{\xi}_k - \Gamma_k \sum_{i=1}^{n_y}\sum_{j=1}^{n_y}[\frac{\partial C^{(i)}\mathbb{P}}{\partial \xi}\left(D_k^{XX^T}\left(C^{(j)}\mathbb{P}\right)^T - D_k^{Xy^{(j)}}\right)]_{\xi=\hat{\xi}_k}. \tag{4.22}$$

In practice, it may be helpful to normalize Equation (4.22) by dividing by $\left(1 + \left\|D_k^{Xy^{(j)}}\right\|_2\right)$. Then the final update law becomes:

$$\hat{\xi}_{k+1} = \hat{\xi}_k - \Gamma_k \sum_{i=1}^{n_y}\sum_{j=1}^{n_y} \left(\frac{[\frac{\partial C^{(i)}\mathbb{P}}{\partial \xi}\left(D_k^{XX^T}\left(C^{(j)}\mathbb{P}\right)^T - D_k^{Xy^{(j)}}\right)]_{\xi=\hat{\xi}_k}}{\left(1 + \left\| D_k^{Xy^{(j)}} \right\|_2\right)} \right). \tag{4.23}$$

The normalization scalar $\left(1 + \left\| D_k^{Xy^{(j)}} \right\|_2\right)^{-1}$ is not a function of the random variables ξ and therefore (4.23) still seeks to maximize the likelihood (4.17).

4.2.2 Solution via Random Search

The gradient descent solution of Section 3.1 does not guarantee that the estimated parameters will *globally* minimize $J_k(\xi)$, but may potentially select parameters satisfying only *local* minima. This subsection proposes a strategy for enabling the parameter update law to escape local minima in order to satisfy the global maximum likelihood criterion.

Using (4.20), the function $J_k(\xi)$ can be evaluated at time t_k for any realization of ξ. This enables the following strategy:

At time t_{k+1}, select an integer number $n_r > 0$ of realizations of ξ randomly, and evaluate the cost of each realization using (4.20). Then compare these costs with the cost of the current parameter estimate $\hat{\xi}_k$, and set the new estimate $\hat{\xi}_{k+1}$ to be the realization with the lowest cost. This random search strategy allows the algorithm to search any point in the entire parameter space and thus escape local minima.

A guided random search policy combines the random search with the gradient search: The cost of the gradient solution from (4.23) is compared with the costs of the n_r randomly selected realizations as well as the cost of the current estimate $\hat{\xi}_k$. The new estimate $\hat{\xi}_{k+1}$ is chosen to be the realization with the lowest cost.

4.3 Example (1): Nonlinear Oscillator

This section uses a simulation study of a nonlinear Van der Pol oscillator to demonstrate the proposed method and to compare it with a (hybrid) Extended Kalman Filter (EKF) (Section 13.2.2 of (Simon, 2006)). The equations for the Van der Pol oscillator are given in state-space representation as follows (page 57 of (Khalil, 2002)):

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -x_1 - \varepsilon({x_1}^2 - 1)x_2 \qquad (4.24) \\ y_k &= x_1(t_k) + v_k. \end{aligned}$$

The term $\varepsilon > 0$ represents a nonlinear damping coefficient, and also provides a measure of the nonlinearity of the system. The trajectory of the Van der Pol oscillator tends to a stable limit cycle. The initial conditions were $[x_1(0), x_2(0)]^T = [0.1, -0.1]$; this study assumes that only $x_1(0)$ is known and that the value of $x_2(0)$ is unknown, but prior information suggests that it could be any value between -0.2 and 0.2 with equal probability. Measurements of the output sequence y_k are taken at a uniform rate with a sampling step size of 0.01 seconds. The measurement noise v_k is an unknown normally distributed random sequence with constant variance R. Without the added noise, *i.e.* $v_k = 0$ for all k, the output signal had a mean-squared value of 1.9, and the (assumed unknown) noise variance R was set to be 8 times smaller, *i.e.*, $R = 0.24$. This simulation study assumes that $\varepsilon \sim U[0.3, 1.3]$ is uniformly distributed, and the true (but unknown) value is $\varepsilon = 1.1$.

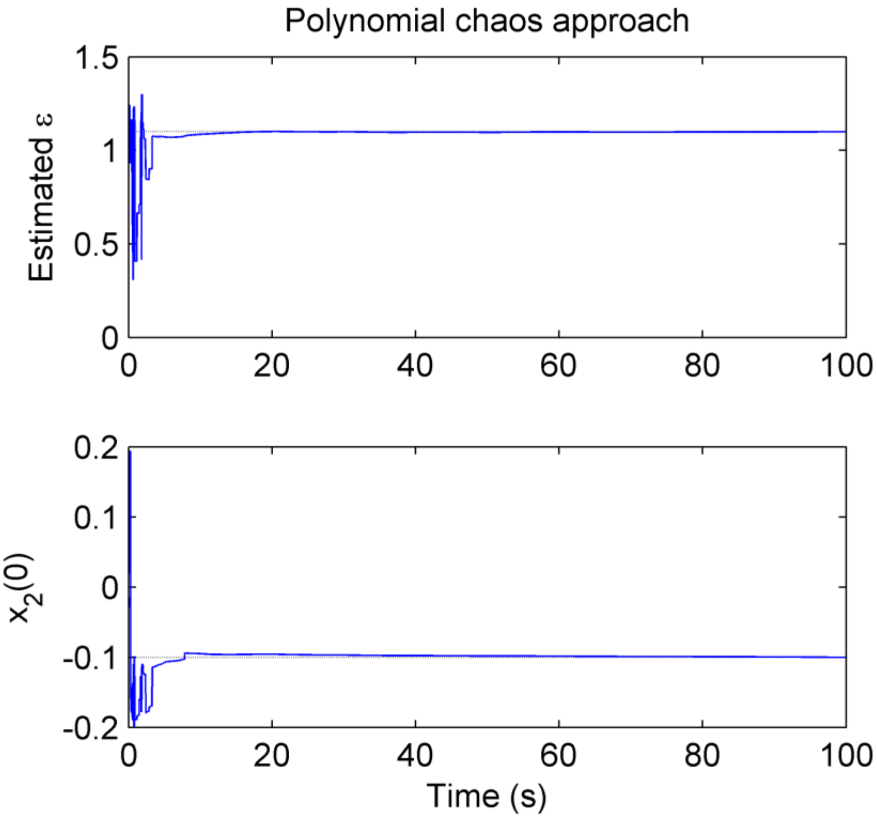


Figure 4.1: Convergence of the proposed polynomial chaos estimator. The dotted lines are the true values.

Because the noise variance R is assumed to be unknown, it becomes a user-specified tuning parameter in both the proposed polynomial chaos algorithm and the EKF filtering algorithm. Because the noise variance is a constant with respect to time and with respect to the unknown parameters, its assumed (positive) value has no effect on the value of ξ that minimizes the cost function shown in (4.19). It therefore has no effect on the estimate calculated by the proposed polynomial chaos approach. Thus, for all assumed values of R, the convergence of the proposed approach is shown in Figure 4.1.

Figure 4.2 shows the convergence of the EKF algorithm for different user-assumed values of R. Clearly, the EKF approach is sensitive to the assumed value of R, and hence is more difficult to tune than the polynomial chaos approach for this simulation example.

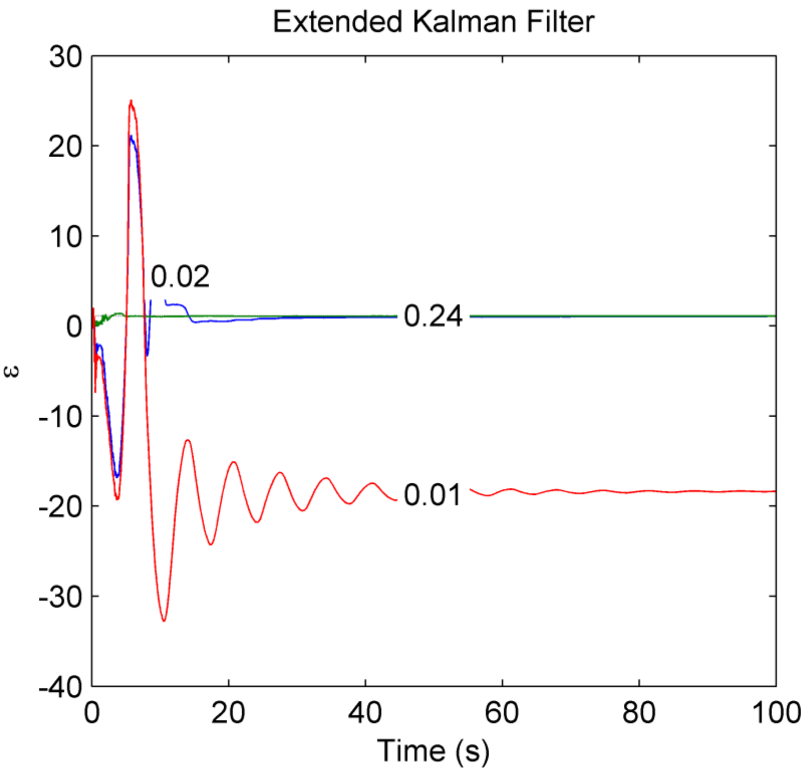


Figure 4.2: Convergence of EKF algorithm to various assumed values of *R* = 0.01, 0.02, 0.24. The dotted line is the true value.

4.4 Chapter 4 Conclusions

This chapter derived a recursive approach based on polynomial chaos theory for estimating the maximum likelihood values of unknown parameters of state space systems. The proposed method was demonstrated using a simulation of a nonlinear Van der Pol oscillator to illustrate the method and to show that it has the potential to be less difficult to tune than the filtering approach. Most importantly, because of its ability to recursively calculate the maximum likelihood values of unknown parameters in both linear and nonlinear systems, the authors believe that the approach will be valuable to a wide range of estimation problems.

Chapter 5

Recursive Bayesian Parameter Estimation using Polynomial Chaos Theory

This chapter introduces a recursive algorithm based on polynomial chaos theory that seeks Bayesian *a posteriori* estimates of unknown parameters of state-space systems given input and output data. The Bayesian parameter estimators of this chapter build on the ideas presented in Chapter 4; the likelihood function of Chapter 4 is an integral part of the Bayesian estimators. The main difference between the maximum likelihood estimator of Chapter 4 and the Bayesian estimators of this chapter is that the Bayesian approach explicitly uses the *a priori* parameter distribution information in calculating optimal parameter values.

Filtering approaches, *e.g.* the Extended Kalman Filter (EKF), Unscented Kalman Filter (UKF), and particle filter, provide potential solutions to the recursive Bayesian parameter estimation problem. To estimate the unknown parameters, these methods formally extend the state-vector to include estimates of the unknown parameters; this generally increases the nonlinearity of the system, especially if the system was originally linear. Like the filtering methods, the approaches of this chapter also result in a higher dimensional state equation, but the resulting expanded equations have the same degree of linearity/nonlinearity as the original system. For example, if the initial system is linear, the expanded system is also linear, etc.

The concept of using polynomial chaos theory for calculating Bayesian estimates of unknown parameters of state space system is not unique to this chapter; however, the *recursive* approach of this chapter is unique. Chapter 2 of this dissertation reviews the scientific literature on polynomial chaos based estimators. The approach of this chapter builds on the framework of these earlier approaches as well as the techniques of Chapter 4 to provide a novel recursive solution to Bayesian parameter estimation of state-space systems.

5.1 Bayesian Estimation

The development of the recursive Bayesian parameter estimator of this chapter requires the same assumptions as were made for the maximum likelihood estimator of Chapter 4. Also, the notation of this chapter adopts the notation from Chapter 4.

Bayes' rule provides the framework for Bayesian estimators. It describes how a prior parameter distribution $\bar{\rho}(\xi)\colon \mathbb{R}^{n_p} \to \mathbb{R}$ of a random vector $\xi \in \mathbb{R}^{n_p}$ evolves to its posterior parameter distribution $\bar{\rho}(\xi|y_{0:k})\colon \mathbb{R}^{n_p} \to \mathbb{R}$. The posterior distribution $\bar{\rho}(\xi|y_{0:k})$ is conditioned on all of the system observations $y_{0:k}$ up through time t_k. One representation of Bayes' rule is as follows (Blanchard et al., 2009):

$$\bar{\rho}(\xi|y_{0:k}) = \frac{1}{\int \mathcal{L}(\xi|y_{0:k})\bar{\rho}(\xi)d\xi}\mathcal{L}(\xi|y_{0:k})\bar{\rho}(\xi). \tag{5.1}$$

The function $\mathcal{L}(\xi|y_{0:k})\colon \mathbb{R}^{n_p} \to \mathbb{R}$ is the likelihood function, and for additive Gaussian noise assumptions, it is defined in Equation (4.17) of Chapter 4. The denominator is integrated over the event space of the random vector ξ (and hence is constant with respect to ξ). Using the notation from Chapter 4, the Gaussian likelihood can be written as follows:

$$\mathcal{L}(\xi|y_{0:k}) \propto exp\{-J_k(\xi)\}. \tag{5.2}$$

Section 4.2 described a process of recursively calculating the negative log likelihood $J_k(\xi): \mathbb{R}^{n_p} \to \mathbb{R}$ for state space systems. Since $J_k(\xi)$ can be calculated recursively using Equation (4.20), the likelihood (5.2) and the Bayes' posterior distribution, Equation (5.1), can also be calculated recursively (the prior distribution function $\bar{\rho}(\xi)$ is known and constant with respect to time).

5.1.1 Recursive MAP Estimator

The Maximum *A Posteriori* (MAP) estimate of the unknown parameter vector ξ is the realization of ξ that maximizes the posterior distribution $\bar{\rho}(\xi|y_{0:k})$. Since the denominator of (5.1) is constant with respect to ξ, the MAP estimator is given by

$$\xi_k^{(MAP)} = \underset{\xi}{\operatorname{argmax}}\ exp\{-J_k(\xi)\}\bar{\rho}(\xi). \tag{5.3}$$

Using the monotonic property of the logarithm function, the MAP estimator also satisfies the following equation (Blanchard et al., 2009):

$$\xi_k^{(MAP)} = \underset{\xi}{\operatorname{argmin}}\ \{J_k(\xi) - \log\left(\bar{\rho}(\xi)\right)\}. \tag{5.4}$$

The techniques used in Section 4.2 for finding the realization of ξ that minimizes $J_k(\xi)$ can also be used to find $\xi_k^{(MAP)}$.

5.1.2 Recursive MMSE Estimator

The Minimum Mean-Squared Error (MMSE) estimate of ξ is given as follows (see pages 350-354 of (Gubner, 2006)):

$$\xi_k^{(MMSE)} = E\{\xi|y_{0:k}\} = \frac{1}{\int \mathcal{L}(\xi|y_{0:k})\bar{\rho}(\xi)d\xi} \int \xi\mathcal{L}(\xi|y_{0:k})\bar{\rho}(\xi)d\xi. \tag{5.5}$$

Both integrals in (5.5) are integrated over the event space of ξ. Because these integrals must be evaluated at each time step, the MMSE estimator is more computationally demanding than the MAP and maximum likelihood estimators. Monte Carlo integration (see Section 6.12 of (Antia, 2002)) or other numerical integration techniques can be applied to numerically evaluate these integrals.

Higher order moments of the estimated posterior distribution $\bar{p}(\xi|y_{0:k})$ can be calculated in a similar manner. The m^{th} moment is calculated as follows:

$$E\{\xi^m|y_{0:k}\} = \frac{1}{\int \mathcal{L}(\xi|y_{0:k})\bar{p}(\xi)d\xi}\int \xi^m \mathcal{L}(\xi|y_{0:k})\bar{p}(\xi)d\xi. \tag{5.6}$$

These higher order moments provide valuable information about the statistical distribution of the unknown parameters. For example, the variance of the estimate is calculated using the MMSE estimate and the second order moment by $var(\xi) = E\{\xi^2|y_{0:k}\} - \left(\xi_k^{(MMSE)}\right)^2$. Like the MMSE estimate, calculations of the higher order moments require evaluating integrals at each time step and thus require more computational resources.

5.1.3 A Note on Numerical Implementation

This section provides a useful note on numerical implementation of the MMSE and higher-order-moments estimators Equations (5.5) and (5.6) respectively. First note, however, that the MMSE estimator of (5.5) is a special case of the higher order moment estimator of (5.6) with $m = 1$. Therefore, the same numerical techniques that apply to (5.6) also apply to (5.5). Substituting the likelihood function $\mathcal{L}(\xi|y_{0:k})$ from (5.2) into Equation (5.6) results in the following equation for the higher order moment estimator:

$$E\{\xi^m|y_{0:k}\} = \frac{1}{\int exp\{-J_k(\xi)\}\bar{p}(\xi)d\xi}\int \xi^m exp\{-J_k(\xi)\}\bar{p}(\xi)d\xi. \tag{5.7}$$

The numerical problem with Equation (5.7) is as follows: Because the objective function $J_k(\xi)$ is the sum over time of the *squared* difference between y and $\hat{y}$ (see Equation (4.19)), its value may tend to positive infinity as time increases. The exponential functions in (5.7) quickly (exponentially) converge to zero as the magnitude of $J_k(\xi)$ increases. Thus, the integrals in the numerator and denominator of (5.7) quickly become smaller than the precision of the computer, and Equation (5.7) becomes a numerical problem of dividing zero by zero.

To avoid this issue of dividing zero by zero, this section formulates Equation (5.7) in the following manner by multiplying by one:

$$E\{\xi^m|y_{0:k}\}$$
$$= \frac{1}{\int exp\{-J_k(\xi)\}\bar{\rho}(\xi)d\xi}\int \xi^m exp\{-J_k(\xi)\}\bar{\rho}(\xi)d\xi \frac{exp\left\{J_k\left(\xi_{k-1}^{(MMSE)}\right)\right\}}{exp\left\{J_k\left(\xi_{k-1}^{(MMSE)}\right)\right\}}. \tag{5.8}$$

Here, $J_k\left(\xi_{k-1}^{(MMSE)}\right)$ is the current cost of the previous MMSE estimate $\xi_{k-1}^{(MMSE)}$. The term $exp\left\{J_k\left(\xi_{k-1}^{(MMSE)}\right)\right\}$ is constant with respect to ξ and can therefore be moved inside the integrals of (5.8). The equation can then be written

$$E\{\xi^m|y_{0:k}\} = \frac{1}{\int exp\left\{-J_k(\xi) + J_k\left(\xi_{k-1}^{(MMSE)}\right)\right\}\bar{\rho}(\xi)d\xi}\int \xi^m exp\left\{-J_k(\xi)\right.$$
$$\left. + J_k\left(\xi_{k-1}^{(MMSE)}\right)\right\}\bar{\rho}(\xi)d\xi. \tag{5.9}$$

The term in the exponent $-J_k(\xi) + J_k\left(\xi_{k-1}^{(MMSE)}\right)$ is small for some ξ (its value is zero for $\xi = \xi_{k-1}^{(MMSE)}$) and thus Equation (5.9) avoids the problem of dividing zero by zero assuming $\xi_{k-1}^{(MMSE)}$ is one of the numerical integration points.

5.2 Example (1) Revisited

This section uses the Bayesian techniques developed in this chapter to estimate the unknown parameters of the nonlinear oscillator of Example (1). Because the prior distribution $\bar{\rho}(\xi)$ is uniform (and hence constant when $\xi_1 \in [-1,1]$ and $\xi_2 \in [-1,1]$), the Bayesian MAP estimator is identical to the maximum likelihood estimator of Chapter 4 for $\xi_1 \in [-1,1]$ and $\xi_2 \in [-1,1]$. Therefore, similar to the maximum likelihood estimator, the MAP estimator does not depend on the assumed value of the noise variance R. However, the MMSE estimator *does* depend on the assumed value of R. The results of the MMSE estimator shown in Figure 5.1 are for $R = 0.24$. The Monte Carlo integration technique (see Section 6.12 of (Antia, 2002)) with 24 randomly sampled points was used to evaluate the integrals in the MMSE estimator. In addition, one integration point at each iteration was chosen to be $\xi_{k-1}^{(MMSE)}$.

In the Matlab® 2008a simulation environment, the time required to compute the estimate based on the MMSE estimator with 25 integration points required roughly 3.8 times the amount of Central Processing Unit (CPU) time than the maximum likelihood estimator of Chapter 4. The MMSE estimator required 64 seconds and the maximum likelihood estimator required 17 seconds. The code for the estimators was not optimized for numerical efficiency, and the Matlab® 2008a simulation environment is generally not as fast as compiled code. Therefore this CPU time evaluation is only used to suggest that the MMSE estimator is more computationally demanding than the maximum likelihood estimator, but the time ratio and time requirements are likely to be different in actual real-time implementation than those listed here.

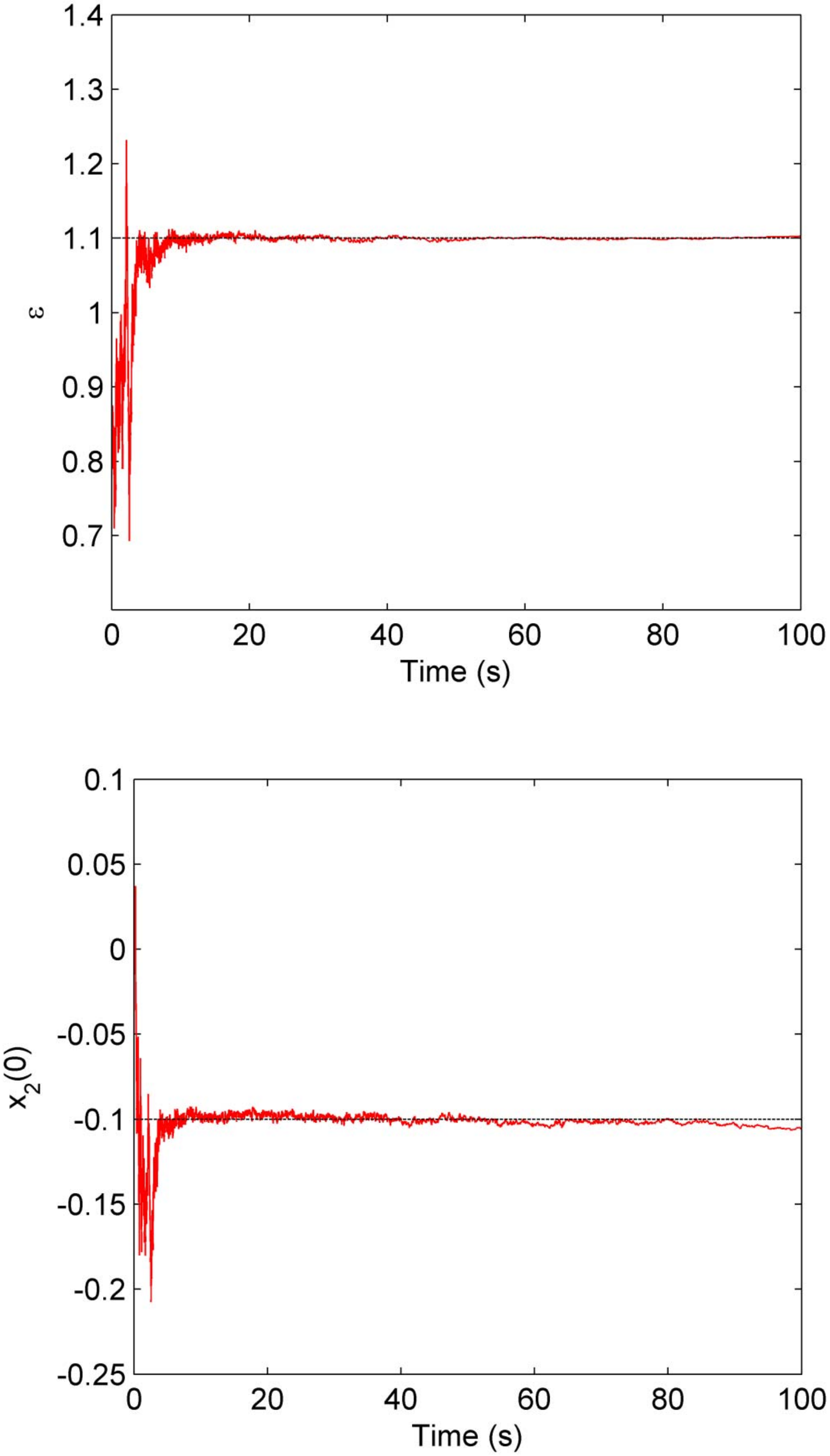


Figure 5.1: Convergence of the MMSE estimator for ε and $x_2(0)$. The dotted line is the true value.

The main benefit of using the Bayesian estimator is that it has the ability to quantify the uncertainty in the parameter estimates. Figure 5.2 shows the estimated values of the standard deviations of the estimates of ε and $x_2(0)$ assuming $R = 0.24$.

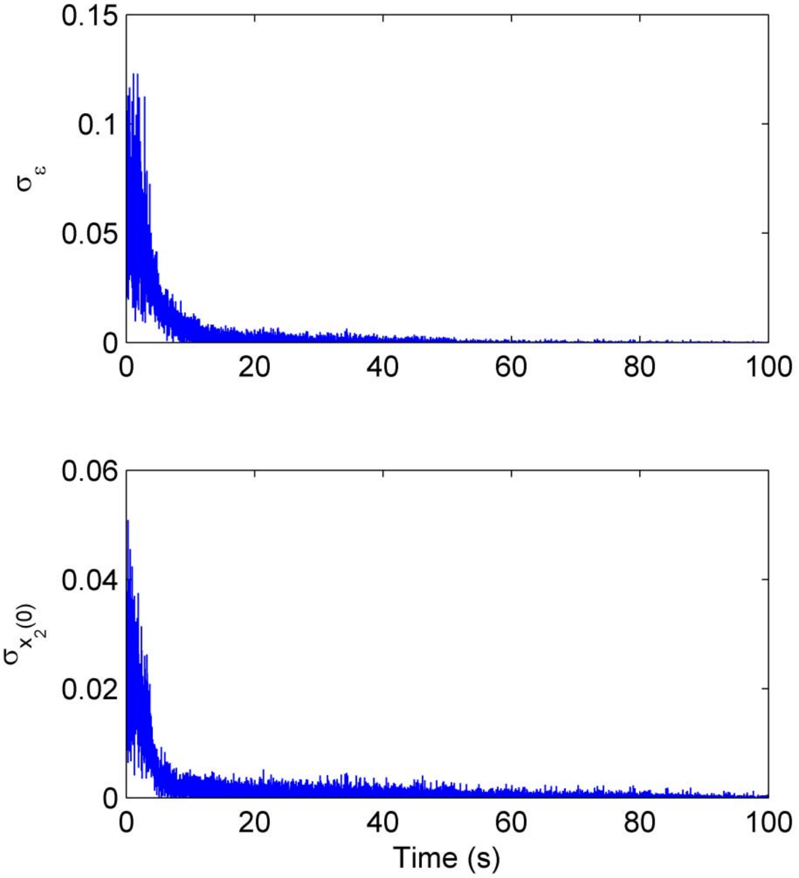


Figure 5.2: Estimates of the standard deviations of the parameter estimates.

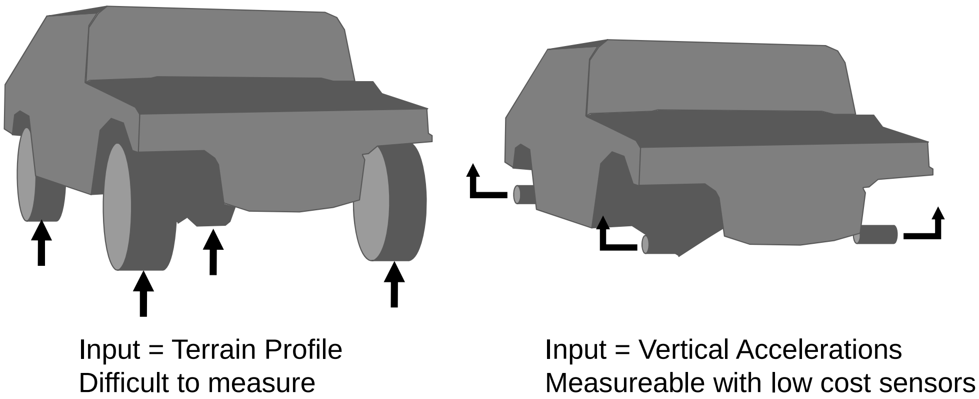
5.3 Chapter 5 Conclusions

This chapter introduced algorithms for recursively calculating the Bayesian estimates of unknown parameters of state-space systems. The MAP, MMSE, and higher-order moment estimators were discussed. In addition to calculating the most likely values of the unknown parameters, this chapter also presented methods for estimating their statistical distributions. The MMSE and higher-order moment estimators are more computationally demanding than the MAP estimator of this chapter and the maximum likelihood estimator of Chapter 4. The methods of this chapter can be applied to a wide range of both linear and nonlinear systems, and thus may be valuable to many parameter estimation problems.

Chapter 6

Base Excitation Modeling of Vehicle Ride Dynamics

This chapter introduces base excitation models of vehicle ride dynamics. These base excitation models enable mass estimation without requiring prior knowledge or real-time estimation of the terrain input. The base excitation models treat the unsprung mass motions instead of the terrain as the model input (as illustrated in Figure 6.1).



Input = Terrain Profile
Difficult to measure

Input = Vertical Accelerations
Measureable with low cost sensors

Figure 6.1: Base excitation concept.

The base excitation concept is adopted from the vibrations literature (see Section 2.4 of (Inman, 2001)). This chapter reviews the quarter-car base excitation model and then introduces the base excitation half and full-car models for mass estimation. The base excitation quarter-car model has been used previously for vehicle sprung mass estimation for vehicles with adaptive suspension components (Song et al., 2005).

6.1 Quarter-Car Model

A two degree-of-freedom quarter-car model of suspension dynamics is shown in Figure 6.2 (see Chapters 10-12 of (Rajamani, 2006)). The vertical motion z_g at the tire/ground contact surface is the input to this model.

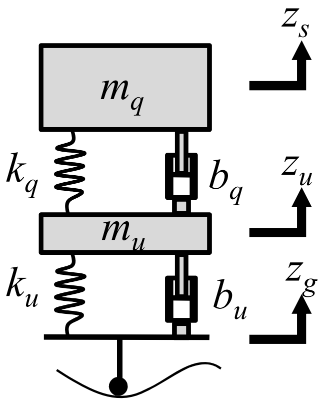


Figure 6.2: Quarter-car model of vehicle ride dynamics.

The base excitation model of quarter-car dynamics differs from the traditional model by treating the unsprung mass motion, instead of the terrain profile, as the model input. The base-excitation quarter-car model is shown in Figure 6.3. The advantage of this model for real-time mass estimation is that the unsprung mass motion may be more easily measured than the terrain profile.

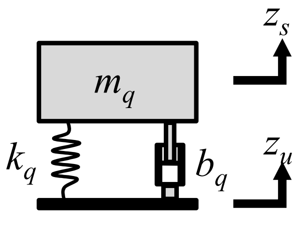


Figure 6.3: Base excitation quarter-car model.

Defining the states x_1 and x_2 to be the relative suspension displacement $(z_s - z_u)$ and velocity $(\dot{z}_s - \dot{z}_u)$ respectively, the input u to be the unsprung acceleration $\ddot{z}_u$, and the output y to be the sprung mass acceleration $\ddot{z}_s$, the state and output equations are as follows:

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix}=\begin{bmatrix}0 & 1\\ \dfrac{-k_q}{m_q} & \dfrac{-b_q}{m_q}\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}+\begin{bmatrix}0\\ -1\end{bmatrix}u$$

$$y=\begin{bmatrix}\dfrac{-k_q}{m_q} & \dfrac{-b_q}{m_q}\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}. \tag{6.1}$$

The low system order of this quarter-car model makes it attractive for real-time estimation in which computational resources may be limited.

6.2 Half-Car Model

The half-car model includes the pitch and vertical motion of vehicle ride dynamics. The development of the half-car model in this section assumes linear suspension elements, left-right symmetry in suspension components, and negligible coupling between longitudinal or roll motion and the vertical/pitch dynamics. The base excitation model of half-car dynamics is shown in Figure 6.4.

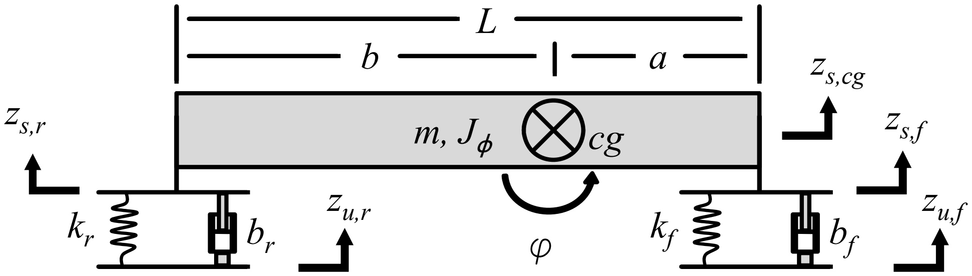


Figure 6.4: Base excitation half-car model.

In Figure 6.4, $z_{u,f}$ ($z_{u,r}$) is the average vertical position of the two front (rear) unsprung masses, $z_{s,f}$ ($z_{s,r}$) is the vertical position of a point on the body of the vehicle that is located directly above the front (rear) axle and centered between the two front (rear) wheels, $z_{s,cg}$ is the vertical position of the center of gravity of the vehicle, J_ϕ is the pitch moment of inertia, m is the vehicle sprung mass, L is the wheelbase length, and the lengths a and b are respectively the distances from the front and rear axles to the center of gravity, cg. The front (rear) stiffness coefficient k_f (k_r) is the sum of the two front (rear) suspension stiffness coefficients, and the front (rear) damping coefficient b_f (b_r) is the sum of the two front (rear) suspension damping coefficients. Assuming small angles, the pitch ϕ can be written as a function of the vertical positions $z_{s,f}$ and $z_{s,r}$ of the front and rear respectively of the vehicle:

$$\phi = \frac{z_{s,f} - z_{s,r}}{L}. \tag{6.2}$$

The vertical position of the center of gravity (cg) can also be written as a function of the two vertical positions $z_{s,f}$ and $z_{s,r}$:

$$z_{s,cg} = \frac{bz_{s,f} + az_{s,r}}{L}. \tag{6.3}$$

The dynamic equations that govern the motion of the base excitation half-car model are given as follows:

$$\begin{aligned} m\ddot{z}_{s,cg} &= -k_f\left(z_{s,f} - z_{u,f}\right) - b_f\left(\dot{z}_{s,f} - \dot{z}_{u,f}\right) - k_r\left(z_{s,r} - z_{u,r}\right) - b_r\left(\dot{z}_{s,r} - \dot{z}_{u,r}\right) \\ J_\phi\ddot{\phi} &= -\left(k_f\left(z_{s,f} - z_{u,f}\right) + b_f\left(\dot{z}_{s,f} - \dot{z}_{u,f}\right)\right)a \\ &\quad + \left(k_r\left(z_{s,r} - z_{u,r}\right) + b_r\left(\dot{z}_{s,r} - \dot{z}_{u,r}\right)\right)b. \end{aligned} \tag{6.4}$$

The states of the base excitation half-car model are as follows: $x_1 = z_{s,f} - z_{u,f}$ is the deflection in the front suspension; $x_2 = \dot{x}_1$ is the relative velocity in the front suspension; $x_3 = z_{s,r} - z_{u,r}$ is the deflection of the rear suspension; and $x_4 = \dot{x}_3$ is the relative velocity in the rear suspension. The system inputs u_f and u_r are the vertical accelerations $\ddot{z}_{u,f}$ and $\ddot{z}_{u,r}$ of the front and rear unsprung masses respectively. Then, using (6.2) and (6.3), Equation (6.4) can be written in the following state-space form:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\left(\frac{a^2}{J_\phi}+\frac{1}{m}\right)k_f & -\left(\frac{a^2}{J_\phi}+\frac{1}{m}\right)b_f & \left(\frac{ab}{J_\phi}-\frac{1}{m}\right)k_r & \left(\frac{ab}{J_\phi}-\frac{1}{m}\right)b_r \\ 0 & 0 & 0 & 1 \\ \left(\frac{ab}{J_\phi}-\frac{1}{m}\right)k_f & \left(\frac{ab}{J_\phi}-\frac{1}{m}\right)b_f & -\left(\frac{b^2}{J_\phi}+\frac{1}{m}\right)k_r & -\left(\frac{b^2}{J_\phi}+\frac{1}{m}\right)b_r \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ -1 & 0 \\ 0 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} u_f \\ u_r \end{bmatrix} \tag{6.5}$$

Choosing the output signals to be the vertical accelerations $y_1 = \ddot{z}_{s,f}$ and $y_2 = \ddot{z}_{s,r}$ of the sprung mass above the front and rear axles, the output equations are $y_1 = \dot{x}_2 + u_f$ and $y_2 = \dot{x}_4 + u_r$. If the acceleration at the sprung mass of the vehicle is chosen as the output signal y, then the output equation is $y = b(\dot{x}_2 + u_f)/L + a(\dot{x}_4 + u_r)/L$. The equation for the output y can be written in terms of the states in (6.5) as follows:

$$y = \begin{bmatrix} \left(\frac{ab^2 - a^3}{J_\phi L}-\frac{1}{m}\right)k_f & \left(\frac{ab^2 - a^3}{J_\phi L}-\frac{1}{m}\right)b_f & \left(\frac{a^2 b - b^3}{J_\phi L}-\frac{1}{m}\right)k_r & \left(\frac{a^2 b - b^3}{J_\phi L}-\frac{1}{m}\right)b_r \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \tag{6.6}$$

The dynamics of the front and rear suspensions can be decoupled under a special case in which $J_\phi = mab$ (see pages 315-321 of (Rajamani, 2006)). Substituting $J_\phi = mab$ into Equation (6.5) results in two decoupled sets of quarter-car equations (see Equation (6.1)):

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \dfrac{-Lk_f}{bm} & \dfrac{-Lb_f}{bm} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ -1 \end{bmatrix} u_f$$

$$\begin{bmatrix} \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \dfrac{-Lk_r}{am} & \dfrac{-Lb_r}{am} \end{bmatrix} \begin{bmatrix} x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ -1 \end{bmatrix} u_r \tag{6.7}$$

The choice of output signals $y_1 = \ddot{z}_{s,f}$ and $y_2 = \ddot{z}_{s,r}$ results in the following decoupled output equations:

$$y_1 = \begin{bmatrix} \dfrac{-Lk_f}{bm} & \dfrac{-Lb_f}{bm} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$y_2 = \begin{bmatrix} \dfrac{-Lk_r}{am} & \dfrac{-Lb_r}{am} \end{bmatrix} \begin{bmatrix} x_3 \\ x_4 \end{bmatrix} \tag{6.8}$$

In this special case, only one set of state and output equations is needed to estimate the vehicle sprung mass. This may allow a designer to reduce costs by using fewer sensors. Alternatively, a second estimator may offer validation and/or redundancy.

6.3 Base Excitation Full Car Ride Model

The vehicle dynamics literature commonly uses a seven degree of freedom system to model the general behavior of a vehicle's ride dynamics. An illustration of this model is provided in Figure 6.5. The seven degrees of freedom include the vertical $z_{s,cg}$, roll θ_s, and pitch φ_s motion of the sprung mass, as well as the vertical motions $z_{u,fl}, z_{u,fr}, z_{u,rl}, z_{u,rr}$ of the four unsprung masses. The motions $z_{g,fl}, z_{g,fr}, z_{g,rl}, z_{g,rr}$ of the four tire surfaces at the ground are the inputs to the seven degree of freedom model.

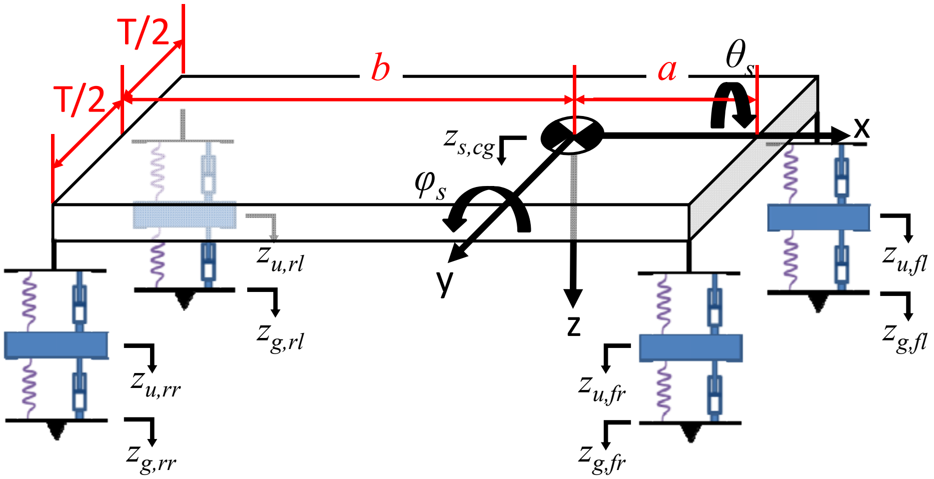


Figure 6.5: Seven degree of freedom model of vehicle ride dynamics.

As a key difference from the traditional model, the full-car base excitation model (as illustrated in Figure 6.6) treats the vertical unsprung mass accelerations $\ddot{z}_{u,fl}, \ddot{z}_{u,fr}, \ddot{z}_{u,rl}, \ddot{z}_{u,rr}$, the longitudinal velocity U in the x-direction with respect to body fixed axes, and the sprung mass pitch velocity $\dot{\varphi}_s$ as measured model inputs. (A reduced order model at the end of this section will require only the unsprung mass accelerations as the system inputs). The resulting model describes the dynamics of the sprung mass in the following three degrees of freedom: vertical $z_{s,cg}$, pitch φ_s, and roll θ_s. The base excitation model is shown in Figure 6.6.

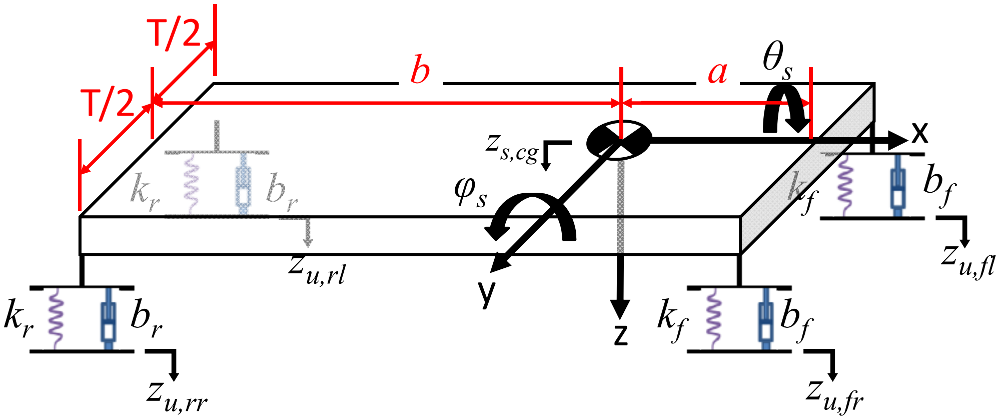


Figure 6.6: Base excitation model of full car ride dynamics.

The base excitation model has a number of key benefits for vehicle mass estimation compared with the traditional full-car ride model: the model avoids requiring knowledge of the values of the unsprung masses, tire stiffness and damping, and, most importantly, terrain profile. Also, the reduced degrees of freedom result in fewer model states and hence, less computational complexity.

This section makes the following simplifying assumptions:

1. Negligible yaw velocity ($\dot{\psi} = 0$)
2. Negligible lateral velocity ($V = 0$)
3. Small pitch φ and roll θ angles
4. Linear suspension elements
5. Left-right symmetry in suspension elements, *e.g.*, the front left spring stiffness is the same as the front right spring stiffness, *e.g.*, $k_{fl} = k_{fr} =: k_f$.
6. The c.g. is at half the track width $T/2$, a known distance a behind the front axle of the vehicle, and a distance b forward from the rear axle. The wheelbase length is $a + b$.
7. The sprung mass is a rigid body.

Applying Euler's laws of motion to the sprung mass of Figure 6.6 results in the following equations which govern the dynamic behavior of the base excitation full car model:

$$\begin{gathered} z - equation\colon m\left(\ddot{z}_{s,cg} - \dot{\varphi}_s U\right) = F_z \\ \varphi - equation\colon J_\varphi \ddot{\varphi}_s = M_\varphi \\ \theta - equation\colon J_\theta \ddot{\theta}_s = M_\theta. \end{gathered} \tag{6.9}$$

Here, F_z is the net downward force acting on the sprung mass, M_φ is the net moment or torque about the y axis, and M_θ is the moment acting about the x axis. These forces and moments are produced via the suspension elements as follows:

$$F_z := \sum_{i=fl,fr,rl,rr} F_i \tag{6.10}$$

$$M_\varphi := -\left(F_{fl} + F_{fr}\right)a + \left(F_{rl} + F_{rr}\right)b \tag{6.11}$$

$$M_\theta := -\frac{T}{2}\left(F_{fl} + F_{rl}\right) + \frac{T}{2}\left(F_{fr} + F_{rr}\right). \tag{6.12}$$

The forces F_i, $i = fl, fr, rl, rr$ are due to suspension deflections and velocities and are calculated by

$$F_{fl} := k_f\left(z_{u,fl} - z_{s,cg} + a\varphi_s + \frac{T}{2}\theta_s\right) + b_f\left(\dot{z}_{u,fl} - \dot{z}_{s,cg} + a\dot{\varphi}_s + \frac{T}{2}\dot{\theta}_s\right) \tag{6.13}$$

$$F_{fr} := k_f\left(z_{u,fr} - z_{s,cg} + a\varphi_s - \frac{T}{2}\theta_s\right) + b_f\left(\dot{z}_{u,fr} - \dot{z}_{s,cg} + a\dot{\varphi}_s - \frac{T}{2}\dot{\theta}_s\right) \tag{6.14}$$

$$F_{rl} := k_r\left(z_{u,rl} - z_{s,cg} - b\varphi_s + \frac{T}{2}\theta_s\right) + b_r\left(\dot{z}_{u,rl} - \dot{z}_{s,cg} - b\dot{\varphi}_s + \frac{T}{2}\dot{\theta}_s\right) \tag{6.15}$$

$$F_{rr} := k_r\left(z_{u,rr} - z_{s,cg} - b\varphi_s - \frac{T}{2}\theta_s\right) + b_r\left(\dot{z}_{u,rr} - \dot{z}_{s,cg} - b\dot{\varphi}_s - \frac{T}{2}\dot{\theta}_s\right). \tag{6.16}$$

For convenience, this section defines the following terms:

$$z_{u,cg} := \frac{b\left(z_{u,fl} + z_{u,fr}\right) + a\left(z_{u,rl} + z_{u,rr}\right)}{2(a+b)} \tag{6.17}$$

$$\varphi_u := \frac{-\left(z_{u,fl} + z_{u,fr}\right) + \left(z_{u,rl} + z_{u,rr}\right)}{2(a+b)} \tag{6.18}$$

$$Z := z_{s,cg} - z_{u,cg} \tag{6.19}$$

$$\Phi := \varphi_s - \varphi_u. \tag{6.20}$$

The terms $z_{u,cg}$ and φ_u can be interpreted as the average vertical position and average pitch of the unsprung masses. Then, the terms Z and Φ can be interpreted as the average vertical and angular displacements of the sprung mass relative to the unsprung masses.

Given definitions (6.10) through (6.20), the z and φ equations of (6.9) can be written as:

$$\begin{gathered} z-equation\colon m\ddot{Z}=F_z+m\left(\dot{\varphi}_s U-\ddot{z}_{u,cg}\right) \\ \varphi-equation\colon J_\varphi\ddot{\Phi}=M_\varphi-J_\varphi\ddot{\varphi}_u. \end{gathered} \tag{6.21}$$

The vertical force F_z and pitch moment M_φ can also be written as functions of the terms defined in (6.17) through (6.20):

$$F_z=-2\left(k_f+k_r\right)Z-2\left(b_f+b_r\right)\dot{Z}+2\left(ak_f-bk_r\right)\Phi+2\left(ab_f-bb_r\right)\dot{\Phi} \tag{6.22}$$

$$\begin{aligned} M_\varphi=2\left(ak_f-bk_r\right)Z+2\left(ab_f-bb_r\right)\dot{Z}-2\left(a^2k_f+b^2k_r\right)\Phi \\ -2\left(a^2b_f+b^2b_r\right)\dot{\Phi}. \end{aligned} \tag{6.23}$$

Equations (6.21) through (6.23) are independent of the roll motion of the sprung mass. As a result, the roll dynamics can be neglected without sacrificing accuracy in the calculation of the vertical and pitch motion. This result enables a lower order (and less computationally expensive) estimator.

The state space representation of Equations (6.21) through (6.23) is as follows:

$$\begin{aligned} \begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\\ \dot{x}_4\end{bmatrix} = & \begin{bmatrix} 0 & 1 & 0 & 0 \\ \dfrac{-2\left(k_f+k_r\right)}{m} & \dfrac{-2\left(b_f+b_r\right)}{m} & \dfrac{2\left(ak_f-bk_r\right)}{m} & \dfrac{2\left(ab_f-bb_r\right)}{m} \\ 0 & 0 & 0 & 1 \\ \dfrac{2\left(ak_f-bk_r\right)}{J_\varphi} & \dfrac{2\left(ab_f-bb_r\right)}{J_\varphi} & \dfrac{-2\left(a^2k_f+b^2k_r\right)}{J_\varphi} & \dfrac{-2\left(a^2b_f+b^2b_r\right)}{J_\varphi} \end{bmatrix} \begin{bmatrix}x_1\\ x_2\\ x_3\\ x_4\end{bmatrix} \\ & + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ \dfrac{-b}{2L} & \dfrac{-b}{2L} & \dfrac{-a}{2L} & \dfrac{-a}{2L} & 1 \\ 0 & 0 & 0 & 0 & 0 \\ \dfrac{1}{2L} & \dfrac{1}{2L} & \dfrac{-1}{2L} & \dfrac{-1}{2L} & 0 \end{bmatrix} \begin{bmatrix}\ddot{z}_{u,fl}\\ \ddot{z}_{u,fr}\\ \ddot{z}_{u,rl}\\ \ddot{z}_{u,rr}\\ \dot{\varphi}_s U\end{bmatrix}. \end{aligned} \tag{6.24}$$

The states are $x_1 = Z$, $x_2 = \dot{Z}$, $x_3 = \Phi$, and $x_4 = \dot{\Phi}$. The set of dynamic state equations in (6.24) govern the motion of the base excitation system shown in Figure 6.6. The resulting dynamic equations are similar to the half car model equations of (6.5). However, the states are defined differently, and Equation (6.24) accounts for the vertical acceleration component $\dot{\varphi}_s U$.

The dynamic equations of (6.24) can be combined with state-space estimation techniques like those of Chapters 4 and 5 for a potential solution to vehicle mass estimation. There are a few factors, however, that motivate reducing this set of state equations/sensor inputs in (6.24) to a lower dimensional set. The first motivating factor is the desire to reduce the computational complexity of the estimator. This is important for online/onboard algorithms which may have limited access to memory and computational resources on an onboard computer. The second factor is motivated by the desire to reduce the number of sensors required to perform the online estimation. Based on these motivating factors, the remainder of this section derives a reduced order model.

To reduce the set of state equations/sensor inputs to a lower dimensional set, this paper notes that in many vehicles, the coupling term between the vertical and pitch dynamics may be negligible. In Equation (6.24), the vertical and pitch dynamics are coupled by the terms $2(ak_f - bk_r)/m$, $2(ab_f - bb_r)/m$, $2(ak_f - bk_r)/J_\varphi$ and $2(ab_f - bb_r)/J_\varphi$. In some cases, the numerator of each term may be small, and the denominator is large. For example, if $ak_f = bk_r$, and $ab_f = bb_r$ the numerators are zero and there is no coupling between the vertical and pitch dynamics. When these coupling terms are not weak, a stronger assumption is required, which is to assume that the pitch motion of the vehicle is negligible. Either of these assumptions enables decoupling of the pitch and vertical dynamics, and the set of equations given in (6.24) can be reduced to the following second order state equations that govern the vertical motion of the sprung mass:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \frac{-2(k_f + k_r)}{m} & \frac{-B_z}{m} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ -1 \end{bmatrix} u$$

$$u = \begin{bmatrix} 0 & 0 & 0 & 0 \\ \frac{b}{2(a+b)} & \frac{b}{2(a+b)} & \frac{a}{2(a+b)} & \frac{a}{2(a+b)} \end{bmatrix} \begin{bmatrix} \ddot{z}_{u,fl} \\ \ddot{z}_{u,fr} \\ \ddot{z}_{u,rl} \\ \ddot{z}_{u,rr} \end{bmatrix}. \tag{6.25}$$

Here, $B_z = 2(b_f + b_r)$. The measured output y is the vertical acceleration $\ddot{z}_{s,cg}$ of the sprung mass at the center of gravity location, and the output equation is as follows:

$$y = \begin{bmatrix} \frac{-2(k_f + k_r)}{m} & \frac{-B_z}{m} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}. \tag{6.26}$$

The known or measured variables/parameters of equations (6.25) – (6.26) are the four unsprung mass accelerations, the distances from the axles to the sprung mass c.g. location, and the front and rear spring stiffness values. The unknown parameters/variables include the value of the vehicle sprung mass m, the damping term B_z, and the state variables x_1 and x_2.

6.4 Experimental Validation

This section compares the base excitation half-car model of Equations (6.5) – (6.6) and the reduced order full-car model of Equations (6.25) – (6.26) with the response of an actual vehicle. The experimental setup is described in Section 8.1 of this dissertation. The vehicle parameters for Equations (6.5) – (6.6) and (6.25) – (6.26) are given in Table 6.1.

Table 6.1: Model parameters for the half-car and reduced order full-car base excitation equations.

Parameter	Half-car Eq. (6.5) – (6.6)	Full-car Eq. (6.25) – (6.26)
Sprung mass	$m = 1295$ kg	$m = 1295$ kilograms
Suspension stiffness	$k_f = 83600$ N/m $k_r = 63400$ N/m	$Total\ Stiffness =$ 147000 N/m
Pitch inertia	$J_\phi = 2192$ kg m^2	Not Applicable
Damping	$b_f = 7000$ Ns/m $b_r = 6000$ Ns/m	$B_z = 13000$ Ns/m
C.G. location	$a = 1.2$ m $b = 1.42$ m	$a = 1.2$ m $b = 1.42$ m
Wheelbase length	$L = 2.62$ m	$L = 2.62$ m

Table 6.2 shows validation results of the reduced full-car and half-car base excitation models for a range of road profiles and vehicle speeds. The Signal-to-Noise Ratio (SNR) is used as a tool for analyzing how well the model outputs match the actual vehicle outputs. The SNR is calculated as follows:

$$SNR = \left(\frac{RMS(Y)}{RMS(Y - Y_m)}\right)^2 \tag{6.27}$$

Here, $Y \in \mathbb{R}^N$ is a vector containing all of the actual vehicle outputs (sprung mass acceleration measurements) and $Y_m \in \mathbb{R}^N$ is a vector containing all of the corresponding model outputs. The term RMS denotes the Root Mean Square value and is calculated as follows:

$$RMS(Y) = \frac{1}{\sqrt{N}} \sqrt{\sum_{i=1}^{N} y_i^2} \tag{6.28}$$

Here, y_i is the i^{th} element of Y.

Table 6.2 lists the results of the experiment in order from best SNR to worst SNR for the reduced full-car model. The SNR values for the reduced order full-car model of (6.25) – (6.26) and the half-car model of (6.5) – (6.6) are shown in the last two columns of the table. The range of SNR values for the reduced full-car model spans from about 0.27 to about 24, and for the half-car model spans from about 0.30 to about 41.

Figure 6.7 shows the output results of the models for the first row in Table 6.2 – the experiment in which the models best match (in the SNR sense) the actual vehicle. Figure 6.8 shows the results for Road No. 21 (halfway through Table 6.2). Finally, Figure 6.9 shows the results for the worst SNR case, Road No. 40 in Table 6.2.

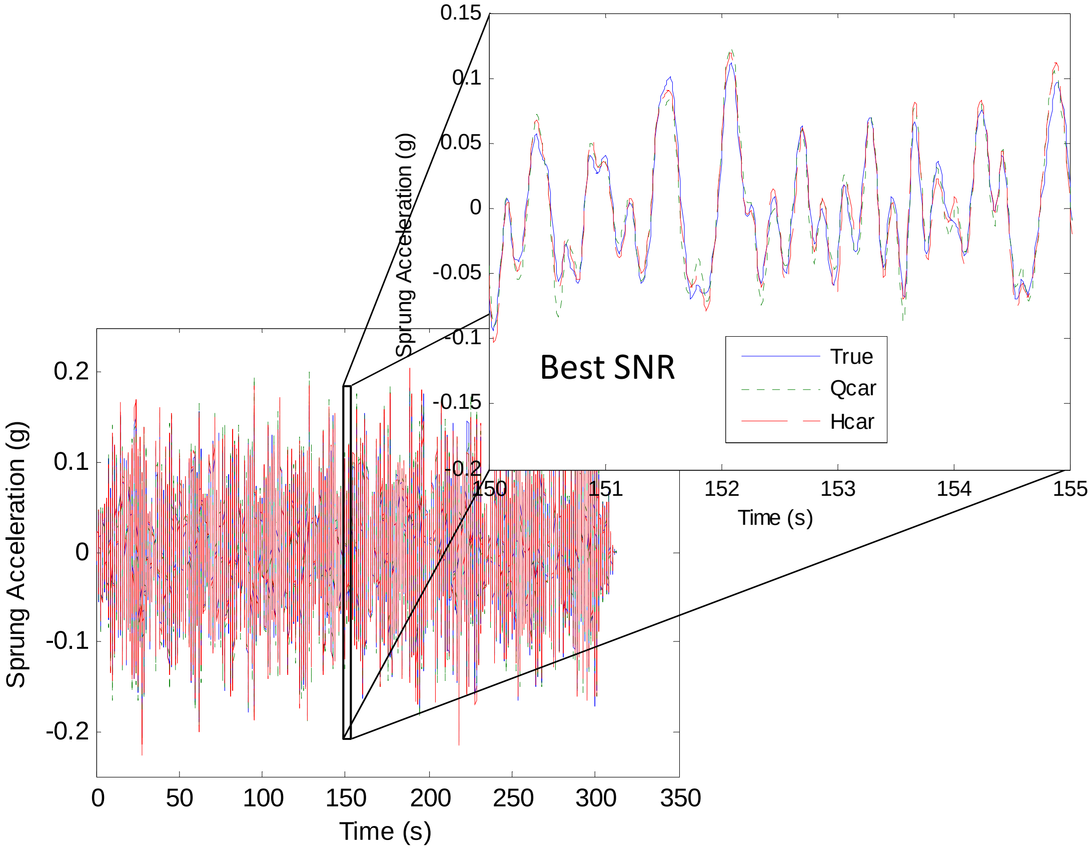


Figure 6.7: Time trajectories of the half-car (Hcar) and reduced order full-car (Qcar) base excitation models compared with the output of the actual vehicle (True) for the best SNR case, Road No. 1.

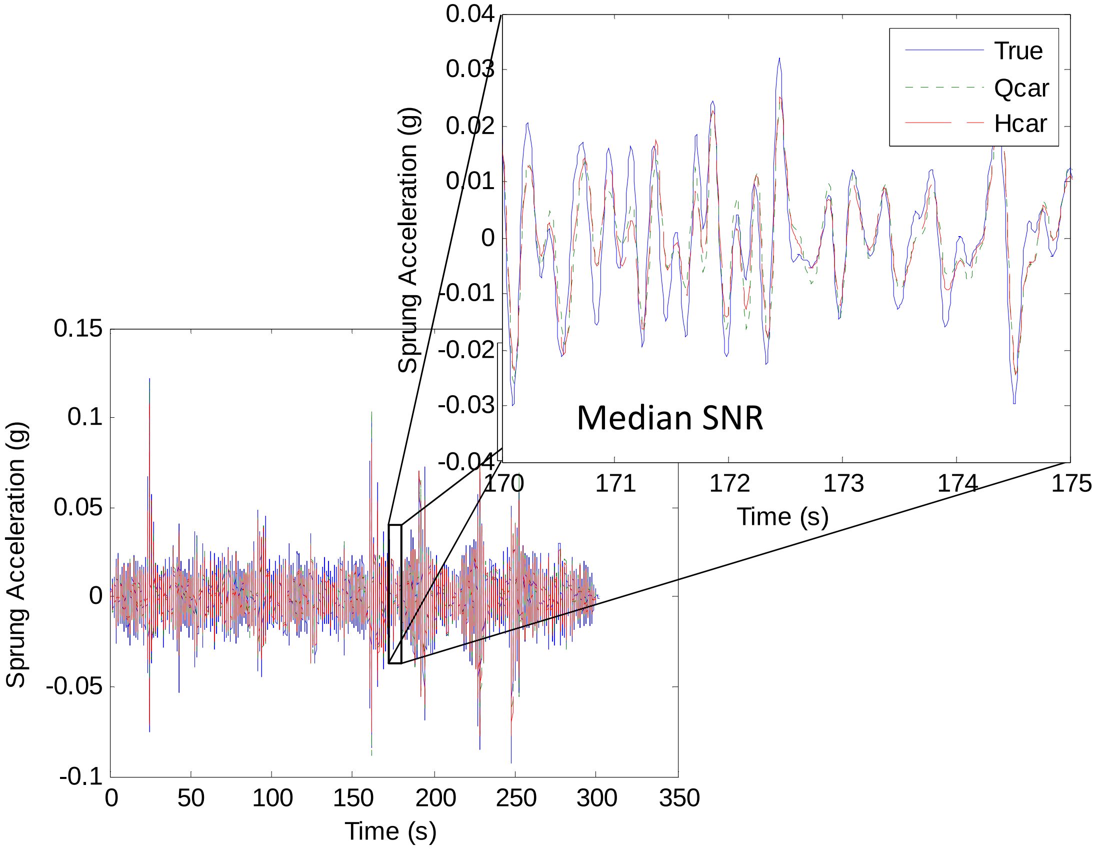


Figure 6.8: Time trajectories of the half-car (Hcar) and reduced order full-car (Qcar) base excitation models compared with the output of the actual vehicle (True) for Road No. 21.

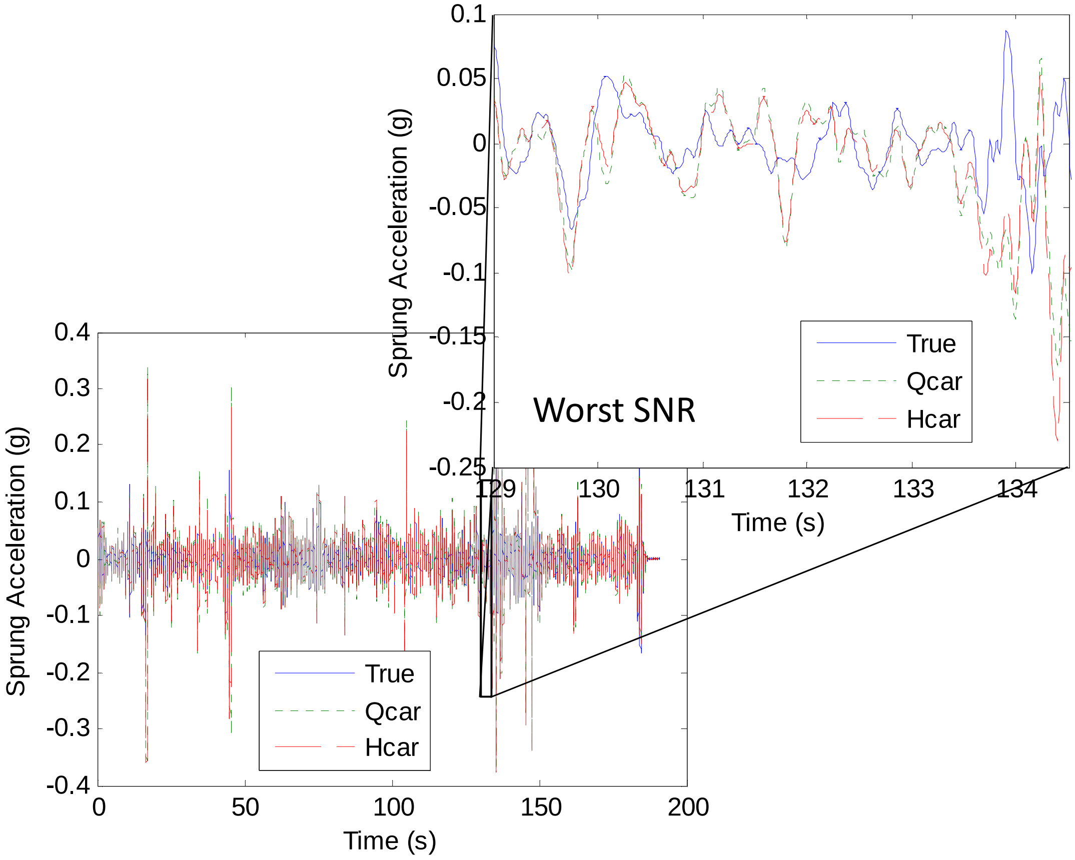


Figure 6.9: Time trajectories of the half-car (Hcar) and reduced order full-car (Qcar) base excitation models compared with the output of the actual vehicle (True) for the worst SNR case, Road No. 40.

Table 6.2: Validation results for the Reduced Full-car and Half-car models.

Road No.	Description	Target Speed (mph)	Name	Input RMS (g)	Output RMS (g)	SNR Reduced Full-car	SNR Half-car
1	Rough Dirt	10	Light10	0.0691	0.0541	**23.6114**	**41.3609**
2	Rough Dirt	5	Light5	0.0465	0.0355	**18.4218**	**33.1298**
3	Rough Dirt	10	PL10	0.071	0.0581	**16.9782**	**35.7559**
4	Rough Dirt	15	Light15	0.0864	0.0678	**15.0068**	**30.8806**
5	Rough Dirt	3	Light2	0.0323	0.0256	**13.7319**	**24.3111**
6	Rough Dirt	5	PL5	0.047	0.0373	**12.6351**	**29.2153**
7	Rough Dirt	5	NCparking	0.0329	0.0258	**12.1297**	**22.5208**
8	Rough Dirt	15	PL15	0.0895	0.0664	**12.0064**	**25.5087**
9	Rough Dirt	3	PL2	0.0314	0.0253	**10.1264**	**20.1533**
10	Rough Dirt	10	hvnew10	0.0255	0.0157	**8.679**	**12.5054**
11	Paved	30	LoopValid	0.0362	0.0231	**8.0943**	**8.2712**
12	Smooth Dirt	15	ltlll15	0.0209	0.0158	**7.2456**	**7.6782**
13	Paved	30	NearGR	0.0178	0.0114	**6.7477**	**7.4152**
14	Smooth Dirt	15	hvlll15	0.0204	0.0148	**6.2971**	**6.6279**
15	Smooth Dirt	10	ltlll10	0.0191	0.014	**5.7535**	**6.535**
16	Paved	70	Free696	0.0396	0.0335	**5.7086**	**6.0526**
17	Rough Dirt	5	hvnew5	0.0173	0.0133	**5.6219**	**7.6379**
18	Smooth Dirt	10	hvjoy10	0.0188	0.0121	**5.0007**	**6.9882**
19	Paved	30	NeGR3800	0.0367	0.0223	**4.5858**	**6.8605**
20	Rough Paved	30	LtFullGas	0.047	0.0252	**4.2848**	**4.6251**

Road No.	**Description**	**Target Speed (mph)**	**Name**	**Input RMS**	**Output RMS**	**SNR Reduced Full-car**	**SNR Half-car**
21	Smooth Dirt	10	hvlll10	0.0174	0.0125	**4.24**	**4.9453**
22	Paved	35	OW2Brick	0.0374	0.0225	**4.2164**	**5.6049**
23	Smooth Dirt	20	ltlll20	0.0233	0.0156	**3.7439**	**4.2131**
24	Paved	70	FreeNew	0.0393	0.037	**3.4469**	**3.3975**
25	Paved	70	I23ply2sal	0.0317	0.0282	**3.1145**	**3.6073**
26	Paved/Dirt	50	PontWar	0.0336	0.024	**2.9586**	**3.859**
27	Rough Dirt	35	CFB2F	0.0164	0.0084	**2.5737**	**2.8026**
28	Rough Dirt	15	hvnew15	0.0274	0.0162	**2.5283**	**3.2742**
29	Rough Paved	30	GR3800	0.0527	0.0252	**2.439**	**3.4255**
30	Smooth Dirt	25	ltlll25	0.0263	0.0159	**2.109**	**2.4989**
31	Smooth Dirt	20	hvlll20	0.0246	0.0164	**1.7685**	**2.0715**
32	Smooth Dirt	25	FLkRdNrth	0.0318	0.0144	**1.1677**	**1.462**
33	Paved/Dirt	50	InSaline	0.0402	0.0255	**1.1436**	**1.2654**
34	Rough Dirt	20	hvnew20	0.0313	0.0154	**1.1291**	**1.4217**
35	Rough Dirt	25	hvnew25	0.0359	0.0184	**1.0088**	**1.2172**
36	Smooth Dirt	25	hvlll25	0.0303	0.0175	**0.9013**	**1.1522**
37	Rough Dirt	30	hvnew30	0.0364	0.0186	**0.6347**	**0.7351**
38	Rough Dirt	20	N2BonC20	0.0343	0.0154	**0.5571**	**0.6966**
39	Rough Dirt	30	B2N30	0.0212	0.0081	**0.394**	**0.4695**
40	Rough Dirt	45	N2BonC45	0.0525	0.0259	**0.2696**	**0.2957**

6.4.1 Discussion of Results

The experiments in the top quarter of Table 6.2, corresponding to the best signal-to-noise ratios, had vehicle speeds in the range of 3-15 miles per hour (mph) (5-25 kilometers per hour) and were on rough dirt terrain. The vehicle speeds in the bottom quarter of the table were above 20 miles per hour (30 kilometers per hour) and were mostly on smooth or rough dirt terrains. The experiments in the middle of the table were mostly on smooth dirt or paved terrains with a wide range of speeds.

The results also show that, in almost every case, the higher-order half-car model matches the true vehicle better (in the SNR sense) than the reduced order full car model. The effect is more significant in the top quarter of Table 6.2 than in the rest of the table. From Road No. 11 to Road No. 40, the SNR for the reduced full-car and half-car models are fairly comparable. This suggests that the (lower computationally expensive) reduced full-car model may be sufficient for the vehicle mass estimators, assuming that the SNR is a good indication of the performance of the mass estimators. This last assumption will be verified in Section 8.2. In that case, the potential improvement due to the half-car model may not be worth the extra computational cost.

If SNR is a good indicator for estimation success (as will be explored and verified in Section 8.2), the mass estimation algorithm could benefit from knowledge of the SNR. For example, a supervisory control strategy could switch the estimator on or off depending on the current value of the SNR. Unfortunately, the SNR requires knowledge of the system parameters, some of which are unknown. Fortunately, as will be shown in Section 7.1.3, polynomial chaos theory enables recursive estimation of the SNR.

6.5 Summary

This chapter derived base-excitation models of vehicle ride dynamics from first principles. The base excitation models are valuable to the sprung mass estimation problem because they enable estimation without knowledge of the terrain profile and without requiring the vehicle to be equipped with an active/semi-active suspension.

This chapter experimentally validated two of the proposed base excitation models against the actual vehicle for 40 different experimental conditions. The two models validated were the base-excitation half-car model and the reduced order base excitation model of the full-car vertical dynamics. The validation showed that the more computationally expensive model, the half-car model, performed better than the reduced-order full-car model. However, the performance of both models was comparable for the majority of the experiments, suggesting that in most cases, the marginal improvement of the half-car model may not be worth the extra computational expense.

In the next chapter, base excitation models are combined with estimation techniques to derive algorithms for estimating the vehicle sprung mass.

Chapter 7

Vehicle Sprung Mass Estimators

This chapter combines base excitation models from Chapter 6 with the polynomial chaos estimators of this dissertation and other benchmark estimators. The benchmark estimators include Recursive Least Squares (RLS) (see pages 192-245 of (Ioannou and Sun, 1996)), Recursive Total Least Squares (RTLS) (see for example (Kubus et al., 2008)), Extended Kalman Filter (EKF) (Simon, 2006), and Unscented Kalman Filter (UKF) (Simon, 2006).

Section 7.1 formulates the sprung mass estimator based on the polynomial chaos techniques of Chapters 4 and 5. Sections 7.2 and 7.3 formulate mass estimators based on the least squares algorithms and filtering algorithms respectively.

7.1 Polynomial Chaos and Maximum Likelihood Approach

Chapter 4 developed a recursive maximum likelihood estimator for state-space systems using polynomial chaos theory. This estimator builds on the polynomial chaos technique for modeling multibody dynamic systems with uncertainties developed by Sandu, Sandu, and Ahmadian (Sandu et al., 2006) and extends the batch maximum *a posteriori* estimator developed by Blanchard, Sandu, and Sandu (Blanchard et al., 2009) to applications that require recursive estimation. Existing research has explored the use of recursive polynomial chaos based algorithms for vehicle mass estimation ((Pence et al.,

2010) and (Shimp, 2008)). The study presented here, however, uses an algorithm that estimates parameters based on an *integrated* cost function which is different from the previous approaches which used an algorithm that updates parameter estimates based on an *instantaneous* cost function. The earlier work by Pence, Fathy, and Stein (Pence et al., 2010) showed that estimators based on the *instantaneous* cost criteria resulted in estimates having significant variance.

7.1.1 Polynomial Chaos Estimator for the Quarter-car/Reduced Order Full Car Model

The polynomial chaos approach is applied directly to the reduced order base excitation equations (recall Equations (6.25) – (6.26)):

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \dfrac{-2(k_f + k_r)}{m} & \dfrac{-B_z}{m} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ -1 \end{bmatrix} u \tag{7.1}$$

$$y = \begin{bmatrix} \dfrac{-2(k_f + k_r)}{m} & \dfrac{-B_z}{m} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}. \tag{7.2}$$

Because these equations are linear, this chapter uses the Galerkin approach described in Chapter 4. To apply the polynomial chaos based algorithms, the unknown parameters are treated as random variables. The polynomial chaos algorithm based on Legendre polynomials requires upper and lower bounds on the unknown parameters which can be inferred from prior knowledge. This paper assumes that the true values of the unknown parameters could potentially be any values between the lower and upper bounds with equal probability, *i.e.*, it assumes that the prior distributions are uniform. With known upper and lower bounds, the random variables representing the unknown mass and damping terms can be written as functions of independently identically distributed (IID) polynomial chaos variables ξ_m and ξ_B, each of which are uniformly distributed over the interval $[-1, 1]$:

$$\begin{aligned} m &= \mu_m + v_m \xi_m \\ B_z &= \mu_B + v_B \xi_B. \end{aligned} \tag{7.3}$$

Here, μ_m and μ_B are respectively the known mean values of the mass m and damping B_z random variables. The terms v_m and v_B are the known maximum variation possible of m and B_z respectively from the mean values μ_m and μ_B; *e.g.*, $v_m = m_{upperLimit} - \mu_m$ where $m_{upperLimit}$ is the largest possible numerical value of the mass m. The goal of polynomial chaos based estimation is to estimate the most likely realizations of ξ_m and ξ_B and hence, by (7.3), the most likely values of the unknown mass m and damping term B_z.

Applying the Galerkin method of polynomial chaos theory to the system of Equations (7.1) – (7.2) (with the unknown parameters m and B_z replaced by their polynomial chaos counterparts given in (7.3)) results in the following set of deterministic state equations:

$$\dot{X} = \begin{bmatrix} \mathbf{0}_{r\times r} & \boldsymbol{I}_{r\times r} \\ A_{21} & A_{22} \end{bmatrix} X + \begin{bmatrix} \mathbf{0}_r \\ -1 \\ \mathbf{0}_{r-1} \end{bmatrix} u. \tag{7.4}$$

Here, $\mathbf{0}_k$ is a matrix/vector of zeros having dimension $k \in \{r \times r, r, r-1\}$, the identity matrix $\boldsymbol{I}_{r\times r}$ has dimensions $r \times r$, and $r = (S^2 + 3S + 2)/2$ where S is the user-selected maximum polynomial chaos order, ($S = 6$ in the experiments of this Chapter). The state vector $X \in \mathbb{R}^{2r}$ is a vector of polynomial chaos expansion coefficients. Finally, the matrices $A_{21} \in \mathbb{R}^{r\times r}$ and $A_{22} \in \mathbb{R}^{r\times r}$ are defined as follows:

$$A_{21} := A_{diag}\begin{bmatrix}\langle\Phi_1, Q\Phi_1\rangle & \langle\Phi_1, Q\Phi_2\rangle & \cdots & \langle\Phi_1, Q\Phi_r\rangle\\ \langle\Phi_2, Q\Phi_1\rangle & \langle\Phi_2, Q\Phi_2\rangle & & \langle\Phi_2, Q\Phi_r\rangle\\ \vdots & & \ddots & \vdots\\ \langle\Phi_r, Q\Phi_1\rangle & \langle\Phi_r, Q\Phi_2\rangle & \cdots & \langle\Phi_r, Q\Phi_r\rangle\end{bmatrix}, Q = \frac{-2(k_f + k_r)}{\mu_m + v_m\xi_m} \tag{7.5}$$

$$A_{22} := A_{diag}\begin{bmatrix}\langle\Phi_1, G\Phi_1\rangle & \langle\Phi_1, G\Phi_2\rangle & \cdots & \langle\Phi_1, G\Phi_r\rangle\\ \langle\Phi_2, G\Phi_1\rangle & \langle\Phi_2, G\Phi_2\rangle & & \langle\Phi_2, G\Phi_r\rangle\\ \vdots & & \ddots & \vdots\\ \langle\Phi_r, G\Phi_1\rangle & \langle\Phi_r, G\Phi_2\rangle & \cdots & \langle\Phi_r, G\Phi_r\rangle\end{bmatrix}, G = \frac{-(\mu_B + v_B\xi_B)}{\mu_m + v_m\xi_m} \tag{7.6}$$

$$A_{diag} := \begin{bmatrix}\langle\Phi_1, \Phi_1\rangle^{-1} & 0 & \cdots & 0\\ 0 & \langle\Phi_2, \Phi_2\rangle^{-1} & & 0\\ \vdots & & \ddots & \vdots\\ 0 & 0 & \cdots & \langle\Phi_r, \Phi_r\rangle^{-1}\end{bmatrix}. \tag{7.7}$$

The polynomial chaos based estimator is asymptotically stable if the eigenvalues of the state transition matrix in Equation (7.4) have negative real parts. For the base excitation model, stability is guaranteed when the lower bounds on the parameter prior distributions are greater than zero. The multivariate polynomials $\Phi_k, k = 1,2, \dots, r$ are functions of the Legendre polynomials (Poularikas, 1999) $\phi_i(\xi_m), i = 0,1, \dots, S$ and $\phi_j(\xi_B), j = 0,1, \dots, S$ as follows:

$$\Phi_k = \phi_i(\xi_m)\,\phi_j(\xi_B), \qquad k = i\left(S + \frac{3-i}{2}\right) + j + 1, \qquad i = 0,1, \dots, S, \\ j = 0,1, \dots, S - i. \tag{7.8}$$

The inner products $\langle f(\xi_m, \xi_B), g(\xi_m, \xi_B)\rangle$ are defined as follows:

$$\langle f(\xi_m, \xi_B), g(\xi_m, \xi_B)\rangle := \int_{-1}^{1}\int_{-1}^{1} g(\xi_m, \xi_B) f(\xi_m, \xi_B) d\xi_m d\xi_B. \tag{7.9}$$

Since the integrals are evaluated over the event space of the random variables, the inner product $\langle f(\xi_m, \xi_B), g(\xi_m, \xi_B)\rangle$ is a known deterministic quantity; as a result, Equation (7.4) is a deterministic set of state equations that can be solved numerically to determine the value of the state vector X. Also, since the multivariate polynomials

$\Phi_k, k = 1,2, \dots, r$ are orthogonal with respect to the inner product in Equation (7.9), the matrix A_{diag} is well defined.

Following the concepts of polynomial chaos theory, the states x_1 and x_2 of Equation (7.1) are functions of the random variables ξ_m and ξ_B, and are related to the state vector X of Equation (7.4) through a polynomial chaos expansion approximation:

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \approx PX, \qquad P := \begin{bmatrix} \Phi_1 & \cdots & \Phi_r & & \mathbf{0}_{1\times r} & \\ & \mathbf{0}_{1\times r} & & \Phi_1 & \cdots & \Phi_r \end{bmatrix}. \tag{7.10}$$

This approximation becomes exact in the least squares sense as $S \to \infty$ (Ghanem and Spanos, 1991). Substituting Equations (7.10) and (7.3) into (6.26) results in the polynomial chaos approximation output $\hat{y}$ which is a function of the unknown variables ξ_m and ξ_B.

$$\hat{y} \approx \begin{bmatrix} \frac{-2(k_f + k_r)}{\mu_m + v_m \xi_m} & \frac{-(\mu_B + v_B \xi_B)}{\mu_m + v_m \xi_m} \end{bmatrix} PX \tag{7.11}$$

The output $\hat{y}$ is not an output *trajectory* but rather an output *process* or family of trajectories. From (7.11), one can see that given realizations of ξ_m and ξ_B, $\hat{y}$ collapses to an output trajectory. The goal of recursive maximum likelihood is to determine the realizations of ξ_m and ξ_B that make $\hat{y}$ most like the measured output trajectory y in some sense. This is done by selecting ξ_m and ξ_B so that they optimize the likelihood function (assuming additive zero mean Gaussian measurement noise with variance ${\sigma_k}^2$):

$$\mathcal{L}(\xi_m, \xi_B | y) \propto exp\left\{ -\sum_{k=0}^{N} \frac{\big(y(t_k) - \hat{y}(t_k, \xi_m, \xi_B)\big)^2}{2{\sigma_k}^2} \right\} \tag{7.12}$$

Chapter 4 developed techniques for recursively calculating estimates of the unknown parameters that recursively seek to optimize the likelihood function (7.12). Since the prior distributions of ξ_m and ξ_B are uniform, the MAP estimator of Chapter 5 reduces to the maximum likelihood estimator for $\xi_m \in [-1, 1]$ and $\xi_B \in [-1, 1]$. The key limitation of the polynomial chaos based methods of Chapters 4 and 5 is the fact that the polynomial chaos approximation is only exact in the limiting sense as $S \to \infty$ which is not numerically feasible. The author has found, as a rule of thumb, that $S \geq 6$ is a good approximation for this mass estimation problem. A general procedure to determine an acceptable value for S is to start with a small value for S and then increase the value until the change in the resulting estimates is acceptably small.

7.1.2 Polynomial Chaos Estimator for the Half-car Model

The half car base excitation model can be described by the following equations (recall Equations (6.5) and (6.6)):

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\left(\frac{a^2}{J_\phi} + \frac{1}{m}\right) k_f & -\left(\frac{a^2}{J_\phi} + \frac{1}{m}\right) b_f & \left(\frac{ab}{J_\phi} - \frac{1}{m}\right) k_r & \left(\frac{ab}{J_\phi} - \frac{1}{m}\right) b_r \\ 0 & 0 & 0 & 1 \\ \left(\frac{ab}{J_\phi} - \frac{1}{m}\right) k_f & \left(\frac{ab}{J_\phi} - \frac{1}{m}\right) b_f & -\left(\frac{b^2}{J_\phi} + \frac{1}{m}\right) k_r & -\left(\frac{b^2}{J_\phi} + \frac{1}{m}\right) b_r \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ -1 & 0 \\ 0 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} u_f \\ u_r \end{bmatrix} \tag{7.13}$$

$$y = \left[\left(\frac{ab^2 - a^3}{J_\phi L} - \frac{1}{m}\right) k_f \quad \left(\frac{ab^2 - a^3}{J_\phi L} - \frac{1}{m}\right) b_f \quad \left(\frac{a^2 b - b^3}{J_\phi L} - \frac{1}{m}\right) k_r \quad \left(\frac{a^2 b - b^3}{J_\phi L} - \frac{1}{m}\right) b_r \right] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}. \tag{7.14}$$

The unknown parameters of this model are assumed to be m, b_f, and b_r. The states are also unknown. (The term J_ϕ is probably unknown as well, but the pitch dynamics are not observable, and J_ϕ is not identifiable based on the input and output signals of (7.13) and (7.14). In the following development, its value is assumed to be known. In the experimental studies, large variations in the assumed value of J_ϕ had a negligible effect on the estimated values of m, b_f, and b_r.)

The development of the polynomial chaos half-car estimator follows the same steps as the development of the quarter-car model in the previous section. The estimator can be summarized by the following procedure:

1. Expand the unknown parameters in terms of the polynomial chaos variables:

$$\begin{aligned} m &= \mu_m + v_m \xi_m, & \xi_m &\sim U[-1,1] \\ b_f &= \mu_{b_f} + v_{b_f} \xi_{b_f}, & \xi_{b_f} &\sim U[-1,1] \\ b_r &= \mu_{b_r} + v_{b_r} \xi_{b_r}, & \xi_{b_r} &\sim U[-1,1]. \end{aligned} \tag{7.15}$$

2. Apply the Galerkin method to get a deterministic set of state equations:

$$\dot{X} = \begin{bmatrix} \mathbf{0}_{r\times r} & \boldsymbol{I}_{r\times r} & \mathbf{0}_{r\times r} & \mathbf{0}_{r\times r} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ \mathbf{0}_{r\times r} & \mathbf{0}_{r\times r} & \mathbf{0}_{r\times r} & \boldsymbol{I}_{r\times r} \\ A_{41} & A_{42} & A_{43} & A_{44} \end{bmatrix} X + \begin{bmatrix} \mathbf{0}_{r\times 2} \\ [-1 \quad 0] \\ \mathbf{0}_{r-1\times 2} \\ \mathbf{0}_{r\times 2} \\ [0 \quad -1] \\ \mathbf{0}_{r-1\times 2} \end{bmatrix} \begin{bmatrix} u_f \\ u_r \end{bmatrix}. \tag{7.16}$$

The components A_{ij} are defined as follows:

$$A_{ij} \coloneqq A_{diag} \begin{bmatrix} \langle \Phi_1, Q_{ij}\Phi_1 \rangle & \langle \Phi_1, Q_{ij}\Phi_2 \rangle & \cdots & \langle \Phi_1, Q_{ij}\Phi_r \rangle \\ \langle \Phi_2, Q_{ij}\Phi_1 \rangle & \langle \Phi_2, Q_{ij}\Phi_2 \rangle & & \langle \Phi_2, Q_{ij}\Phi_r \rangle \\ \vdots & & \ddots & \vdots \\ \langle \Phi_r, Q_{ij}\Phi_1 \rangle & \langle \Phi_r, Q_{ij}\Phi_2 \rangle & \cdots & \langle \Phi_r, Q_{ij}\Phi_r \rangle \end{bmatrix}, \tag{7.17}$$

$$Q_{21} = -\left(\frac{a^2}{J_\phi} + \frac{1}{\mu_m + v_m\xi_m}\right) k_f,$$

$$Q_{22} = -\left(\frac{a^2}{J_\phi} + \frac{1}{\mu_m + v_m\xi_m}\right) \left(\mu_{b_f} + v_{b_f}\xi_{b_f}\right),$$

$$Q_{23} = \left(\frac{ab}{J_\phi} - \frac{1}{\mu_m + v_m\xi_m}\right) k_r,$$

$$Q_{24} = \left(\frac{ab}{J_\phi} - \frac{1}{\mu_m + v_m\xi_m}\right)\left(\mu_{b_r} + v_{b_r}\xi_{b_r}\right),$$

$$Q_{41} = \left(\frac{ab}{J_\phi} - \frac{1}{\mu_m + v_m\xi_m}\right)k_f,$$

$$Q_{42} = \left(\frac{ab}{J_\phi} - \frac{1}{\mu_m + v_m\xi_m}\right)\left(\mu_{b_f} + v_{b_f}\xi_{b_f}\right),$$

$$Q_{43} = -\left(\frac{b^2}{J_\phi} + \frac{1}{\mu_m + v_m\xi_m}\right)k_r,$$

$$Q_{44} = -\left(\frac{b^2}{J_\phi} + \frac{1}{\mu_m + v_m\xi_m}\right)\left(\mu_{b_r} + v_{b_r}\xi_{b_r}\right),$$

$$A_{diag} \coloneqq \begin{bmatrix} \langle\Phi_1, \Phi_1\rangle^{-1} & 0 & \cdots & 0 \\ 0 & \langle\Phi_2, \Phi_2\rangle^{-1} & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & \langle\Phi_r, \Phi_r\rangle^{-1} \end{bmatrix}. \tag{7.18}$$

The inner product and polynomial chaos terms Φ_i are also defined differently from the case with only two unknown parameters:

$$\begin{gathered} \Phi_k \coloneqq \phi_i(\xi_m)\,\phi_j\left(\xi_{b_f}\right)\,\phi_l\left(\xi_{b_f}\right), \;\; i = 0,1,\dots,S, \qquad j = 0,1,\dots,S-i, \\ l = 0,1,\dots,S-i-j, \qquad k = 1,\dots,r. \end{gathered} \tag{7.19}$$

$$\langle f(\xi_m, \xi_{b_f}, \xi_{b_r}), g(\xi_m, \xi_{b_f}, \xi_{b_r})\rangle \coloneqq \int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1}(f\cdot g)\,d\xi_m d\xi_{b_f} d\xi_{b_r}. \tag{7.20}$$

3. Determine the polynomial chaos output $\hat{y}_k$ as a function of the unknown polynomial chaos parameters ξ_m, ξ_{b_f}, and ξ_{b_r}:

$$\hat{y}_k \approx [C_1 \quad C_2 \quad C_3 \quad C_4]\mathbb{P}X(t_k) \tag{7.21}$$

$$C_1 \coloneqq \left(\frac{ab^2 - a^3}{J_\phi L} - \frac{1}{\mu_m + v_m\xi_m}\right)k_f,$$

$$C_2 := \left(\frac{ab^2 - a^3}{J_\phi L} - \frac{1}{\mu_m + v_m \xi_m}\right)\left(\mu_{b_f} + v_{b_f}\xi_{b_f}\right)$$

$$C_3 := \left(\frac{a^2 b - b^3}{J_\phi L} - \frac{1}{\mu_m + v_m \xi_m}\right) k_r$$

$$C_4 := \left(\frac{a^2 b - b^3}{J_\phi L} - \frac{1}{\mu_m + v_m \xi_m}\right)\left(\mu_{b_r} + v_{b_r}\xi_{b_r}\right)$$

The matrix $\mathbb{P}$ and vector X follow from Chapter 4, Equations (4.15) and (4.16).

4. Calculate estimates of ξ_m, ξ_{b_f}, and ξ_{b_r} using the methods of Chapter 4, Section 4.2, and/or Chapter 5, Section 5.1.
5. Substitute the estimates of ξ_m, ξ_{b_f}, and ξ_{b_r} into Equation (7.15) to get the estimates of m, b_f, and b_r.

7.1.3 Polynomial Chaos Estimator for the Signal-to-Noise Ratio

As discussed in Section 6.4, the Signal-to-Noise Ratio (SNR) can potentially be used as a tool for analyzing how well the model outputs match the actual vehicle outputs. If an estimate of the SNR is available, it may be used to improve the robustness of the estimation algorithm (see Section 8.2). Unlike the filtering and regressor model based estimators which will be discussed later, polynomial chaos theory can be used to calculate an estimate of the SNR. Using polynomial chaos theory, the estimated SNR can be calculated iteratively as follows assuming that y_k is a scalar:

$$\widehat{SNR}_k = \frac{M_k^y}{\widehat{M}_k^{y-\hat{y}}},$$

$$M_k^y := M_{k-1}^y + y_k^2, \qquad M_0^y = 0, \tag{7.22}$$

$$\widehat{M}_k^{y-\hat{y}} := M_k^y + C(\hat{\xi}_k)P(\hat{\xi}_k)M_k^{XX^T}\left(C(\hat{\xi}_k)P(\hat{\xi}_k)\right)^T - 2C(\hat{\xi}_k)P(\hat{\xi}_k)M_k^{Xy},$$

$$M_k^{XX^T} := M_{k-1}^{XX^T} + X(t_k)X^T(t_k), \qquad M_0^{XX^T} = 0,$$

$$M_k^{XX^T} := M_{k-1}^{Xy} + X(t_k)y_k, \qquad M_0^{Xy} = 0.$$

Here, $X(t) \in \mathbb{R}^{r \cdot n_s}$ is the polynomial chaos system's state vector, and for the reduced full-car model, it is calculated using Equation (7.4). For the reduced order full-car model, $P(\hat{\xi}_k)$ is defined in Equation (7.10) and $C(\hat{\xi}_k)$ follows from Equation (7.11). The term $\hat{\xi}_k$ is the current estimate calculated by the polynomial chaos estimators of Chapters 4 and/or 5.

7.2 Regressor Model Based Estimation Methods

An alternative approach to using polynomial chaos based estimation is to use regressor-model based algorithms such as recursive least squares (RLS) (Ioannou and Sun, 1996) or recursive total least squares (RTLS) (Kubus et al., 2008). Under certain Gaussian measurement noise assumptions, and if the states x_1 and x_2 of Equations (7.1) – (7.2) are measured explicitly, RLS methods can potentially produce unbiased estimates. Unfortunately, full-state measurements require measuring suspension displacements and velocities at each corner of the vehicle, as well as the sprung mass acceleration. This dissertation follows Fathy, Kang, and Stein's (Fathy et al., 2008) method in using pre-filtering as a precursor to mass estimation. Such pre-filtering has two attractive advantages, namely, (a) it allows the estimation process to focus on those frequencies where inertial dynamics are more visible, and (b) it makes it possible to estimate sprung mass using sprung and unsprung mass accelerations, without any need to use additional sensors for displacement and velocity. This section outlines the regressor approach to vehicle sprung mass estimation for the reduced full-car model.

RLS and RTLS algorithms rely on the regressor model shown in Equation (7.23) and cannot be applied directly to state-space systems such as the system modeled by Equations (7.1) – (7.2).

$$y_R = {\phi_R}^T \theta_R \tag{7.23}$$

The term y_R is the measured *regressor output*, the term ϕ_R is the *regressor vector* which is also known/measured, and the term θ_R is the *unknown parameter vector*. For vehicle mass estimation, the regressor model can be derived from the output equation, Equation (7.2). Then, the regressor output y_R is $-2(k_f + k_r)x_1$, the regressor vector ϕ_R is $[y \quad x_2]^T$, and the unknown parameter vector θ_R is$[m \quad B_z]^T$, *i.e.*,

$$-2(k_f + k_r)x_1 = [y \quad x_2]\begin{bmatrix} m \\ B_z \end{bmatrix}. \tag{7.24}$$

The regressor output y_R and regressor vector ϕ_R must only contain known/measured variables, but the states x_1 and x_2 are not measured. To address this problem, this paper applies Laplace-domain filtering to obtain a regressor model in which both y_R and ϕ_R only contain known/measured values. These Laplace domain filters act as pseudo-integrators to obtain estimates of velocity and displacement from measurements of suspension acceleration. Assuming zero initial conditions for the states x_1 and x_2, the system of equations given in (7.1) – (7.2) can be represented in the Laplace-domain by the following transfer function:

$$\frac{Y(s)}{U(s)} = \frac{B_z s + 2(k_f + k_r)}{m s^2 + B_z s + 2(k_f + k_r)}. \tag{7.25}$$

In (7.25), s is the Laplace operator, $Y(s)$ is the Laplace transform of the sprung mass accelerations y, and $U(s)$ is the Laplace transform of the input u. Alternatively, (7.25) can be written as follows:

$$-2(k_f + k_r)(Y(s) - U(s)) = ms^2Y(s) + B_z s(Y(s) - U(s)). \tag{7.26}$$

Dividing both sides of (7.26) by $\Lambda(s)$, results in the following filtered regressor model:

$$-2(k_f + k_r)\frac{(Y(s) - U(s))}{\Lambda(s)} = \begin{bmatrix} \frac{s^2Y(s)}{\Lambda(s)} & \frac{s(Y(s) - U(s))}{\Lambda(s)} \end{bmatrix} \begin{bmatrix} m \\ B_z \end{bmatrix}. \tag{7.27}$$

The user-selected denominator $\Lambda(s)$ should be a polynomial function of s that is at least second order to ensure that the transfer functions in (7.27) are proper. The roots of the polynomial $\Lambda(s)$ must have negative real parts to guarantee stability. Reverting back to the time domain, the regressor model shown in Equation (7.27) becomes:

$$-2(k_f + k_r)\left\{\frac{1}{\Lambda(p)}\right\}(y - u) = \begin{bmatrix} \left\{\frac{p^2}{\Lambda(p)}\right\}y & \left\{\frac{p}{\Lambda(p)}\right\}(y - u) \end{bmatrix} \begin{bmatrix} m \\ B_z \end{bmatrix}. \tag{7.28}$$

The term $\{1/\Lambda(p)\}(y - u)$ represents the time domain signal $y - u$ filtered by the Laplace domain transfer function $\{1/\Lambda(s)\}$. The use of the operator p instead of s is to distinguish between the time and Laplace domains.

The regressor output $y_R = -2(k_f + k_r)\{1/\Lambda(p)\}(y - u)$ and regressor vector $\phi_R = [\{p^2/\Lambda(p)\}y \quad \{p/\Lambda(p)\}(y - u)]^T$ of Equation (7.28) contain only known/measured variables. The only unknown variables in (7.28) are the elements of the parameter vector $\theta_R = [m \quad B_z]^T$. Therefore, Equation (7.28) is a valid for regressor model based algorithms. However, two significant problems arise when using Equation (7.28). First, both the regressor output y_R and regressor vector ϕ_R contain measurements of both y and u. As a result, the noise in the output y_R and regressor vector ϕ_R is correlated. This leads to biased estimates when using recursive least squares algorithms. This approach was explored in a simulation study by Pence, Fathy, and Stein (Pence et al., 2009). Total least squares regression can potentially lead to less biased estimates (see

Section 7.7 of (Moon and Stirling, 2000)). However, a second problem with using Equation (7.28) affects both least squares and total least squares methods: these methods are both sensitive to the user-selected denominator $\Lambda(s)$. As the experimental validation section will show, slightly different tunings of $\Lambda(s)$ lead to significantly different estimates of the unknown parameters, and tunings that are appropriate for one terrain are not necessarily appropriate for another.

7.3 Filtering Methods

Filtering algorithms (see Appendix A2), such as the Extended Kalman Filter (EKF), Unscented Kalman Filter (UKF) (Simon, 2006), particle filter (Ristic et al., 2004), etc., can estimate both the states and unknown parameters of state space systems. Unlike the estimation methods discussed in the paragraphs above which require a regressor model, the filtering algorithms can be applied to state space systems such as the base excitation system of Equations (7.1) – (7.2) and Equations (7.13) – (7.14). First, however, the state vector must be augmented to include estimates of the unknown parameters.

7.3.1 Augmented Equations for the Quarter-car/Reduced Order Full Car Model

The augmented set of state and output equations for the base excitation system of Equations (7.1) – (7.2) is given as follows:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} x_2 \\ -\dfrac{2\left(k_f + k_r\right)}{x_3} x_1 - \dfrac{x_4}{x_3} x_2 - u \\ 0 \\ 0 \end{bmatrix} \tag{7.29}$$

$$y = -\frac{2(k_f + k_r)}{x_3}x_1 - \frac{x_4}{x_3}x_2. \tag{7.30}$$

The state x_3 is an estimate of the unknown value of the vehicle sprung mass m, and the state x_4 estimates the unknown damping term B_z. The filtering methods summarized in Appendix A2 can be applied to this augmented system of equations.

The EKF algorithm requires Jacobain matrices, *i.e.*, matrices of partial derivatives. The Jacobian matrices for Equations (7.29) – (7.30) are provided in Appendix A2, Equations (A2.6) and (A2.7).

The augmented system, Equations (7.29) – (7.30), has a higher dimension and is significantly more nonlinear than the (linear) system of Equations (7.1) – (7.2). Similar to the regressor model approaches, the filtering methods may be more difficult to tune than the proposed polynomial chaos approach. The filtering methods have a large number of user-defined tuning variables: the process noise covariance matrix, the measurement noise variance, the initial conditions for the estimate covariance matrix, and the initial conditions for the state vector (a total of at least 25 scalar tuning parameters for the mass estimation problem). Despite being difficult to tune, the EKF is especially attractive because of its low computational demand compared with the polynomial chaos approach, the UKF approach, and particle filter. Appendix A2 summarizes the filtering algorithms used in this dissertation and offers insights on how to tune the filters so that they calculate estimates that are similar to those calculated by the polynomial chaos estimators of this dissertation.

7.3.2 Augmented Equations for the Half-car Model

The augmented set of state equations for the half car model described by Equations (7.13) and (7.14) is as follows:

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\\ \dot{x}_4\\ \dot{x}_5\\ \dot{x}_6\\ \dot{x}_7\end{bmatrix}$$

$$= \begin{bmatrix} x_2 \\ -\left(\frac{a^2}{J_\phi}+\frac{1}{x_5}\right)k_f x_1 - \left(\frac{a^2}{J_\phi}+\frac{1}{x_5}\right)x_6 x_2 + \left(\frac{ab}{J_\phi}-\frac{1}{x_5}\right)k_r x_3 + \left(\frac{ab}{J_\phi}-\frac{1}{x_5}\right)x_7 x_4 - u_f \\ x_4 \\ \left(\frac{ab}{J_\phi}-\frac{1}{x_5}\right)k_f x_1 + \left(\frac{ab}{J_\phi}-\frac{1}{x_5}\right)x_6 x_2 - \left(\frac{b^2}{J_\phi}+\frac{1}{x_5}\right)k_r x_3 - \left(\frac{b^2}{J_\phi}+\frac{1}{x_5}\right)x_7 x_4 - u_r \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{7.31}$$

$$y = \left(\frac{ab^2-a^3}{J_\phi L}-\frac{1}{x_5}\right)k_f x_1 + \left(\frac{ab^2-a^3}{J_\phi L}-\frac{1}{x_5}\right)x_6 x_2 + \left(\frac{a^2 b-b^3}{J_\phi L}-\frac{1}{x_5}\right)k_r x_3 + \left(\frac{a^2 b-b^3}{J_\phi L}-\frac{1}{x_5}\right)x_7 x_4. \tag{7.32}$$

The state x_5 is the sprung mass m, x_6 is the front suspension damping coefficient b_f, and x_7 is the rear suspension damping coefficient b_r.

Filtering algorithms such as the EKF, UKF and particle filter can be applied directly to the augmented state equations of (7.31) and (7.32). The Jacobain matrices that the EKF requires are provided in Appendix A2.

Chapter 8

Experimental Validation of Vehicle Mass Estimators

This chapter describes the experimental setup, procedures, and results for estimating the mass of an actual vehicle traversing a variety of terrains. This chapter has two purposes, first to asses the validity of using base excitation models with polynomial chaos estimators for accurate and robust vehicle mass estimation; Section 8.2 explores this objective. The second purpose of this chapter is to experimentally compare the performance of the polynomial chaos estimators of this dissertation with benchmark algorithms such as filtering methods and least squares methods. This comparison is done in Sections 8.3 and 8.4. Finally, Section 8.5 offers conclusions.

8.1 Experimental Setup

The vehicle used for the experiments of this dissertation was a 2001 Nissan Altima, GXE sedan. The experimental work used single-axis accelerometers, model 2210-005 made by Silicon Designs, Inc. Unsprung mass accelerometers were attached to the suspension struts below the spring (Figure 8.1) using J.B. Weld® epoxy. The sprung mass accelerometer was attached to the base of the cup holder between the front seats (Figure 8.2) using Duro® super-glue. The data acquisitioning system was a National Instruments® model NI USB-6221. Data samples were acquired at either a 500 Hz or 1 kHz rate, band-pass filtered, and then downsampled to a 100 Hz rate. The band-pass filter attenuated frequencies below 0.3 Hz to remove accelerometer bias and above 5 Hz to remove high frequency noise. The accelerometers were connected to the acquisitioning system in differential mode.

A Proform® 67650 Vehicle Scale System was used to measure the total vehicle weight (including the driver and acquisitioning equipment weight), which was found to be 1460 kg (3220 lbs). This Vehicle Scale System also determined the Center of Gravity (cg) location to be a distance of 62% of the wheelbase length (2.62 m) forward from the rear axle. The tires and wheels were removed from the vehicle and weighed; they were found to have a total mass of 70 kg (150 lbs). The remaining mass of the unsprung components (hub, suspension, axle, etc.) was estimated to be 94 kg (210 lbs) based on shipping weight calculations published on an internet website: Amazon.com^{TM}. These unsprung components with their associated shipping weight estimates are listed in Table 8.1. Based on the measurements of the tire and wheel weight and the estimated value of the remaining unsprung mass, the total sprung mass was determined to be 1296 kg (2860 lbs)[1].

[1] In some of the experiments of this paper, additional known mass was added to the sprung mass of the vehicle, either by adding passengers, equipment, and/or boxes filled with sand. The values shown in Table 6.2 and Table 8.2 automatically account for this added mass. For example, in Table 6.2, the SNR calculation for an experiment with added mass used a vehicle model that had the added mass included. The estimation errors reported in Table 8.2 are with respect to the total sprung mass (including the added mass). The value of the added mass for any experiment was less than 260 kg (575 lbs).

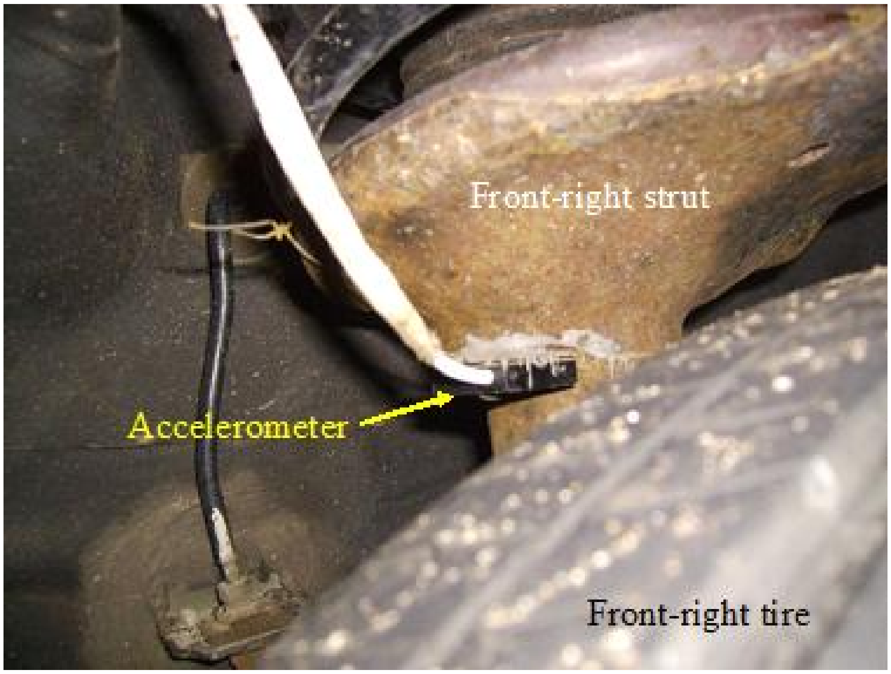


Figure 8.1: Placement of the unsprung accelerometer.

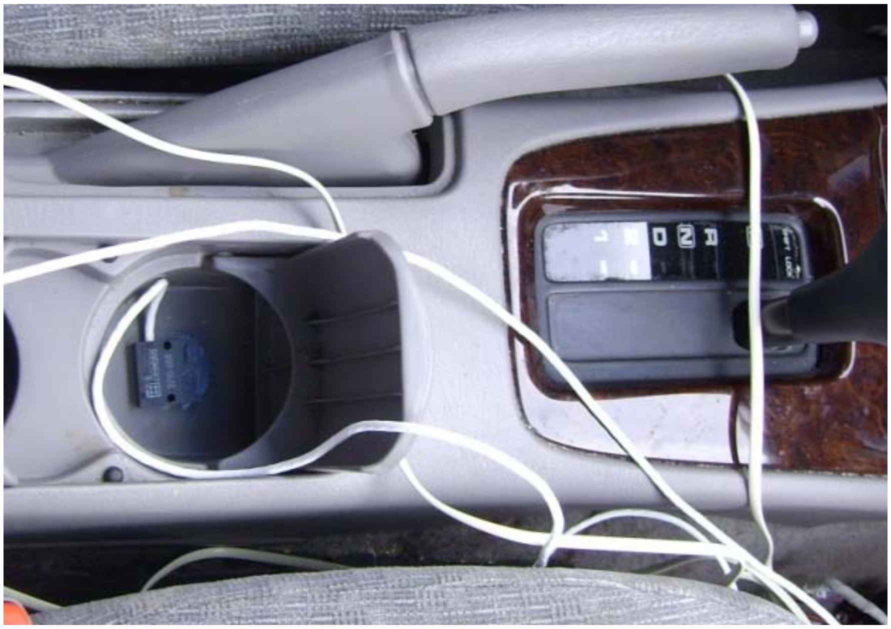


Figure 8.2: Location of sprung mass accelerometer.

Table 8.1: Mass of unsprung suspension components.

Unsprung Suspension Components and Corresponding Shipping Weights (kg)			
Front suspension (one corner)		**Rear suspension (one corner)**	
Hub Assembly	2.3 kg	Hub Assembly	2.4 kg
½ of Front Axle	4.2 kg	Knuckle	6.4 kg
Steering Knuckle	3.7 kg	Strut Assembly	4.8 kg
Strut Assembly	4.4 kg	Brake Drum	4.4 kg
Disc Brake Rotor	6.3 kg	Drum Shoes	1.0 kg
Brake Caliper and Pads	2.9 kg	Brake Hardware	0.8 kg
½ of Control Arm	2.2 kg	Wheel Cylinder	0.5 kg
		½ of Bottom Control Arm	0.6 kg
Total	**26 kg**	**Total**	**20.9 kg**

The stiffness values of the front and rear suspensions were determined respectively to be 41800 ± 3000 N/m and 31700 ± 3000 N/m per corner, *e.g.*, the front-right suspension stiffness was 41,800 N/m which was also the value of the front-left stiffness; the total front stiffness was 83,600 N/m and the total rear stiffness was 63,400 N/m. These values were determined by placing a known load of 53 kg on the sprung body at the front center and then rear center of the vehicle and measuring the respective suspension displacement at each corner of the vehicle using a ruler with millimeter precision. Then the experiment used a vertical static force balance (evaluating vertical force verses suspension displacement assuming a linear suspension) to determine the stiffness values of the suspensions.

8.2 Experimental Validation of Polynomial Chaos Estimators

This section explores the validity and robustness of the vehicle mass estimators that use the polynomial chaos algorithms and base excitation models developed in this dissertation. Section 7.1.3 provided an algorithm based on polynomial chaos theory that can be used to iteratively calculate an estimate of the signal-to-noise ratio (SNR). This section, Section 8.2, aims to address the question of whether the estimated SNR is a good indicator of estimator success. If so, robust estimation can be achieved by using supervisory logic that only provides mass estimates when the current estimated SNR is above a given threshold; otherwise, the supervisory controller resets the estimator.

The estimator is described in Section 7.1. The estimation algorithm was set as follows: The polynomial order was set to be $S = 6$, the bounds on the mass estimate were $800 \leq m \leq 2300$ kilograms, and the bounds on the damping coefficient were $4000 \leq B_z \leq 28000$ Newton seconds per meter. The value of σ_k^2 was estimated at each iteration to be $\sigma_k^2 = 0.007 + \left(y_k - C(\hat{\xi}_{k-1})P(\hat{\xi}_{k-1})X(t_k)\right)^2$. The MMSE estimator used 25 Monte Carlo integration points, one of which was $\hat{\xi}_{k-1}$.

8.2.1 Results

The recursive estimation algorithm based on polynomial chaos theory and the reduced order full-car model was applied to the 40 experiments in Table 6.2 of Chapter 6. The last two columns of Table 8.2 report the results for the estimated SNR and the error in the estimated sprung mass. The error in the mass estimate was calculated as follows:

$$Error\ in\ Mass\ Estimate = \frac{mean\left(|True\ Mass - Estimated\ Mass|_{t_k=100}^{t_{end}}\right)}{True\ Mass} \times 100.$$

Table 8.2: Polynomial Chaos Results for the Reduced Full-car Estimator.

Road No.	Description	Target Speed (mph)	Name	SNR Reduced Full-car	Estimated SNR Full-car	Error in Mass Estimate
1	Rough Dirt	10	Light10	23.6114	**18**	**8.5%**
2	Rough Dirt	5	Light5	18.4218	**15**	**2.6%**
3	Rough Dirt	10	PL10	16.9782	**14**	**2.7%**
4	Rough Dirt	15	Light15	15.0068	**15**	**15%**
5	Rough Dirt	3	Light2	13.7319	**11**	**4.8%**
6	Rough Dirt	5	PL5	12.6351	**11**	**1.7%**
7	Rough Dirt	5	NCparking	12.1297	**11**	**2.0%**
8	Rough Dirt	15	PL15	12.0064	**12**	**7.7%**
9	Rough Dirt	3	PL2	10.1264	**9.0**	**1.9%**
10	Rough Dirt	10	hvnew10	8.679	**7.3**	**3.6%**
11	Paved	30	LoopValid	8.0943	**8.5**	**0.007%**
12	Smooth Dirt	15	ltlll15	7.2456	**8.2**	**4.8%**
13	Paved	30	NearGR	6.7477	**6.9**	**1.2%**
14	Smooth Dirt	15	hvlll15	6.2971	**7.4**	**8.9%**
15	Smooth Dirt	10	ltlll10	5.7535	**8.0**	**2.9%**
16	Paved	70	Free696	5.7086	**4.8**	**21%**
17	Rough Dirt	5	hvnew5	5.6219	**6.7**	**1.5%**
18	Smooth Dirt	10	hvjoy10	5.0007	**4.7**	**8.7%**
19	Paved	30	NeGR3800	4.5858	**4.5**	**6.0%**
20	Rough Paved	30	LtFullGas	4.2848	**4.1**	**4.9%**

Road No.	Description	Target Speed (mph)	Name	SNR Reduced Full-car	Estimated SNR Full-car	Error in Mass Estimate
21	Smooth Dirt	10	hvlll10	4.24	**6.4**	**3.7%**
22	Paved	35	OW2Brick	4.2164	**3.8**	**4.2%**
23	Smooth Dirt	20	ltlll20	3.7439	**3.7**	**6.7%**
24	Paved	70	FreeNew	3.4469	**3.2**	**10%**
25	Paved	70	I23ply2sal	3.1145	**3.3**	**6.9%**
26	Paved/Dirt	50	PontWar	2.9586	**3.0**	**7.6%**
27	Rough Dirt	35	CFB2F	2.5737	**2.5**	**4.1%**
28	Rough Dirt	15	hvnew15	2.5283	**2.6**	**9.4%**
29	Rough Paved	30	GR3800	2.439	**2.4**	**3.1%**
30	Smooth Dirt	25	ltlll25	2.109	**2.3**	**4.0%**
31	Smooth Dirt	20	hvlll20	1.7685	**2.0**	**2.9%**
32	Smooth Dirt	25	FLkRdNrth	1.1677	**1.4**	**70%**
33	Paved/Dirt	50	InSaline	1.1436	**1.4**	**59%**
34	Rough Dirt	20	hvnew20	1.1291	**1.4**	**53%**
35	Rough Dirt	25	hvnew25	1.0088	**1.1**	**7.8%**
36	Smooth Dirt	25	hvlll25	0.9013	**1.2**	**42%**
37	Rough Dirt	30	hvnew30	0.6347	**0.7**	**9.8%**
38	Rough Dirt	20	N2BonC20	0.5571	**0.76**	**72%**
39	Rough Dirt	30	B2N30	0.394	**0.57**	**81%**
40	Rough Dirt	45	N2BonC45	0.2696	**0.38**	**70%**

Figure 8.3 through Figure 8.9 plot the sprung mass and SNR estimates for seven roads: Road Numbers 1, 4, 11, 16, 21, 31, and 40.

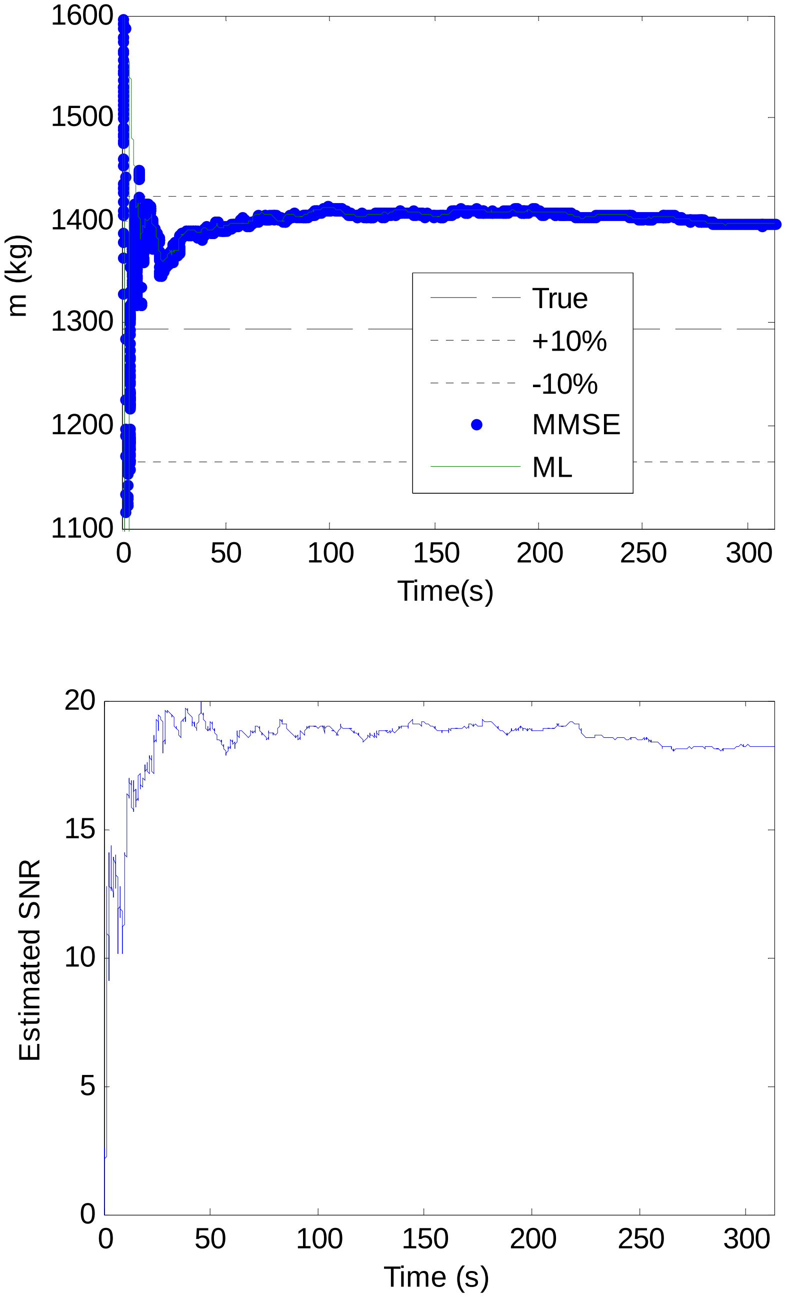


Figure 8.3: Estimates of sprung mass and SNR using the polynomial chaos estimators for Road No. 1. The dot sequence is calculated via the MMSE estimator of Chapter 5, and the solid line is calculated via the maximum likelihood (ML) estimator of Chapter 4.

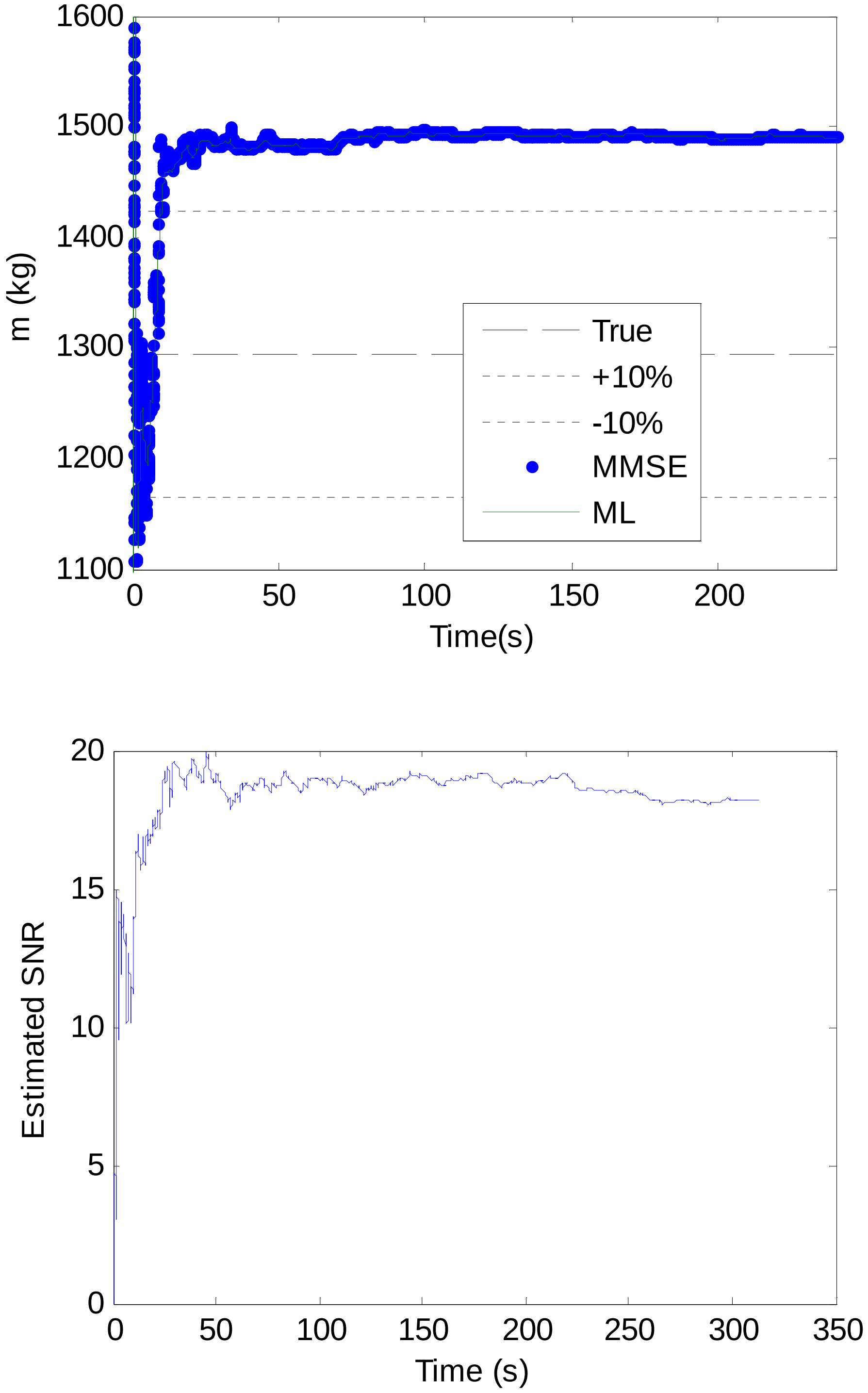


Figure 8.4: Estimates of sprung mass and SNR using the polynomial chaos estimators for Road No. 4. The dot sequence is calculated via the MMSE estimator of Chapter 5, and the solid line is calculated via the maximum likelihood (ML) estimator of Chapter 4.

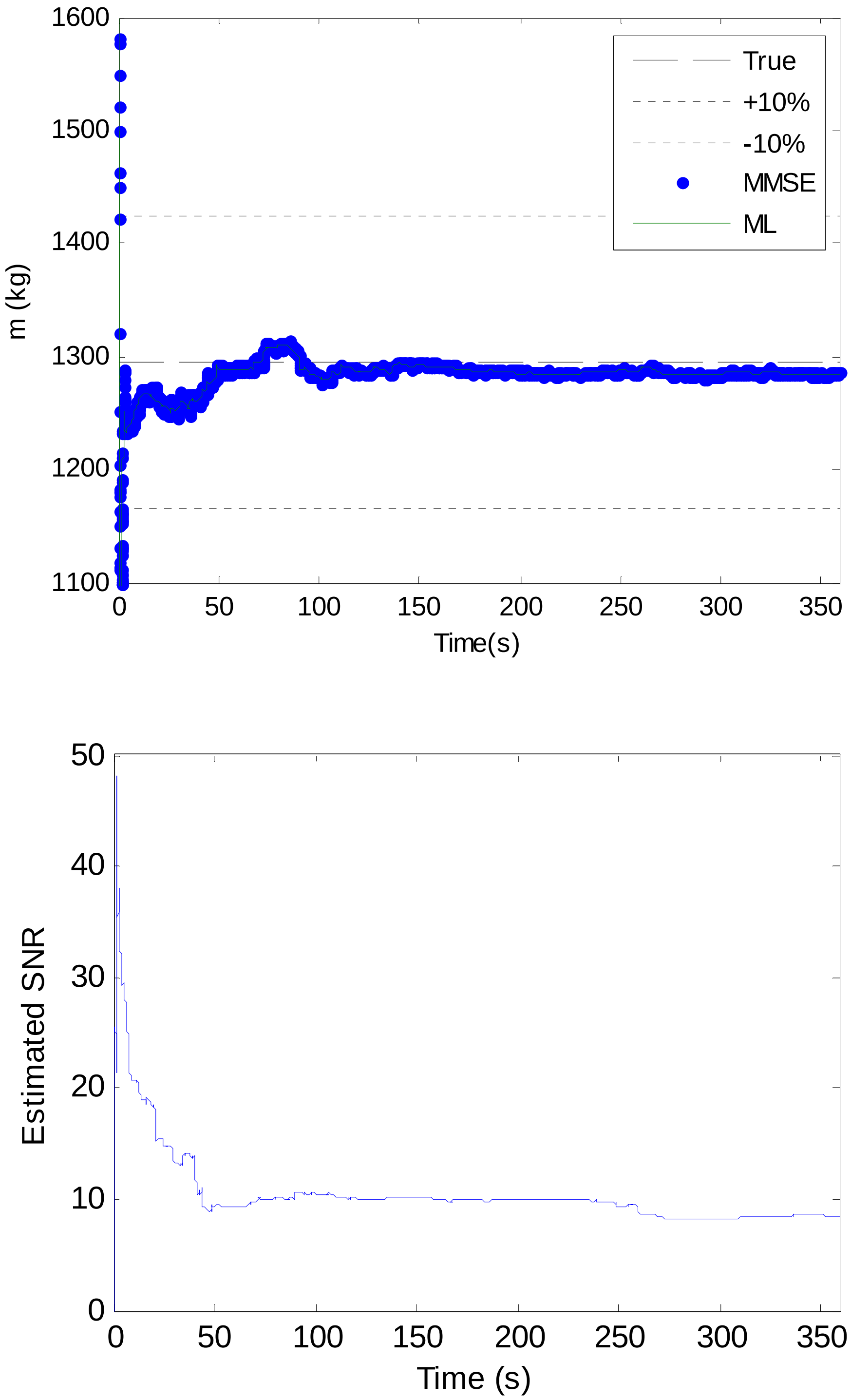
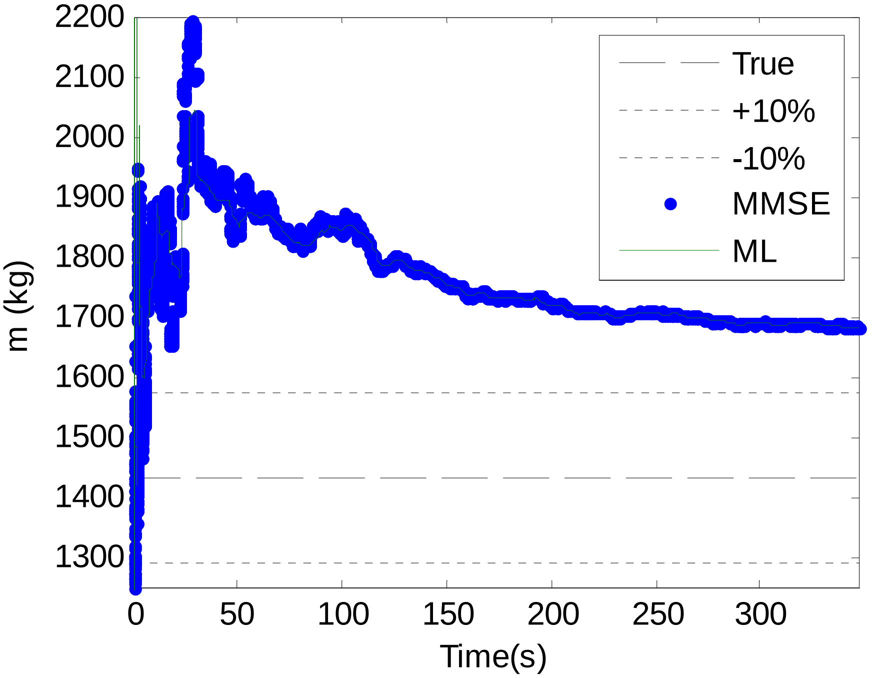


Figure 8.5: Estimates of sprung mass and SNR using the polynomial chaos estimators for Road No. 11. The dot sequence is calculated via the MMSE estimator of Chapter 5, and the solid line is calculated via the maximum likelihood (ML) estimator of Chapter 4.



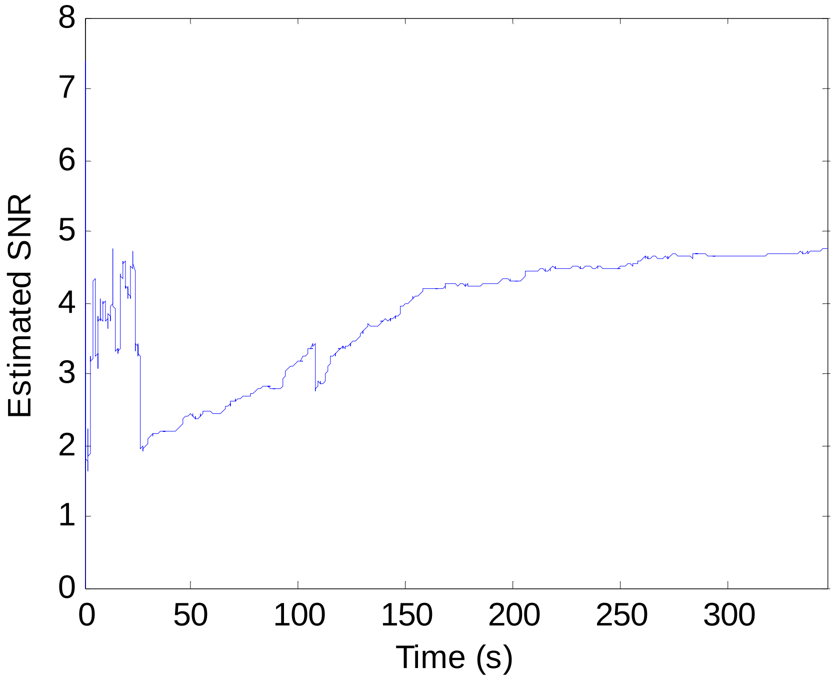


Figure 8.6: Estimates of sprung mass and SNR using the polynomial chaos estimators for Road No. 16. The dot sequence is calculated via the MMSE estimator of Chapter 5, and the solid line is calculated via the maximum likelihood (ML) estimator of Chapter 4.

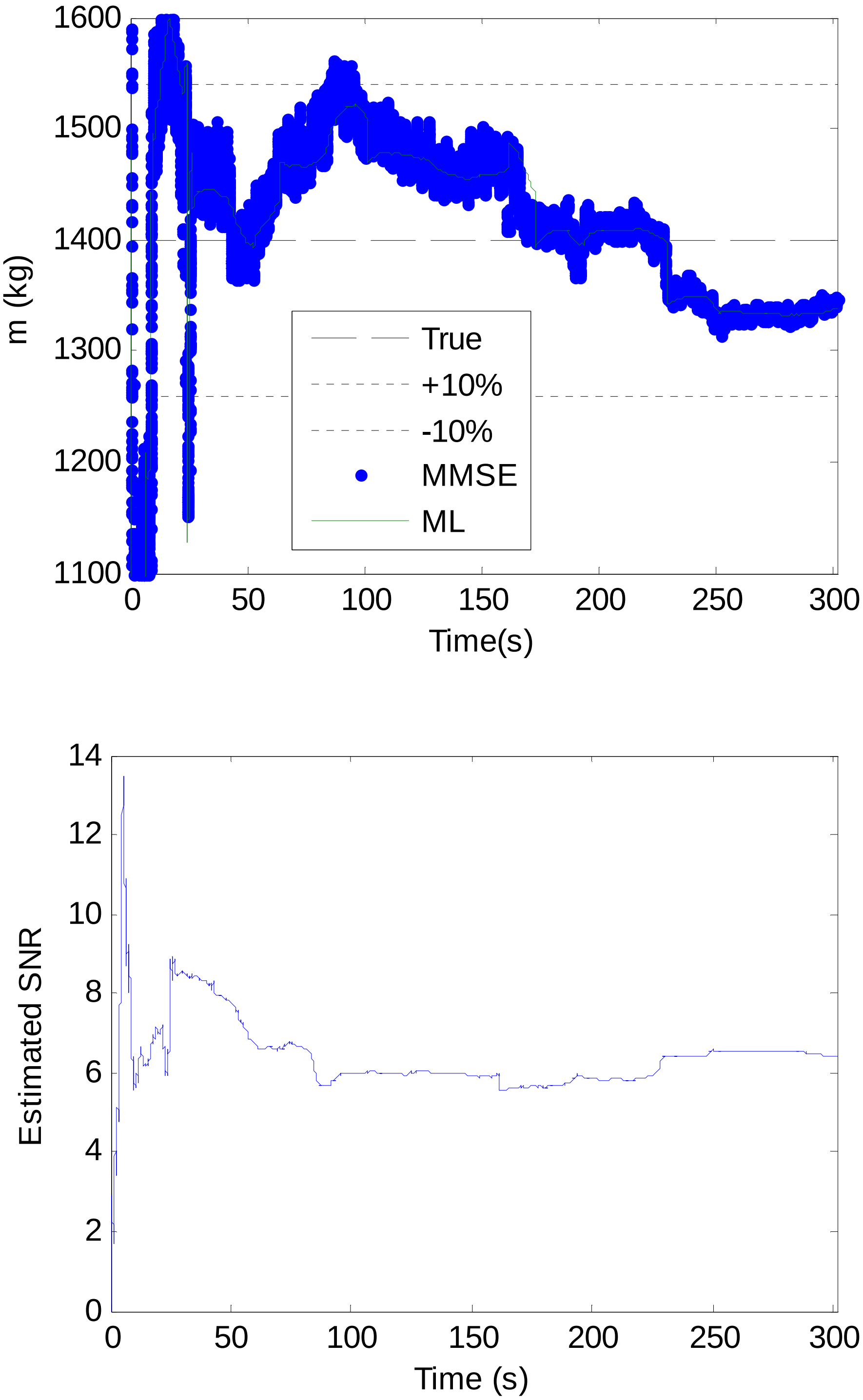


Figure 8.7: Estimates of sprung mass and SNR using the polynomial chaos estimators for Road No. 21. The dot sequence is calculated via the MMSE estimator of Chapter 5, and the solid line is calculated via the maximum likelihood (ML) estimator of Chapter 4.

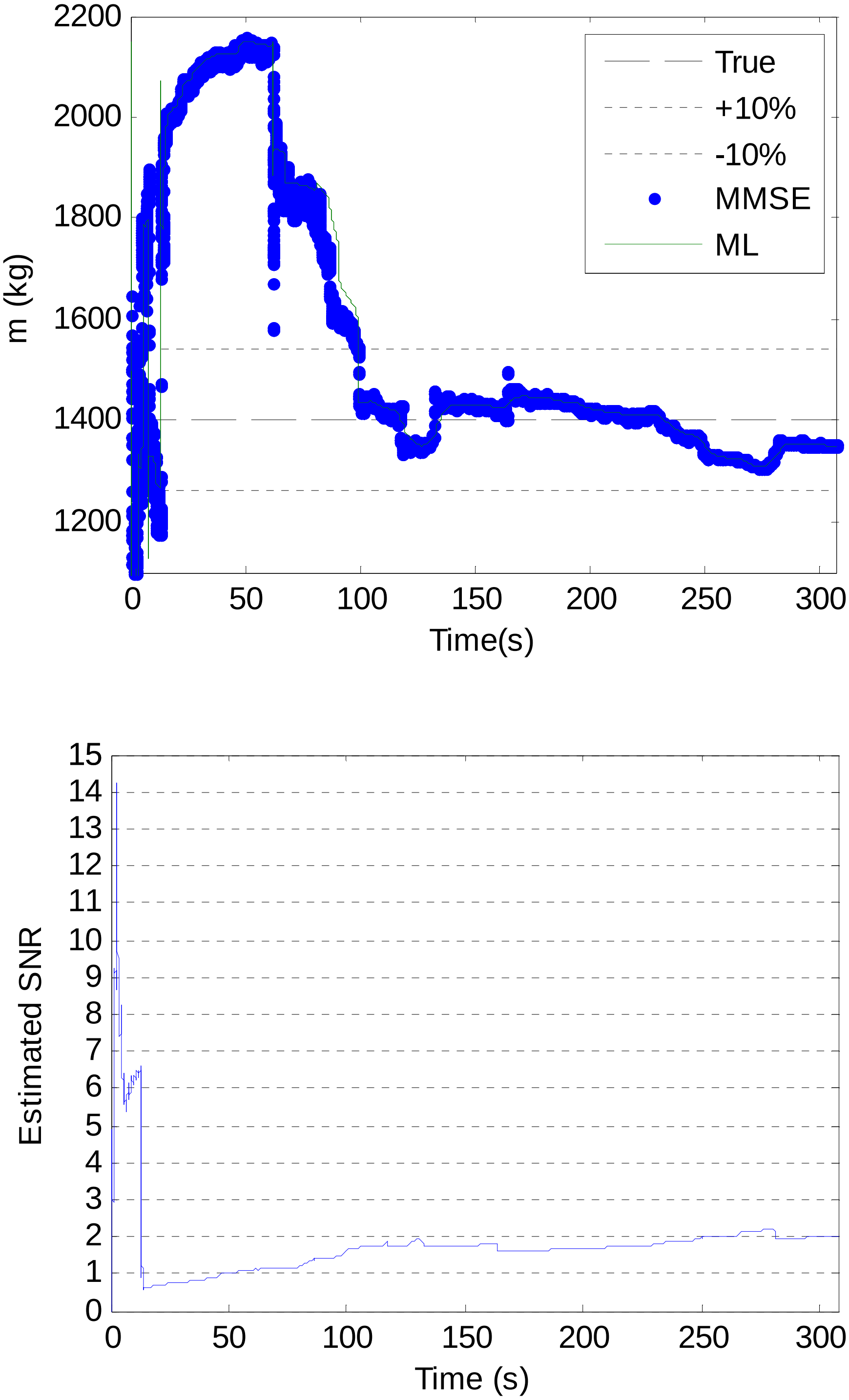
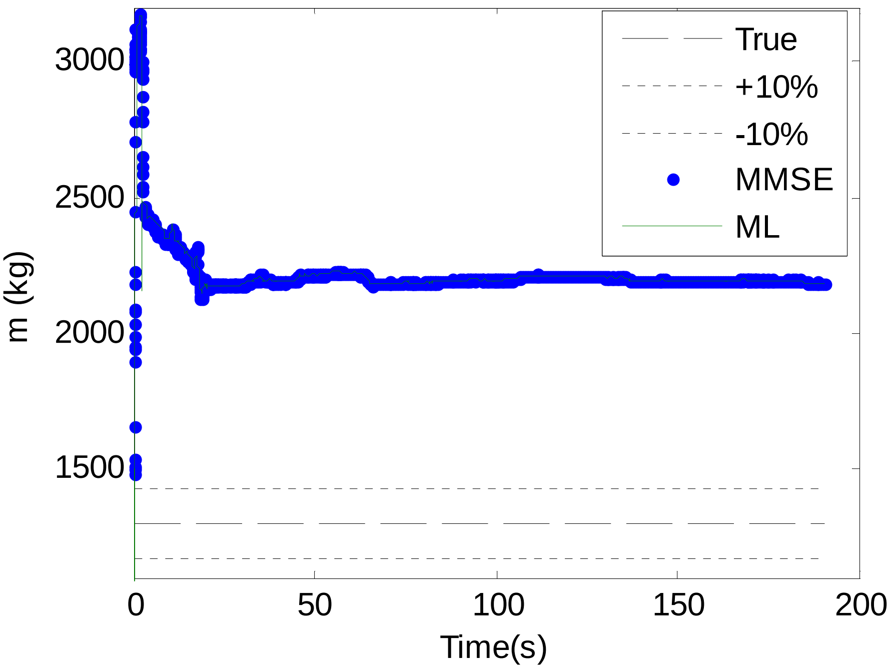


Figure 8.8: Estimates of sprung mass and SNR using the polynomial chaos estimators for Road No. 31. The dot sequence is calculated via the MMSE estimator of Chapter 5, and the solid line is calculated via the maximum likelihood (ML) estimator of Chapter 4.



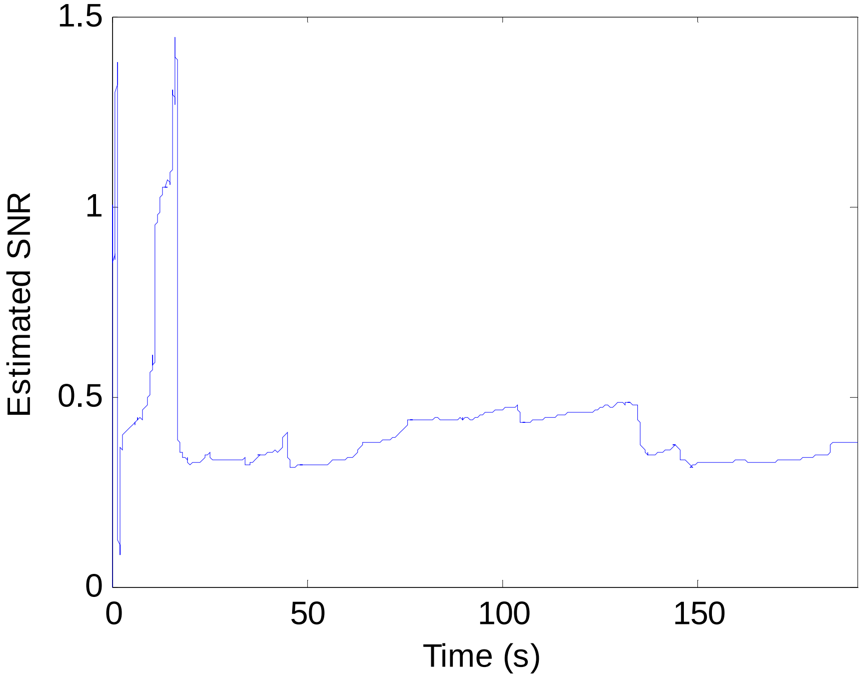


Figure 8.9: Estimates of sprung mass and SNR using the polynomial chaos estimators for Road No. 40. The dot sequence is calculated via the MMSE estimator of Chapter 5, and the solid line is calculated via the maximum likelihood (ML) estimator of Chapter 4.

8.2.2 Discussion of Results

The results of Table 8.2 show that with only two exceptions, an estimated SNR above 2.0 was a good indicator that the estimator would converge to within the acceptable ±10% error range. Out of 31 "predicted positives", there were only two "false positives" – a success rate of 94%. The two exceptions include Road No. 4 (see Figure 8.4) and Road No. 16 (see Figure 8.6). The experiment in Road No. 4 took place in an oval shaped, dirt parking lot with a driving speed of 15 miles per hour (24 kilometers per hour). At that speed in the small parking lot, the vehicle experienced significant yaw velocities and lateral velocities; these conditions violate the modeling assumptions of Section 6.3 and may account for the 15% error in the mass estimate (conversely, however, this does not answer the question of why the SNR was so high). The experiment in Road No. 16 was on a sectioned freeway at a speed of 70 miles per hour (112 kilometers per hour). Because the vehicle had two adult passengers in the rear seat for this experiment, it is the author's hypothesis that the rigid body assumption for the sprung mass was violated causing the estimator to fail. Seven out of nine experiments with an estimated SNR below 2.0 converged to a value outside of the ±10% error range.

Figure 8.3 through Figure 8.9 illustrate that the polynomial chaos based estimators, *i.e.*, the Minimum Mean Squared Error (MMSE) estimator of Chapter 5 and the Maximum Likelihood (ML) estimator of Chapter 4, converge to nearly identical estimates of the vehicle mass. Because of the inherent randomness in Monte Carlo integration, the MMSE estimator is more "noisy" than the maximum likelihood estimator. And as discussed in Chapter 5, the MMSE estimator is more computationally demanding than the maximum likelihood estimator. Therefore, these results suggest that the maximum likelihood estimator is more appropriate for the mass estimation problem.

The results of Table 8.2 show that the proposed mass estimators of this chapter can be used to robustly estimate the mass of the vehicle to within ±10% of the true value.

8.3 Experimental Validation of Regressor Methods

The regressor model approach to vehicle mass estimation was outlined in Section 7.2. This section, Section 8.3, uses experiments performed on Road No. 1 (best SNR case) and Road No. 12 of Table 6.2 to demonstrate the performance of the regressor model based estimators.

This dissertation chose the denominator $\Lambda(s)$ (see Section 7.2) to be second order, *i.e.*, $\Lambda(s) = s^2 + 2\zeta\omega s + \omega^2$. In Figure 8.10, the value of ζ was fixed at $\zeta = 0.1$ for RLS and $\zeta = 0.2$ for RTLS, and the value of ω was varied between 6 and 20 rad/s (0.95 – 3.2 Hz). In Figure 8.11, the value of ω was fixed at 13 for RLS and 14 for RTLS, and the value of ζ was varied between 0.1 and 0.8.

8.3.1 Discussion of Results

Figure 8.10 through Figure 8.12 illustrate two important limitations of the regressor model approaches for sprung mass estimation. Figure 8.10 and Figure 8.11 illustrate that the bias of the estimator is strongly dependent on the user-specified tuning of the filter denominator $\Lambda(s)$, even for data having a good signal-to-noise ratio. Figure 8.12 illustrates that although the tuning of the estimators was appropriate for Road No. 1 (see Figure 8.10 and Figure 8.11), it was not appropriate (within the 10% error limit) for

Road No. 12 (especially for the RLS algorithm). *I.e.*, the tuning of $\Lambda(s)$ is case dependent, and different speeds/terrains require different tuning.

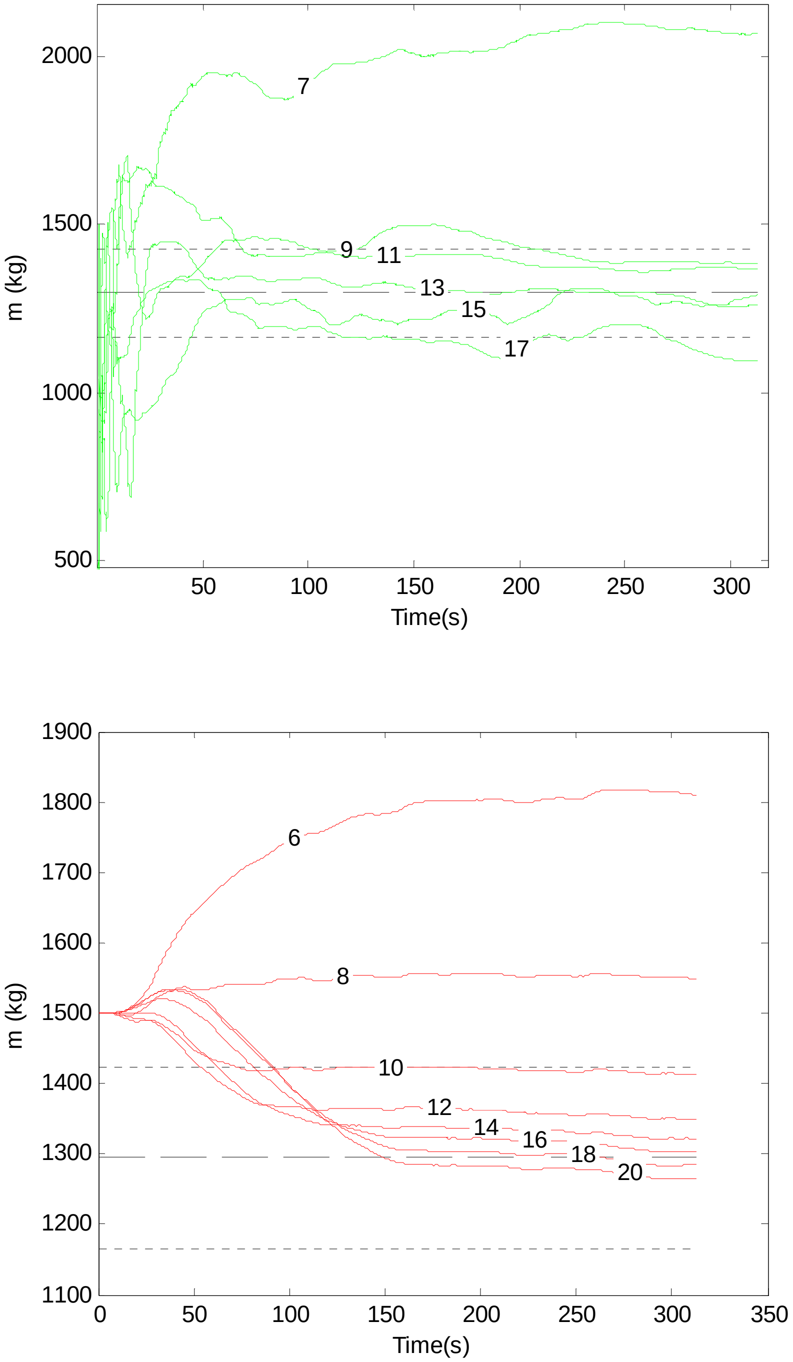
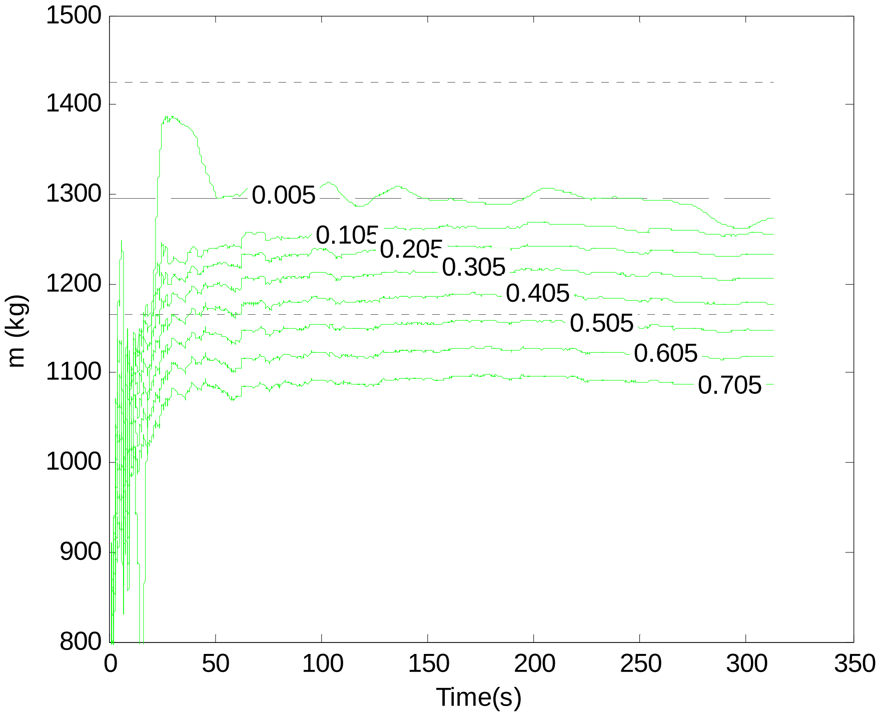


Figure 8.10: Convergence of RLS (top figure) and RTLS (bottom figure) for variations in ω.



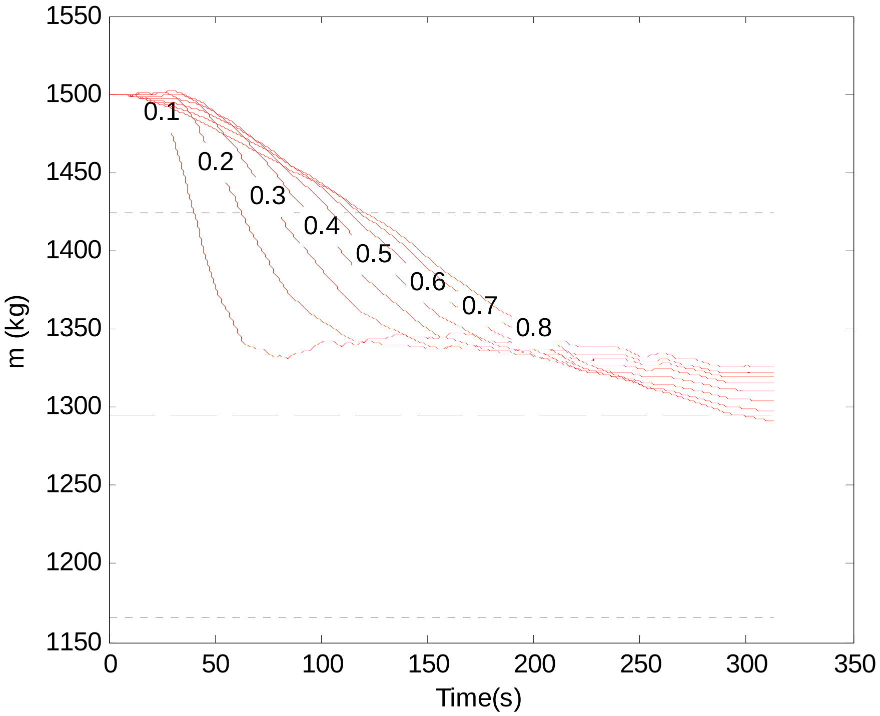


Figure 8.11: Convergence of RLS (top figure) and RTLS (bottom figure) for variations in ζ.

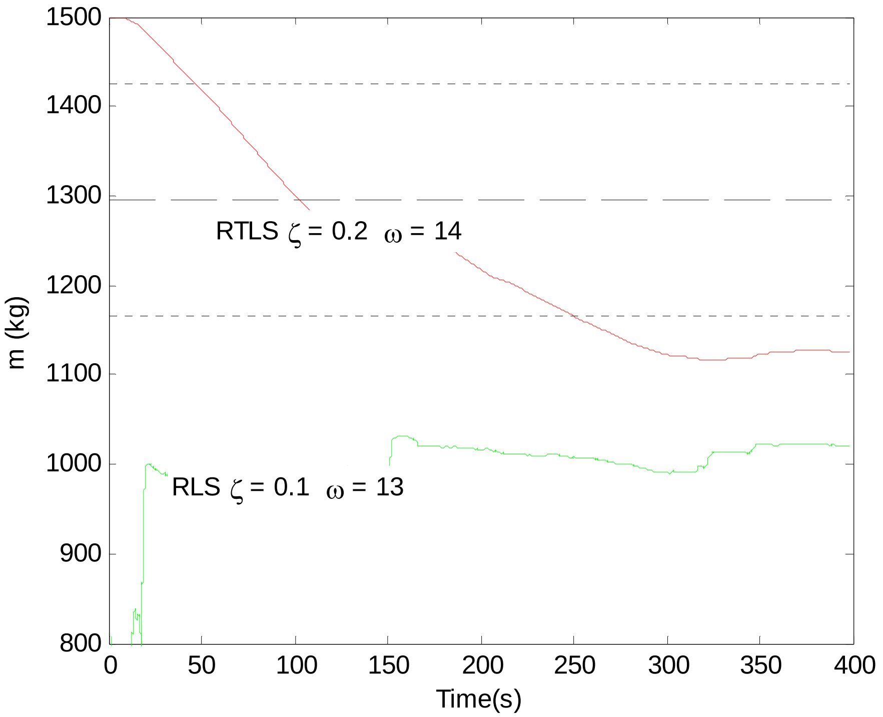


Figure 8.12: Convergence of RLS and RTLS for Road No. 12.

8.4 Experimental Validation of Filtering Methods

This section uses the experiments of Table 6.2 to explore two questions: (a) How well do the mass estimation techniques that use state-filtering perform? (b) Does a low estimator variance indicate good filter performance, *i.e.*, mass estimate within the acceptable 10% error range?

One important feature of the polynomial chaos estimators is their ability to iteratively estimate the Signal-to-Noise Ratio (SNR). The author is unaware of any current technique for estimating the SNR via the Extended Kalman Filter (EKF), Unscented Kalman Filter (UKF), or particle filter. These filtering algorithms, however, do have the ability to estimate the covariance of the estimated state vector (which includes the unknown parameters). This estimator covariance may provide a sense of how well the estimates can be trusted. If so, a supervisory controller could set a threshold on the estimator variance to indicate when the current estimate can be trusted.

8.4.1 Results

Section 7.3 presented the augmented equations used by the filtering algorithms to estimate the vehicle sprung mass. Appendix A2 summarizes the filtering algorithms and describes the tuning of the filtering methods. This section applies the EKF and UKF mass estimation algorithms to the 40 experiments of Table 6.2; the results are given in Table 8.3. The notation “EKF StDev” corresponds to the square root of the estimated variance of the mass estimate produced by the EKF. The “EKF Mass Error” and “UKF Mass Error” are calculated in the same way as “Error in Mass Estimate” was calculated in Section 8.2.1.

This dissertation applied a particle filter having 25 particles to most of the experiments of Table 6.2. The particle filter was significantly more computationally demanding than the other estimators, and the results were also less repeatable (likely due to the stochastic nature, *e.g.* simulated process noise and Monte Carlo integration, of the particle filter). Therefore, the results of the particle filter are not reported in Table 8.3; however, to demonstrate the particle filter performance, Figure 8.13 and Figure 8.14 plot the results of the particle filter along with the results of the EKF and UKF for Road No. 1 and 11.

Table 8.3: State-Filtering Results for the Reduced Full-car Estimator.

Road No.	Description	Target Speed (mph)	Name	SNR Reduced Full-car	EKF StDev	EKF Mass Error	UKF StDev	UKF Mass Error
1	Rough Dirt	10	Light10	23.6114	21.1	8.9%	21.5	8.9%
2	Rough Dirt	5	Light5	18.4218	33.9	3.7%	32.3	3.1%
3	Rough Dirt	10	PL10	16.9782	19.2	2.8%	19.4	2.7%
4	Rough Dirt	15	Light15	15.0068	18.2	15%	18.6	15%
5	Rough Dirt	3	Light2	13.7319	49.3	6.5%	46.9	5.7%
6	Rough Dirt	5	PL5	12.6351	29.2	0.94%	28.3	1.6%
7	Rough Dirt	5	NCparking	12.1297	65	0.19%	60.6	1.5%
8	Rough Dirt	15	PL15	12.0064	22.5	8.1%	21.7	7.6%
9	Rough Dirt	3	PL2	10.1264	45.7	0.67%	43.4	0.79%
10	Rough Dirt	10	hvnew10	8.679	82.7	6.6%	85.6	4.5%
11	Paved	30	LoopValid	8.0943	37.5	1.1%	34.9	0.27%
12	Smooth Dirt	15	ltlll15	7.2456	71.8	1.4%	68	4.3%
13	Paved	30	NearGR	6.7477	66.7	6.8%	59.8	2.8%
14	Smooth Dirt	15	hvlll15	6.2971	76.4	5.4%	71.6	8.2%
15	Smooth Dirt	10	ltlll10	5.7535	95.2	1.0%	111	0.72%
16	Paved	70	Free696	5.7086	42.6	15%	49.6	25%
17	Rough Dirt	5	hvnew5	5.6219	97.8	1.7%	118	1.3%
18	Smooth Dirt	10	hvjoy10	5.0007	94.2	7.9%	108	9.0%
19	Paved	30	NeGR3800	4.5858	35.8	6.1%	35	7.2%
20	Rough Paved	30	LtFullGas	4.2848	46.7	9.0%	42.6	6.4%

Road No.	**Description**	**Target Speed (mph)**	**Name**	**SNR Reduced Full-car**	**EKF StDev**	**EKF Mass Error**	**UKF StDev**	**UKF Mass Error**
21	Smooth Dirt	10	hvlll10	4.24	**101**	**4.2%**	**125**	**3.6%**
22	Paved	35	OW2Brick	4.2164	**49.9**	**3.8%**	**46.7**	**2.1%**
23	Smooth Dirt	20	ltlll20	3.7439	**84.02**	**4.5%**	**82.9**	**3.2%**
24	Paved	70	FreeNew	3.4469	**27.4**	**11%**	**430**	**1355%**
25	Paved	70	I23ply2sal	3.1145	**104.1**	**10%**	**336**	**55.6%**
26	Paved/Dirt	50	PontWar	2.9586	**47.8**	**6.4%**	**218**	**119%**
27	Rough Dirt	35	CFB2F	2.5737	**143**	**31%**	**411**	**204%**
28	Rough Dirt	15	hvnew15	2.5283	**116.2**	**13%**	**311**	**96.9%**
29	Rough Paved	30	GR3800	2.439	**51.9**	**5.5%**	**47.7**	**3.2%**
30	Smooth Dirt	25	ltlll25	2.109	**120.3**	**9.5%**	**406**	**312%**
31	Smooth Dirt	20	hvlll20	1.7685	**105**	**2.4%**	**340**	**153%**
32	Smooth Dirt	25	FLkRdNrth	1.1677	**183**	**44%**	**265**	**266%**
33	Paved/Dirt	50	InSaline	1.1436	**86.6**	**130%**	**240**	**392%**
34	Rough Dirt	20	hvnew20	1.1291	**172.3**	**21%**	**459**	**224%**
35	Rough Dirt	25	hvnew25	1.0088	**53.61**	**7.4%**	**544**	**9500%**
36	Smooth Dirt	25	hvlll25	0.9013	**124.7**	**20%**	**472**	**3854%**
37	Rough Dirt	30	hvnew30	0.6347	**85.7**	**14%**	**548**	**515%**
38	Rough Dirt	20	N2BonC20	0.5571	**110.1**	**48%**	**535**	**2329%**
39	Rough Dirt	30	B2N30	0.394	**141.1**	**34%**	**550**	**1510%**
40	Rough Dirt	45	N2BonC45	0.2696	**116**	**160%**	**538**	**5483%**

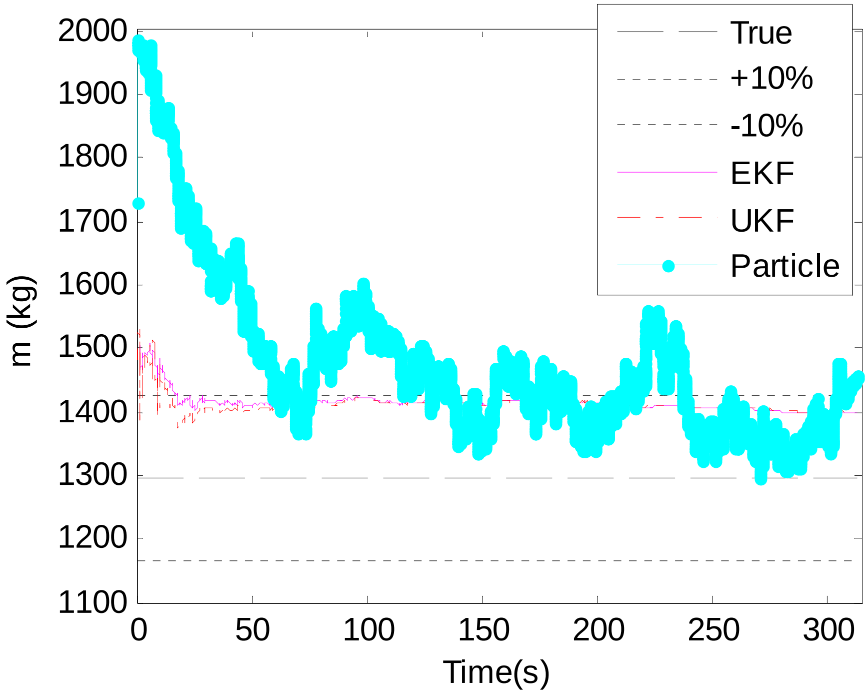


Figure 8.13: Convergence of EKF, UKF, and Particle filter for Road No. 1.

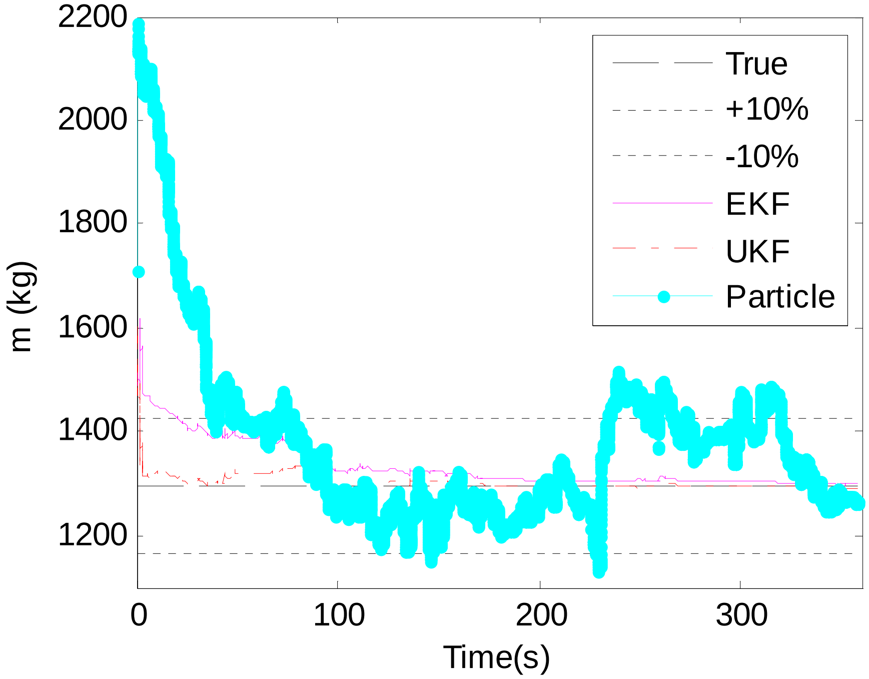


Figure 8.14: Convergence of EKF, UKF, and Particle filter for Road No. 11.

8.4.2 Discussion of Results

The first objective was to address the question, "How well do the mass estimation techniques that use state-filtering perform?" Comparing Table 8.3 with Table 8.2 shows that whenever the EKF or UKF algorithm estimated the mass successfully within the acceptable 10% error range, then so did the polynomial chaos methods. Also, the polynomial chaos algorithms calculated estimates within the acceptable 10% error range for Road No. 24, 27, 28, and 37; the EKF and UKF algorithms failed in these cases. The UKF algorithm also failed for Road No. 25, 26, 30, 31, and 35 whereas the EKF and polynomial chaos approaches were successful for these experiments.

The second question was, "Does a low estimator variance indicate good filter performance, *i.e.*, mass estimate within the acceptable 10% error range?" Table 8.3 shows that an estimator standard deviation less than 106 kilograms corresponded to an EKF success rate of 84% (26 successes out of 31 predicted successes). An estimator standard deviation below 130 kilograms corresponded to a UKF success rate of 92% (22 successes out of 24 predicted successes).

Based on these results, this section concludes that the performance of the EKF and UKF was less accurate than the polynomial chaos methods for the vehicle mass estimation problem. It also concludes that the estimator standard deviation is a good indicator of estimator success based on the 84% success rate for the EKF and 92% success rate for the UKF.

This dissertation does not recommend the particle filter for mass estimation due to its high computational demand and low repeatability compared with the EKF and UKF in the experiments of this section.

8.5 Conclusions

The experimental results of this chapter demonstrate that the proposed method which joins polynomial chaos estimators with base excitation models is viable for vehicle sprung mass estimation for vehicles driving on rough (and smooth) terrain. No prior knowledge of the terrain profile was required, and no active/semi-active suspension was required. The polynomial chaos approach was compared with regressor model approaches and filtering approaches. The regressor methods are concluded as not appropriate for this sprung mass estimation approach because of their sensitivity to tuning parameters. The polynomial chaos technique had a success rate of 94% whereas the EKF had a success rate of 84% and the UKF had a success rate of 92%. Based on these results, this chapter concludes that the polynomial chaos approach is best suited for the vehicle mass estimation problem of this dissertation.

The experimental results also demonstrated that the signal-to-noise ratio calculated iteratively by the polynomial chaos estimator is a good indicator of estimator success (94% of the time). For the filtering methods, the estimated variance of the mass estimate was a good (84% for EKF and 92% for UKF) indicator of estimator success.

Chapter 9

Conclusions

This dissertation has addressed recursive parameter estimation of state space systems using polynomial chaos theory applied to vehicle mass estimation for rough terrain. It reviewed the scientific literature that addresses state space estimators based on polynomial chaos theory and has proposed novel maximum likelihood and Bayesian estimation algorithms. This dissertation also reviewed the vehicle mass estimation literature and has proposed novel approaches that rely on base excitation modeling of vertical vehicle dynamics. The unique concepts of this dissertation which join this dissertation's estimators and base excitation modeling concepts provide a solution to sprung mass estimation of vehicles traversing rough terrain that was successful 94% of the time (only two false positives out of 31 predicted successes). The estimator that combined the extended Kalman filter (EKF) with the base excitation models was successful 84% of the time (four false positives out of 31 predicted successes), and the estimator based on the unscented Kalman filter was successful 92% of the time (two false positives out of 24 predicted successes).

This dissertation has successfully demonstrated the proposed polynomial chaos based estimation algorithms in an experimental setting. It has validated its proposed base excitation models of vehicle ride dynamics against experimental data. The combination of the base excitation modeling and the polynomial chaos estimators provides a viable solution to the vehicle mass estimation problem for rough terrain.

Appendices

Appendix A1

Treatment of Unknown Initial Conditions in Polynomial Chaos Estimation

The development of the polynomial chaos estimators of Chapters 4 and 5 assumes that the initial conditions are either known or estimated as part of the estimation routine. This assumption may be relaxed in special cases; if the underlying dynamic equations are asymptotically stable for all possible realizations of the unknown parameters, the effect of initial conditions may become negligible as time progresses. This appendix provides a sketch of the proof for linear systems that are asymptotically stable and time-invariant. Specifically, this appendix proves that the contribution of the initial conditions on the objective function is bounded. Then based on an intuitive argument it conjectures that the objective function tends to infinity as time increases. Hence the effect of the initial conditions on the estimate becomes negligible.

The development assumes that the input signal satisfies some type of persistent excitation requirement. *E.g.*, if the initial conditions and excitation source are both zero, the dynamics of the system will not be excited, and any realization of $\hat{\xi}_k$ will satisfy $\hat{\xi}_k = \operatorname{argmin}_{\xi} J_k(\xi)$. Since this appendix assumes the initial conditions are unknown, the input signal must be persistently exciting in some sense. Similar persistent excitation requirements are common for other estimation algorithms to guarantee that the estimate is unique (*e.g.*, see Section 5.2 of (Ioannou and Sun, 1996)). Another assumption that this appendix makes is that the input signal is bounded.

Consider the following linear, time-invariant state and output equations in which the matrices $A_c \in \mathbb{R}^{n_s \times n_s}$, $B_c \in \mathbb{R}^{n_s \times n_u}$, and $C \in \mathbb{R}^{n_y \times n_s}$ may be functions of the unknown polynomial chaos vector $\xi \in \mathbb{R}^{n_p}$:

$$\begin{aligned} \dot{x}(t,\xi) &= A_c x(t,\xi) + B_c u(t) \\ y_k &= Cx(t_k,\xi) + v_k. \end{aligned} \tag{A1.1}$$

The state vector is $x(t,\xi) \in \mathbb{R}^{n_s}$, and $u(t) \in \mathbb{R}^{n_u}$ is the input vector. The discrete-time output vector is $y_k \in \mathbb{R}^{n_y}$. The true output response of the system is $Cx(t,\xi)$, however the measured output signal y_k is corrupted by the additive Gaussian noise sequence $v_k \in \mathbb{R}^{n_y}$. The noise sequence v_k, $k = 0,1,\ldots$ is zero mean with positive definite covariance matrix $R_k \in \mathbb{R}^{n_y \times n_y}$.

Because the system of Equation (A1.1) is asymptotically stable for all realizations of ξ, applying the Galerkin projection (see Section 4.1.1) to Equation (A1.1) results in an asymptotically stable, (higher order) deterministic set of state equations with stochastic output equations which can be written as follows:

$$\begin{aligned} \dot{X}(t) &= AX(t) + Bu(t) \\ \hat{y}_k &= \hat{C}\mathbb{P}(\xi)X(t_k). \end{aligned} \tag{A1.2}$$

Here, $x(t_k,\xi) = \mathbb{P}(\xi)X(t_k)$ follows from Equation (4.14) of Chapter 4. The eigenvalues of the matrix $A \in \mathbb{R}^{n_s r \times n_s r}$ (which is not a function of ξ) all have negative real parts; the matrix $B \in \mathbb{R}^{n_s r \times n_u}$ is not a function of ξ. The term $\hat{y}_k \in \mathbb{R}^{n_y}$ is an approximation of the output y_k, and $\hat{C} \in \mathbb{R}^{n_y \times n_s}$ (which may be a function of ξ) is an approximation of C. Because the state equations of (A1.2) are deterministic and time-invariant, its solution is given by (see Section 4.2 of (Chen, 1999)):

$$X(t) = \exp\{At\}\, X(0) + \int_0^t \exp\{A(t-\tau)\}\, Bu(\tau)d\tau. \tag{A1.3}$$

If the initial conditions $x(0)$ for (A1.1) were known, then the best choice of the initial conditions $X(0)$ for (A1.2) could be determined as a function of $x(0)$. However, this development assumes that the initial conditions $x(0)$ are unknown, and hence $X(0)$ is user-selected (and known). Using (A1.3), the output equation of (A1.2) can be written as follows:

$$\begin{gathered} \hat{y}_k = \hat{y}_k^{(h)} + \hat{y}_k^{(f)} \\ \hat{y}_k^{(h)} = \hat{C}\mathbb{P}(\xi)\exp\{At_k\}\, X(0), \\ \hat{y}_k^{(f)} = \hat{C}\mathbb{P}(\xi)\int_0^{t_k} \exp\{A(t_k-\tau)\}\, Bu(\tau)d\tau. \end{gathered} \tag{A1.4}$$

The measured system output y_k can be written in a similar manner:

$$\begin{gathered} y_k = y_k^{(h)} + y_k^{(f)} \\ y_k^{(h)} = C\exp\{A_c t_k\}\, x(0), \\ y_k^{(f)} = C\int_0^{t_k} \exp\{A_c(t_k-\tau)\}\, B_c u(\tau)d\tau + v_k. \end{gathered} \tag{A1.5}$$

The superscripts h and f denote the homogenous (or non-forced) and forced responses respectively. The initial conditions, $x(0)$ and $X(0)$ appear only in the homogenous responses. The proof that the effect of initial conditions on the parameter estimates becomes negligible as time progresses will follow from showing that the homogenous responses converge to zero exponentially in time. Then, as time progresses, the effect of the homogenous response becomes negligible and hence the effect of initial conditions also becomes negligible.

The following development assumes that the eigenvalues of the matrices A_c and A are all distinct so that A_c and A can be written in diagonal form. (This development can be extended to the general case by using the Jordan form instead of the diagonal form for the matrices A_c and A.) Then these matrices have the following diagonal form (see Section 3.5 of (Chen, 1999)):

$$\begin{aligned} A_c &= M_c \Lambda_c M_c^{-1} \\ A &= M \Lambda M^{-1}. \end{aligned} \tag{A1.6}$$

Here, $\Lambda_c \in \mathbb{C}^{n_s \times n_s}$ is a diagonal matrix containing the eigenvalues of A_c as the diagonal entries, and $\Lambda \in \mathbb{C}^{n_s r \times n_s r}$ is a diagonal matrix containing the eigenvalues of A as the diagonal entries. The matrices $M_c \in \mathbb{C}^{n_s \times n_s}$ and $M \in \mathbb{C}^{n_s r \times n_s r}$ are composed of eigenvectors of A_c and A respectively. Using (A1.6), the homogenous responses $y_k^{(h)}$ and $\hat{y}_k^{(h)}$ can be written as follows:

$$\begin{aligned} y_k^{(h)} &= C \exp\{M_c \Lambda_c M_c^{-1} t_k\}\, x(0) \\ \hat{y}_k^{(h)} &= \hat{C} \mathbb{P}(\xi) \exp\{M \Lambda M^{-1} t_k\}\, X(0). \end{aligned} \tag{A1.7}$$

Then using a series expansion of the exponential function and also using the fact that t_k is a scalar variable, $y_k^{(h)}$ can be written as

$$y_k^{(h)} = C \left(\sum_{n=0}^{\infty} \frac{(M_c \Lambda_c M_c^{-1})^n {t_k}^n}{n!} \right) x(0) \tag{A1.8}$$

Note that $(M_c\Lambda_c M_c^{-1})^n = (M_c\Lambda_c M_c^{-1})(M_c\Lambda_c M_c^{-1})(M_c\Lambda_c M_c^{-1})\cdots = M_c\Lambda_c^n M_c^{-1}$, and since M_c is not a function of n, it can be moved outside the series expansion, then $y_k^{(h)}$ and $\hat{y}_k^{(h)}$ can be written as follows:

$$\begin{aligned} y_k^{(h)} &= CM_c\left(\sum_{n=0}^{\infty}\frac{\Lambda_c^n {t_k}^n}{n!}\right)M_c^{-1}x(0) = CM_c\exp\{\Lambda_c t_k\}M_c^{-1}x(0) \\ \hat{y}_k^{(h)} &= \hat{C}\mathbb{P}(\xi)M\exp\{\Lambda t_k\}M^{-1}X(0) \end{aligned} \tag{A1.9}$$

Because all of the entries of Λ_c and Λ have negative real parts, each entry of the matrices $\exp\{\Lambda t_k\}$ and $\exp\{\Lambda_c t_k\}$ converge to zero at an exponential rate as time progresses. Therefore, $y_k^{(h)}$ and $\hat{y}_k^{(h)}$ also converge to zero at an exponential rate as time progresses. Due to the persistent excitation property of the input signal, the effect of the initial conditions becomes negligible in the complete responses y_k and $\hat{y}_k$ as time increases.

The objective function of Equation (4.19) can be written in terms of the homogeneous and forced responses:

$$\begin{aligned} J_k(\xi) &= \frac{1}{2}\sum_{\tau=0}^{k}\left(y_\tau^{(h)} + y_\tau^{(f)} - \left(\hat{y}_\tau^{(h)} + \hat{y}_\tau^{(f)}\right)\right)^T R_\tau^{-1}\left(y_\tau^{(h)} + y_\tau^{(f)} \right. \\ &\qquad \left. - \left(\hat{y}_\tau^{(h)} + \hat{y}_\tau^{(f)}\right)\right) \\ &= J^{(h)} + J^{(h,f)} + J^{(f)} \\ J^{(h)} &:= \frac{1}{2}\sum_{\tau=0}^{k}\left(y_\tau^{(h)} - \hat{y}_\tau^{(h)}\right)^T R_\tau^{-1}\left(y_\tau^{(h)} - \hat{y}_\tau^{(h)}\right) \\ J^{(h,f)} &:= \frac{1}{2}\sum_{\tau=0}^{k}\left(y_\tau^{(h)} - \hat{y}_\tau^{(h)}\right)^T R_k^{-1}\left(y_\tau^{(f)} - \hat{y}_\tau^{(f)}\right) \end{aligned} \tag{A1.10}$$

$$J^{(f)} := \frac{1}{2}\sum_{\tau=0}^{k}\left(y_\tau^{(f)} - \hat{y}_\tau^{(f)}\right)^T R_\tau^{-1}\left(y_\tau^{(f)} - \hat{y}_\tau^{(f)}\right)$$

Using (A1.9), the terms $J^{(h)}$ and $J^{(h,f)}$ are functions of the initial conditions; the term $J^{(f)}$ is not a function of the initial conditions. The remainder of the proof requires showing that $J^{(h)}$ and $J^{(h,f)}$ are convergent, and hence their effect is limited, whereas $J^{(f)}$ goes to infinity as time progresses to infinity.

This appendix first shows that $J^{(h)}$ converges to a finite value. Given any two vectors $w, z \in \mathbb{R}^{n_y}$ and the (positive definite) inverse covariance matrix R_k^{-1}, the inner product $\langle w, z\rangle := w^T R_k^{-1} z$ on the vector space $\mathbb{R}^{n_y}$ defines an inner product space in which the Cauchy-Schwartz and triangle inequalities apply. The purely homogenous part $J^{(h)}$ of the objective function can be written in terms of this inner product as follows:

$$J^{(h)} = \frac{1}{2}\sum_{\tau=0}^{k}\langle y_\tau^{(h)} - \hat{y}_\tau^{(h)}, y_\tau^{(h)} - \hat{y}_\tau^{(h)}\rangle. \qquad \text{(A1.11)}$$

Then by the triangle inequality, the following condition holds:

$$J^{(h)} \leq \frac{1}{2}\sum_{\tau=0}^{k}\langle y_\tau^{(h)}, y_\tau^{(h)}\rangle + \langle \hat{y}_\tau^{(h)}, \hat{y}_\tau^{(h)}\rangle + 2\left(\langle y_\tau^{(h)}, y_\tau^{(h)}\rangle\right)^{1/2}\left(\langle \hat{y}_\tau^{(h)}, \hat{y}_\tau^{(h)}\rangle\right)^{1/2}. \qquad \text{(A1.12)}$$

Breaking this inequality into three terms will simplify the development.

$$\begin{aligned} J^{(h)} &\leq J^{(h),1} + J^{(h),2} + J^{(h),3} \\ J^{(h),1} &:= \frac{1}{2}\sum_{\tau=0}^{k}\langle y_\tau^{(h)}, y_\tau^{(h)}\rangle \\ J^{(h),2} &:= \frac{1}{2}\sum_{\tau=0}^{k}\langle \hat{y}_\tau^{(h)}, \hat{y}_\tau^{(h)}\rangle \\ J^{(h),3} &:= \sum_{\tau=0}^{k}\left(\langle y_\tau^{(h)}, y_\tau^{(h)}\rangle\right)^{\frac{1}{2}}\left(\langle \hat{y}_\tau^{(h)}, \hat{y}_\tau^{(h)}\rangle\right)^{\frac{1}{2}} \end{aligned} \qquad \text{(A1.13)}$$

The process for showing that the first two terms, $J^{(h),1}$ and $J^{(h),2}$, are convergent is identical for both terms, and so this appendix will only show that the first and third terms, $J^{(h),1}$ and $J^{(h),3}$, are convergent. Using (A1.9), the product $\langle y_k^{(h)}, y_k^{(h)} \rangle$ becomes:

$$\langle y_k^{(h)}, y_k^{(h)} \rangle = \left(CM_c \exp\{\Lambda_c t_k\} M_c^{-1} x(0)\right)^T R_k^{-1} \left(CM_c \exp\{\Lambda_c t_k\} M_c^{-1} x(0)\right) \qquad \text{(A1.14)}$$

Let $\lambda_c < 0$ be the real part of the slowest decaying eigenvalue of Λ_c, and let 1_{n_s} be a unitary column vector of length n_s (all elements equal to one). Finally, let $a_c > 0$ be chosen such that $a_c \geq \left(C1_{n_s}\right)^T R_0^{-1} \left(C1_{n_s}\right)$. Note that the actual (fixed) value for a_c does not need to be known; the only requirement is that a_c exists and is finite. Then the following inequalities hold:

$$\begin{aligned} \langle y_k^{(h)}, y_k^{(h)} \rangle &\leq a_c \exp\{2\lambda_c t_k\}, \\ J^{(h),1} &\leq \frac{a_c}{2} \sum_{\tau=0}^{k} \exp\{2\lambda_c t_\tau\}. \end{aligned} \qquad \text{(A1.15)}$$

Assuming a uniform sampling period $\Delta t = t_k - t_{k-1}$, the last inequality in (A1.15) can be written:

$$J^{(h),1} \leq \frac{a_c}{2\Delta t} \sum_{\tau=0}^{k} \exp\{2\lambda_c t_\tau\} \Delta t < \frac{a_c}{2\Delta t} \int_{t_{-1}}^{t_k} \exp\{2\lambda_c \tau\} \, d\tau. \qquad \text{(A1.16)}$$

The fact that the last inequality in (A1.16) holds can be seen from Figure A1.1. For a more formal proof, see Theorem 123 on pages 600-601 of (Garner, 2002). Therefore, if the integral in (A1.16) has a well defined solution in the limit as $k \to \infty$, the series in (A1.16) also converges to a finite number that provides an upper bound for $J^{(h),1}$. Noting that $t_{-1} = -\Delta t$, the limiting solution to the integral in (A1.16) is as follows:

$$\lim_{k\to\infty}\frac{a_c}{2\Delta t}\int_{t_{-1}}^{t_k}\exp\{2\lambda_c\tau\}\,d\tau = -\frac{a_c}{4\lambda_c\Delta t}\exp\{-2\lambda_c\Delta t\}. \tag{A1.17}$$

The limit exists and is finite, and thus $J^{(h),1}$ also converges to a finite value.

The process for showing that $J^{(h),2}$ also converges to a finite value is identical to the argument for $J^{(h),1}$, and so it is not shown here. Next, this appendix shows that $J^{(h),3}$ is convergent. This development has already shown that $\langle y_k^{(h)}, y_k^{(h)}\rangle \le a_c\exp\{2\lambda_c t_k\}$. By a similar argument, $\langle \hat{y}_k^{(h)}, \hat{y}_k^{(h)}\rangle \le a\exp\{2\lambda t_k\}$. Here $\lambda < 0$ is the real part of the slowest decaying eigenvalue of Λ, and $a > 0$ satisfies $a \ge \left(\hat{C}\mathbb{P}(\xi)1_{n_s r}\right)^T R_0^{-1}\left(\hat{C}\mathbb{P}(\xi)1_{n_s r}\right)$ for all possible realizations of ξ (this assumes that $\hat{C}\mathbb{P}(\xi)$ is finite for all ξ). Then the following inequality holds:

$$\begin{aligned}
J^{(h),3} &\le \sum_{\tau=0}^{k}(a_c\exp\{2\lambda_c t_\tau\})^{\frac{1}{2}}(a\exp\{2\lambda t_\tau\})^{\frac{1}{2}} \\
&= \sum_{\tau=0}^{k}\left(\sqrt{a_c}\exp\{\lambda_c t_\tau\}\right)\left(\sqrt{a}\exp\{\lambda t_\tau\}\right) \\
&= \frac{\sqrt{a_c a}}{\Delta t}\sum_{\tau=0}^{k}(\exp\{(\lambda_c+\lambda)t_\tau\})\Delta t < \frac{\sqrt{a_c a}}{\Delta t}\int_{t_{-1}}^{t_k}\exp\{(\lambda_c+\lambda)\tau\}\,d\tau
\end{aligned} \tag{A1.18}$$

In the limit as $k \to \infty$, the integral in (A1.18) is finite, thus bounding the term $J^{(h),3}$. Therefore the purely homogenous part $J^{(h)}$ of the objective function converges to a finite value as time progresses. The goal of this appendix now is to show that $J^{(h,f)}$ also converges to a finite limit in time. The function $J^{(h,f)}$ can be written in terms of the inner product (defined above) as follows:

$$J^{(h,f)} = \frac{1}{2}\sum_{\tau=0}^{k}\langle y_\tau^{(h)} - \hat{y}_\tau^{(h)}, y_\tau^{(f)} - \hat{y}_\tau^{(f)}\rangle \tag{A1.19}$$

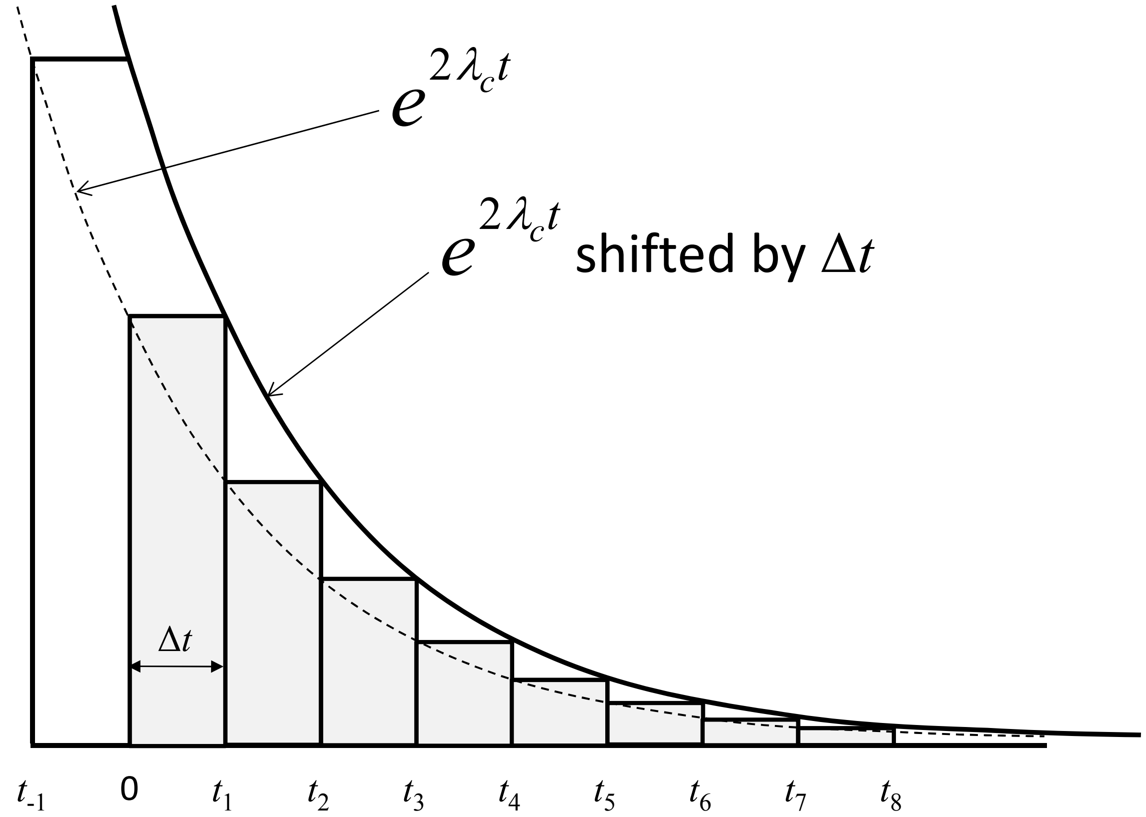


Figure A1.1: The area under $exp\{2\lambda_c t\}$ from t_{-1} to t_k is greater than $\sum_{\tau=0}^{k} \exp\{2\lambda_c t_\tau\}\Delta t$.

An asymptotically stable, linear time-invariant system with a bounded input signal has a bounded output response (see pages 121-132 of (Chen, 1999), also note that this assumes that the noise sequence v_k is bounded). Let $b_c > 0$ and $b > 0$ be the bounds on the outputs $y_k^{(f)}$ and $\hat{y}_k^{(f)}$ respectively, *i.e.* $|y_k^{(f)}| \leq b_c$ and $|\hat{y}_k^{(f)}| \leq b$. Then the following inequality holds:

$$\left|J^{(h,f)}\right| \leq \frac{1}{2}\sum_{\tau=0}^{k} \langle \left|y_\tau^{(h)}\right| + \left|\hat{y}_\tau^{(h)}\right|, \left|y_\tau^{(f)}\right| + \left|\hat{y}_\tau^{(f)}\right| \rangle = \frac{(b_c + b)}{2}\sum_{\tau=0}^{k}\left|y_\tau^{(h)}\right| + \left|\hat{y}_\tau^{(h)}\right| \qquad \text{(A1.20)}$$

In (A1.20), $\left|y_k^{(h)}\right| \leq \sqrt{a_c}\exp\{\lambda_c t_k\}$ and $\left|\hat{y}_k^{(h)}\right| \leq \sqrt{a}\exp\{\lambda t_k\}$ using the arguments and definitions above. Hence, $\left|J^{(h,f)}\right|$ satisfies the following inequality:

$$\begin{aligned}\left|J^{(h,f)}\right| &\leq \frac{(b_c+b)}{2}\sum_{\tau=0}^{k}\sqrt{a_c}\exp\{\lambda_c t_\tau\}+\sqrt{a}\exp\{\lambda t_\tau\} \\ &\leq \frac{(b_c+b)}{2\Delta t}\left(\sqrt{a_c}\int_{t_{-1}}^{t_k}\exp\{\lambda_c\tau\}\,d\tau+\sqrt{a}\int_{t_{-1}}^{t_k}\exp\{\lambda\tau\}\,d\tau\right)\end{aligned} \tag{A1.21}$$

Both integrals are bounded as $k \to \infty$ and thus $J^{(h,f)}$ is also bounded.

For the final part of the proof, this appendix shows that the expected value of the objective function, $E\{\lim_{k\to\infty} J_k(\hat{\xi}_k)\}$, tends to infinity. Then the effect of the homogenous response, and hence the effect of the unknown initial conditions, becomes negligible. The expected cost $E\{J_k(\hat{\xi}_k)\}$ can be written as follows:

$$\begin{aligned}E\{J_k(\hat{\xi}_k)\} &= E\{J_{k-1}(\hat{\xi}_k)\}+E\{D_k(\hat{\xi}_k)\}, \\ D_k(\hat{\xi}_k) &:= \langle y_k-\hat{y}_k(\hat{\xi}_k), y_k-\hat{y}_k(\hat{\xi}_k)\rangle\end{aligned} \tag{A1.22}$$

This appendix assumes that $E\{D_{k+1}(\hat{\xi}_{k+1})\} \geq E\{D_k(\hat{\xi}_k)\}$ based on the following intuition. The estimate $\hat{\xi}_{k+1} = \operatorname{argmin}_\xi J_{k+1}(\xi)$ has all of the constraints that $\hat{\xi}_k = \operatorname{argmin}_\xi J_k(\xi)$ has plus one more because $\hat{\xi}_{k+1}$ must minimize $J_k(\xi)+D_{k+1}(\xi)$. Therefore, the set of all possible realizations of ξ that minimize $E\{D_{k+1}(\hat{\xi}_{k+1})\}$ is smaller than the set of all possible realizations of ξ that minimize $E\{D_k(\hat{\xi}_k)\}$. *I.e.*, $\hat{\xi}_{k+1}$ has less flexibility to minimize$E\{D_{k+1}(\hat{\xi}_{k+1})\}$ than $\hat{\xi}_k$ has to minimize $E\{D_k(\hat{\xi}_k)\}$, therefore,

$$E\{D_{k+1}(\hat{\xi}_{k+1})\} \geq E\{D_k(\hat{\xi}_k)\}. \tag{A1.23}$$

Equality in (A1.23), *i.e.* $E\{D_{k+1}(\hat{\xi}_{k+1})\} = E\{D_k(\hat{\xi}_k)\}$, is only possible when $\hat{\xi}_{k+1} = \hat{\xi}_k$. In that case:

$$E\{D_{k+1}(\hat{\xi}_{k+1})\} = E\{D_{k+1}(\hat{\xi}_k)\} \geq n_y. \tag{A1.24}$$

The proof of the inequality in (A1.24) is as follows. By assumption, the noise sequence $v_\tau,\ \tau = 0,\dots,k$ is independent of the dynamics of the system. The estimate $\hat{\xi}_k$

(and hence $\hat{y}_k$) is a function of $v_\tau,\ \tau = 0, \dots, k$ since it minimizes $J_k(\xi)$, but v_{k+1} and $\hat{\xi}_k$ are mutually independent. Let $\hat{y}_{k+1}(\hat{\xi}_k) = y_k^{(h)} + y_k^{(f)} + \Delta_\mathrm{k}(\hat{\xi}_k)$ where $\Delta_\mathrm{k}(\hat{\xi}_k) \coloneqq \hat{y}_{k+1}(\hat{\xi}_k) - \left(y_k^{(h)} + y_k^{(f)}\right)$. The following holds:

$$\begin{aligned} &E\{\langle y_{k+1} - \hat{y}_{k+1}(\hat{\xi}_k), y_{k+1} - \hat{y}_{k+1}(\hat{\xi}_k)\rangle\} \\ &\qquad = E\{\langle v_{k+1} - \Delta_\mathrm{k}, v_{k+1} - \Delta_\mathrm{k}\rangle\} \\ &\qquad = E\{\langle v_{k+1}, v_{k+1}\rangle\} + E\{\langle \Delta_\mathrm{k}, \Delta_\mathrm{k}\rangle\} - 2\langle E\{v_{k+1}\}, E\{\Delta_\mathrm{k}\}\rangle \\ &\qquad = E\{\langle v_{k+1}, v_{k+1}\rangle\} + E\{\langle \Delta_\mathrm{k}, \Delta_\mathrm{k}\rangle\} \\ &\qquad = n_y + E\{\langle \Delta_\mathrm{k}, \Delta_\mathrm{k}\rangle\} \\ &\qquad \geq n_y. \end{aligned} \tag{A1.25}$$

The second equality is satisfied because v_{k+1} and $\hat{y}_{k+1}(\hat{\xi}_k)$ are independent and hence v_{k+1} and $\Delta_\mathrm{k}(\hat{\xi}_k)$ are independent. The third equality holds because $E\{v_{k+1}\} = 0$. The fourth equality holds because $E\{\langle v_{k+1}, v_{k+1}\rangle\} = n_y$ (see Appendix B of (Seghers)). The inequality follows from the definition of the inner product, and equality is only satisfied when $\Delta_\mathrm{k} = 0$.

This appendix has shown that $E\{D_{k+1}(\hat{\xi}_{k+1})\} \geq \min\,(n_y, \delta_k + E\{D_k(\hat{\xi}_k)\})$ where $\delta_k > 0$ is defined as $\delta_k \coloneqq E\{D_{k+1}(\hat{\xi}_{k+1})\} - E\{D_k(\hat{\xi}_k)\}$. It follows from the previous sentence that $E\{D_{k+1}(\hat{\xi}_{k+1})\} > 0$ for $k \geq 1$. Now this appendix will show that $E\{J_{k+1}(\hat{\xi}_{k+1})\} \geq E\{\sum_{\tau=0}^{k+1} D_\tau(\hat{\xi}_\tau)\}$, and because $E\{D_{k+1}(\hat{\xi}_{k+1})\} \geq E\{D_k(\hat{\xi}_k)\} > 0$ for $k \geq 1$, $E\{\lim_{k\to\infty} J_k(\hat{\xi}_k)\}$ tends to infinity.

First consider $E\{J_1(\hat{\xi}_1)\}$:

$$E\{J_1(\hat{\xi}_1)\} = E\{D_0(\hat{\xi}_1)\} + E\{D_1(\hat{\xi}_1)\} \geq E\{D_0(\hat{\xi}_0)\} + E\{D_1(\hat{\xi}_1)\} \tag{A1.26}$$

The inequality follows from the fact that $E\{D_0(\hat{\xi}_0)\} = E\{J_0(\hat{\xi}_0)\} \leq E\{J_0(\hat{\xi}_1)\}$ since $\hat{\xi}_0 = \operatorname{argmin}_\xi J_0(\xi)$. Now consider $E\{J_2(\hat{\xi}_2)\}$:

$$\begin{aligned} E\{J_2(\hat{\xi}_2)\} &= E\{D_0(\hat{\xi}_2)\} + E\{D_1(\hat{\xi}_2)\} + E\{D_2(\hat{\xi}_2)\} \\ &\geq E\{J_1(\hat{\xi}_1)\} + E\{D_2(\hat{\xi}_2)\} \\ &\geq E\{D_0(\hat{\xi}_0)\} + E\{D_1(\hat{\xi}_1)\} + E\{D_2(\hat{\xi}_2)\} \end{aligned} \tag{A1.27}$$

The first inequality follows from the fact that $E\{J_1(\hat{\xi}_1)\} \leq E\{J_1(\hat{\xi}_2)\}$. The second inequality follows from (A1.26). Continuing this process leads to the desired result: $E\{J_{k+1}(\hat{\xi}_{k+1})\} \geq E\{\sum_{\tau=0}^{k+1} D_\tau(\hat{\xi}_\tau)\}$ where $E\{D_{k+1}(\hat{\xi}_{k+1})\} \geq E\{D_k(\hat{\xi}_k)\} > 0$. Thus $E\{\lim_{k\to\infty} J_k(\hat{\xi}_k)\}$ tends to infinity, and the proof is complete.

In summary, this appendix has shown that the effect of initial conditions on the parameter estimates becomes negligible as time progresses to infinity under certain assumptions. This result relaxes the requirement that the initial conditions must either be known or estimated by the algorithm.

This appendix outlines a proof for linear time-invariant state equations. The main observation that made the proof possible is the fact that the effect of initial conditions on the system's output response eventually becomes negligible. Based on this observation, this dissertation conjectures that the results of the above proof will apply for any dynamic system whose unforced response eventually becomes negligible. In other words, this appendix conjectures that for all asymptotically stable systems, the effect of initial conditions on the parameter estimates obtained via polynomial chaos estimation will become negligible.

Appendix A2

Implementation of Filters for Sprung Mass Estimation

This appendix discusses the implementation of the filtering methods that this dissertation uses for vehicle mass estimation. As Example (1) of Chapter 4 suggests, the filtering algorithms may in some cases be more difficult to tune than the polynomial chaos estimators of this dissertation. This appendix first summarizes the filtering algorithms used by this dissertation and then discusses how the algorithms were tuned in order to produce the results presented in the experimental sections of this dissertation.

Consider the following continuous-time state equation having a discrete-time output equation:

$$\begin{gathered}\dot{x}(t) = f\big(t, x(t), u(t), w(t)\big) \\ y_k = h_k(x(t_k), u(t_k), v_k)\end{gathered} \tag{A2.1}$$

The state vector $x(t) \in \mathbb{R}^{n_s}$ has been extended to include the unknown parameters. The input vector is $u(t) \in \mathbb{R}^{n_u}$ and is assumed to be known at time t, and the white process noise $w(t) \in \mathbb{R}^{n_s}$ is zero mean with covariance matrix $Q(t) \in \mathbb{R}^{n_s \times n_s}$. The discrete-time output vector $y_k \in \mathbb{R}^{n_y}$ is distorted by the measurement noise sequence $v_k \in \mathbb{R}^{n_y}$ which is zero mean with covariance matrix $R_k \in \mathbb{R}^{n_y \times n_y}$.

A2.1 Hybrid Extended Kalman Filtering

The hybrid Extended Kalman Filter (EKF) algorithm (see page 405 of (Simon, 2006)) is as follows. Integrate the following equations from time t_{k-1} to t_k to predict the state mean $\hat{x}_{k|k-1} = \hat{x}(t_k)$ and covariance $P_{k|k-1} = P(t_k) \in \mathbb{R}^{n_s \times n_s}$.

$$\begin{aligned} \dot{\hat{x}} &= f(t, \hat{x}, u, 0) \\ \dot{P} &= FP + PF^T + LQL^T \end{aligned} \tag{A2.2}$$

At time t_{k-1} the initial conditions are set as $\hat{x}(t_{k-1}) = \hat{x}_{k-1|k-1}$ and $P(t_{k-1}) = P_{k-1|k-1}$ (at time t_0, the initial conditions are $\hat{x}(t_0) = E\{x(t_0)\}$ and $P(t_0) = E\left\{\big(x(t_0) - \hat{x}(t_0)\big)\big(x(t_0) - \hat{x}(t_0)\big)^T\right\}$). The matrices $F \in \mathbb{R}^{n_s \times n_s}$ and $L \in \mathbb{R}^{n_s \times n_s}$ are defined as follows:

$$\begin{aligned} F &:= \left.\frac{\partial f}{\partial x}\right|_{\hat{x}(t), u(t)} \\ L &:= \left.\frac{\partial f}{\partial w}\right|_{\hat{x}(t), u(t)}. \end{aligned} \tag{A2.3}$$

Once the measurement y_k is available, the Kalman gain $K_k \in \mathbb{R}^{n_s \times n_y}$ is calculated, and the predictions of the state mean and covariance are updated as follows:

$$\begin{aligned} K_k &= P_{k|k-1} H_k^T \big(H_k P_{k|k-1} H_k^T + M_k R_k M_k^T\big)^{-1} \\ \hat{x}_{k|k} &= \hat{x}_{k|k-1} + K_k \left(y_k - h_k\big(\hat{x}_{k|k-1}, u(t_k), 0\big)\right) \\ P_{k|k} &= (I - K_k H_k) P_{k|k-1} (I - K_k H_k)^T + K_k M_k R_k M_k^T K_k^T \end{aligned} \tag{A2.4}$$

The matrices $H_k \in \mathbb{R}^{n_y \times n_s}$ and $M_k \in \mathbb{R}^{n_y \times n_y}$ are defined as follows:

$$H_k := \left.\frac{\partial h_k}{\partial x}\right|_{\hat{x}_{k|k-1}, u(t_k)} \tag{A2.5}$$

$$M_k := \left.\frac{\partial h_k}{\partial v_k}\right|_{\hat{x}_{k|k-1},u(t_k)}$$

Equations (A2.2) through (A2.5) have summarized the hybrid EKF algorithm. In the vehicle mass estimation problems of this dissertation, the process and measurement noise sequences are assumed additive, so L and M_k are identity matrices. For the reduced full car model of Equations (7.29) and (7.30), the F and H_k are found by analytically calculating the partial derivatives of the functions of Equations (7.29) and (7.30) respectively:

$$F = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{2(k_f + k_r)}{\hat{x}_3} & -\frac{\hat{x}_4}{\hat{x}_3} & \frac{2(k_f + k_r)\hat{x}_1 + \hat{x}_4\hat{x}_2}{\hat{x}_3{}^2} & -\frac{\hat{x}_2}{\hat{x}_3} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{A2.6}$$

$$H_k = \begin{bmatrix} -\frac{2(k_f + k_r)}{\hat{x}_3} & -\frac{\hat{x}_4}{\hat{x}_3} & \frac{2(k_f + k_r)\hat{x}_1 + \hat{x}_4\hat{x}_2}{\hat{x}_3{}^2} & -\frac{\hat{x}_2}{\hat{x}_3} \end{bmatrix} \tag{A2.7}$$

For the half-car model with the augmented equations of (7.31) and (7.32), the Jacobian matrices F and H_k are given as follows noting that $F = \left[F^{(i,j)}\right]$

$$F = \left[\begin{matrix} 0 & 1 & 0 & 0 \\ -\left(\frac{a^2}{J_\phi} + \frac{1}{\hat{x}_5}\right)k_f & -\left(\frac{a^2}{J_\phi} + \frac{1}{\hat{x}_5}\right)\hat{x}_6 & \left(\frac{ab}{J_\phi} - \frac{1}{\hat{x}_5}\right)k_r & \left(\frac{ab}{J_\phi} - \frac{1}{\hat{x}_5}\right)\hat{x}_7 \\ 0 & 0 & 0 & 1 \\ \left(\frac{ab}{J_\phi} - \frac{1}{\hat{x}_5}\right)k_f & \left(\frac{ab}{J_\phi} - \frac{1}{\hat{x}_5}\right)\hat{x}_6 & -\left(\frac{b^2}{J_\phi} + \frac{1}{\hat{x}_5}\right)k_r & -\left(\frac{b^2}{J_\phi} + \frac{1}{\hat{x}_5}\right)\hat{x}_7 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{matrix}\right. \cdots$$

$$\left.\begin{matrix} 0 & 0 & 0 \\ \dfrac{k_f\hat{x}_1 + \hat{x}_6\hat{x}_2 + k_r\hat{x}_3 + \hat{x}_7\hat{x}_4}{\hat{x}_5^2} & -\left(\dfrac{a^2}{J_\phi} + \dfrac{1}{\hat{x}_5}\right)\hat{x}_2 & \left(\dfrac{ab}{J_\phi} - \dfrac{1}{\hat{x}_5}\right)\hat{x}_4 \\ 0 & 0 & 0 \\ \dfrac{k_f\hat{x}_1 + \hat{x}_6\hat{x}_2 + k_r\hat{x}_3 + \hat{x}_7\hat{x}_4}{\hat{x}_5^2} & \left(\dfrac{ab}{J_\phi} - \dfrac{1}{\hat{x}_5}\right)\hat{x}_2 & -\left(\dfrac{b^2}{J_\phi} + \dfrac{1}{\hat{x}_5}\right)\hat{x}_4 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{matrix}\right]$$

$$H_k = \left[\left(\frac{ab^2 - a^3}{J_\phi L} - \frac{1}{\hat{x}_5}\right)k_f \quad \left(\frac{ab^2 - a^3}{J_\phi L} - \frac{1}{\hat{x}_5}\right)\hat{x}_6 \quad \left(\frac{a^2b - b^3}{J_\phi L} - \frac{1}{\hat{x}_5}\right)k_r \ldots\right.$$

$$\left(\frac{a^2b - b^3}{J_\phi L} - \frac{1}{\hat{x}_5}\right)\hat{x}_7 \quad \frac{k_f\hat{x}_1 + \hat{x}_6\hat{x}_2 + k_r\hat{x}_3 + \hat{x}_7\hat{x}_4}{\hat{x}_5^2} \quad \left(\frac{ab^2 - a^3}{J_\phi L} - \frac{1}{\hat{x}_5}\right)\hat{x}_2 \cdots$$

$$\left.\left(\frac{a^2b - b^3}{J_\phi L} - \frac{1}{\hat{x}_5}\right)\hat{x}_4\right]$$

For the vehicle mass estimation problem, the EKF parameters $P(t_0)$, $\hat{x}(t_0)$, $Q(t)$, and R_k are unknown and are treated as user specified tuning parameters. Section A2.4 will discuss how these parameters were selected. The next section outlines the Unscented Kalman Filter (UKF).

A2.2 Unscented Kalman Filtering

The Unscented Kalman Filter (UKF) algorithm (see Chapter 14 of (Simon, 2006)) is provided in this section. This section assumes that the noise sequences $w(t)$ and v_k are additive, *i.e.*, $\dot{x} = f(t, x, u) + w$ and $y_k = h_k(x, u) + v_k$.

The UKF filter is initialized in the same way as the EKF: $\hat{x}_{0|0} = E\{x(t_0)\}$ and $P_{0|0} = E\left\{\left(x(t_0) - \hat{x}_{0|0}\right)\left(x(t_0) - \hat{x}_{0|0}\right)^T\right\}$. Then at time t_{k-1}, $2n_s + 1$ "sigma points" $x^{(i)} \in \mathbb{R}^{n_s}$ are chosen as follows:

$$
\begin{aligned}
x_{k-1}^{(0)} &= \hat{x}_{k-1|k-1} \\
x_{k-1}^{(i)} &= \hat{x}_{k-1|k-1} + \tilde{x}^{(i)} \\
\tilde{x}^{(i)} &= \left(\sqrt{(n_s+\kappa)P_{k-1|k-1}}\right)_i^T, \quad i = 1, \dots, n_s \\
\tilde{x}^{(n_s+i)} &= -\left(\sqrt{(n_s+\kappa)P_{k-1|k-1}}\right)_i^T, \quad i = 1, \dots, n_s.
\end{aligned}
\tag{A2.8}
$$

The operator $\sqrt{\cdot}$ in (A2.8) is the matrix square root operator such that $\left(\sqrt{P}\right)^T\sqrt{P} = P$ where P is a square matrix, and $\left(\sqrt{\cdot}\right)_i$ is the i^{th} row of the matrix square root. A set of $2n_s + 1$ weighting coefficients corresponding to the sigma points are chosen as follows:

$$
\begin{aligned}
W^{(0)} &= \frac{\kappa}{n_s + \kappa} \\
W^{(i)} &= \frac{1}{2(n_s+\kappa)}, \qquad i = 1, \dots, 2n_s.
\end{aligned}
\tag{A2.9}
$$

The variable $\kappa \neq -n_s$ is a design choice, and $\kappa = 3 - n_s$ is optimal in some sense if x is Gaussian (see page 454 of (Simon, 2006)). The prediction step integrates the following state equations from time t_{k-1} to time t_k to calculate the transformed sigma points $x_k^{(i)}$:

$$
\dot{x}^{(i)} = f\left(t, x^{(i)}, u\right), \qquad i = 0, \dots, 2n_s \tag{A2.10}
$$

Equation (A2.10) is initialized at time t_{k-1} by $x^{(i)}(t_{k-1}) = x_{k-1}^{(i)}$. The predicted mean $\hat{x}_{k|k-1}$, output $\hat{y}_k$, and output covariance $P_y \in \mathbb{R}^{n_y \times n_y}$ are given as follows:

$$
\begin{aligned}
\hat{x}_{k|k-1} &= \sum_{i=0}^{2n_s} W^{(i)} x_k^{(i)} \\
y^{(i)} &:= h_k\left(x_k^{(i)}, u(t_k)\right)
\end{aligned}
\tag{A2.11}
$$

$$\hat{y}_k = \sum_{i=0}^{2n_s} W^{(i)} y^{(i)}$$

$$P_y = \sum_{i=0}^{2n_s} W^{(i)} \left(y^{(i)} - \hat{y}_k\right)\left(y^{(i)} - \hat{y}_k\right)^T + R_k.$$

The cross covariance $P_{xy} \in \mathbb{R}^{n_s \times n_y}$ is also estimated as a function of the sigma points:

$$P_{xy} = \sum_{i=0}^{2n_s} W^{(i)} \left(x_k^{(i)} - \hat{x}_{k|k-1}\right)\left(y^{(i)} - \hat{y}_k\right)^T. \tag{A2.12}$$

Finally the following equations calculate the Kalman gain K_k and update the predictions of the state mean and covariance:

$$K_k = P_{xy} P_y^{-1}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(y_k - \hat{y}_k)$$

$$P_{k|k-1} \coloneqq \sum_{i=0}^{2n_s} W^{(i)} \left(x_k^{(i)} - \hat{x}_{k|k-1}\right)\left(x_k^{(i)} - \hat{x}_{k|k-1}\right)^T + Q_{k-1} \tag{A2.13}$$

$$P_{k|k} = P_{k|k-1} - K_k P_y K_k^T.$$

Equations (A2.8) through (A2.13) have summarized the UKF algorithm. To use the UKF algorithm, the dissertation work had to evaluate $2n_s + 1$ sets of state equations and compute the matrix square root given in Equation (A2.8) at each time interval. This was found to be significantly more computationally burdensome than the EKF algorithm and even than the proposed polynomial chaos algorithms (in the Matlab® environment without optimizing the code for computational speed). Sections 14.4.2 and 14.4.3 of (Simon, 2006) suggest alternative options for selecting sigma points that minimize the computational burden.

As with the EKF, the UKF parameters $P(t_0)$, $\hat{x}(t_0)$, $Q(t)$, and R_k are unknown and are treated as user specified tuning parameters. Appendix section A2.4 will discuss how these parameters were selected.

A2.3 Particle Filtering

In addition to the EKF and UKF, this dissertation explored using a particle filter for vehicle mass estimation (see Chapter 15 of (Simon, 2006)). The conclusion, however, was that no visible benefits were achieved by using the particle filter compared with the EKF, UKF, and proposed polynomial chaos algorithms, and the particle filter resulted in the highest computational burden (for non-optimized code in the Matlab® environment). This result was not surprising because the particle filter finds its greatest advantages over Kalman filtering algorithms in highly nonlinear estimation problems with non-Gaussian disturbance sources (Arulampalam et al., 2002).

The particle filtering algorithm used in this dissertation is presented in this section. To initialize the filter, N "particles" - *i.e.* estimates $x_{0|0}^{(i)} \in \mathbb{R}^{n_s},\ i = 1, \ldots, N$ of the state vector - are drawn randomly from the prior distribution $p\big(x(t_0)\big)\colon \mathbb{R}^{n_s} \to \mathbb{R}$. Each of these particles are assigned a weight $W^{(i)} = 1/N$.

The following state equations are integrated from time t_{k-1} to t_k to predict the new values of the particles $x_{k|k-1}^{(i)}$:

$$\dot{x}^{(i)} = f\big(t, x^{(i)}, u\big) + \widetilde{w}^{(i)}, \qquad i = 0, \ldots, N. \tag{A2.14}$$

Equation (A2.14) is initialized at time t_{k-1} by $x_{k-1|k-1}^{(i)}$. This step of the particle filter is similar to one of the steps in the UKF algorithm. However, there are two important differences: (a) The particles are randomly selected whereas the sigma points of the UKF are chosen in a deterministic manner. (b) The particle filter simulates the process noise when evaluating the state equation (A2.14) whereas the UKF does not. This dissertation simulated the process noise $\widetilde{w}^{(i)} \in \mathbb{R}^{n_s},\ i = 1, \dots, N$ ($\widetilde{w}^{(i)}$ was assumed constant over the interval t_{k-1} to t_k) by randomly sampling from the Gaussian distribution with zero mean and covariance $Q(t)$ (whose true value is unknown, further discussion is in Section A2.4).

Evaluating (A2.14) from time t_{k-1} to t_k results in an updated set of particles $x_{k|k-1}^{(i)}$. The prediction of the state mean $\hat{x}_{k|k-1}$ is computed as a function of the particles and their weights:

$$\hat{x}_{k|k-1} = \sum_{i=1}^{N} W^{(i)} x_{k|k-1}^{(i)} \tag{A2.15}$$

Once the output measurement y_k is available, each weight $W^{(i)}$ is updated based on the likelihood of its corresponding particle:

$$\begin{gathered} W^{(i)} \propto \mathcal{L}\left\{x_{k|k-1}^{(i)} | y_k\right\} \\ \sum_{i=1}^{N} W^{(i)} = 1. \end{gathered} \tag{A2.16}$$

For zero mean, additive Gaussian noise v_k, Equation (A2.16) becomes:

$$\begin{gathered} W^{(i)} \propto exp\left\{-\frac{1}{2}\left(y_k - h_k\left(x_{k|k-1}^{(i)}, u(t_k)\right)\right)^T R_k^{-1}\left(y_k - h_k\left(x_{k|k-1}^{(i)}, u(t_k)\right)\right)\right\} \\ \sum_{i=1}^{N} W^{(i)} = 1. \end{gathered} \tag{A2.17}$$

The particle filter used in this dissertation updates the set of particles at each time step using resampling. With resampling, particles with greater weights, *i.e.* higher likelihood, are more likely to be sampled and reused as initialization particles for the next time step. This resampling step is sometimes referred to as “Survival of the Fittest”. Particles with lower likelihood are likely to be replaced by particles with higher likelihood. Resampling is carried out in the following three steps (see page 467 of (Simon, 2006)): (a) Randomly sample a number $\gamma \in [0,1]$ such that γ is uniformly distributed on [0,1]. (b) Find the integer j such that $\sum_{l=1}^{j-1} W^{(l)} < r$ but $\sum_{l=1}^{j} W^{(l)} \geq r$. Then set the updated particle as $x_{k|k}^{(i)} = x_{k|k-1}^{(j)}$. (c) Repeat steps (a) and (b) for $i = 1, \dots, N$. Finally, reset each weight as $W^{(i)} = 1/N$. Then the updated state estimate at time t_k is as follows:

$$\hat{x}_{k|k} = \sum_{i=1}^{N} W^{(i)} x_{k|k}^{(i)}. \tag{A2.18}$$

This appendix section has summarized the particle filter used by this dissertation. As with the other filtering methods, the parameters $P(t_0)$, $\hat{x}(t_0)$, $Q(t)$, and R_k are unknown and are treated as user specified tuning parameters. The following section will discuss how these parameters were selected for the vehicle mass estimation problem.

A2.4 Tuning the Filtering Algorithms

In the vehicle mass estimation problem of this dissertation, the filtering parameters $P(t_0)$, $\hat{x}(t_0)$, $Q(t)$, and R_k are unknown. This section discusses how they were selected for the purposes of this dissertation. The intuition gained by this dissertation for tuning these algorithms may be helpful to similar problems, *i.e.* problems in which the main objective is parameter estimation, not state filtering.

The goal of the polynomial chaos estimators is to accurately estimate system parameters and not necessarily the system states. The assumptions underlying the polynomial chaos estimators may therefore provide insights into the tuning of the filtering algorithms when parameter estimation is the main objective.

A key assumption of the polynomial chaos estimators is that the uncertainty in the state equations is entirely due to the uncertainty in the system parameters. In other words, there is no state process noise, only output measurement noise; the model of the system dynamics is perfect except that some of the parameters are unknown. This same assumption can be captured by the filtering algorithms via a specific tuning of the initial state covariance $P(t_0)$ and process noise covariance $Q(t)$. Setting $Q(t) = 0$ is similar to saying that there is no process noise, and that the unknown parameters are static variables. Tuning the initial state covariance $P(t_0)$ to mimic the polynomial chaos algorithm is only slightly more complicated. Consider the extended state vector $x^e \in \mathbb{R}^{n_s+n_p}$ written explicitly in terms of the system states $x \in \mathbb{R}^{n_s}$ and unknown static parameters $\theta \in \mathbb{R}^{n_p}$:

$$x^e = \begin{bmatrix} x \\ \theta \end{bmatrix}. \tag{A2.19}$$

Then, assuming that the model of the system differs from the true system only by the unknown parameters, the initial state covariance $P(t_0)$ can be written as follows:

$$\begin{aligned} P(t_0) &= E\left\{\begin{bmatrix} x(t_0) - \hat{x}(t_0) \\ \theta - \hat{\theta}(t_0) \end{bmatrix} \begin{bmatrix} x^T(t_0) - \hat{x}^T(t_0) & \theta^T - \hat{\theta}^T(t_0) \end{bmatrix}\right\} \\ &= E\left\{\begin{bmatrix} 0 \\ \theta - \hat{\theta}(t_0) \end{bmatrix} \begin{bmatrix} 0^T & \theta^T - \hat{\theta}^T(t_0) \end{bmatrix}\right\}. \end{aligned} \tag{A2.20}$$

The above discussion has outlined a procedure for selecting $P(t_0)$ and $Q(t)$; unfortunately, the polynomial chaos based estimators do not provide insight into selecting the value of R_k, even if it is constant. As shown in Chapter 4, this is one of the reasons that the polynomial chaos algorithm is easier to tune in some situations than the filtering algorithms.

Table A2.1 shows the values of the tuning parameters that were used in the experiments of this dissertation. The term ε is $\varepsilon = 2.22 \cdot 10^{-16}$.

Table A2.1: Filter parameters for the EKF, UKF, and Particle filters.

	$\boldsymbol{Q(t)}$	$\boldsymbol{R_k}$	$\boldsymbol{P(t_0)}$	$\boldsymbol{x(t_0)}$
EKF	$\begin{bmatrix} \varepsilon & 0 & 0 & 0 \\ 0 & \varepsilon & 0 & 0 \\ 0 & 0 & 0.2 & 0 \\ 0 & 0 & 0 & 0.2 \end{bmatrix}$	0.007	$5 \cdot 10^6 \times \begin{bmatrix} \varepsilon & 0 & 0 & 0 \\ 0 & \varepsilon & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 1500 \\ 16000 \end{bmatrix}$
UKF	$\begin{bmatrix} \varepsilon & 0 & 0 & 0 \\ 0 & \varepsilon & 0 & 0 \\ 0 & 0 & 0.008 & 0 \\ 0 & 0 & 0 & 0.008 \end{bmatrix}$	0.007	$3 \cdot 10^6 \times \begin{bmatrix} \varepsilon & 0 & 0 & 0 \\ 0 & \varepsilon & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 1500 \\ 16000 \end{bmatrix}$
Particle Filter	$\begin{bmatrix} \varepsilon & 0 & 0 & 0 \\ 0 & \varepsilon & 0 & 0 \\ 0 & 0 & 2.2 & 0 \\ 0 & 0 & 0 & 9.5 \end{bmatrix}$	0.007	$10^6 \times \begin{bmatrix} 10^{-8} & 0 & 0 & 0 \\ 0 & 10^{-8} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 2.5 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 1500 \\ 16000 \end{bmatrix}$

Bibliography

ANTIA, H. M. 2002. *Numerical Methods for Scientists and Engineers,* Boston, Birkhauser Verlag.

ARULAMPALAM, M. S., MASKELL, S., GORDON, N. & CLAPP, T. 2002. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing,* 50**,** 174-88.

BAE, H. S., RYU, J. & GERDES, J. C. Year. Road grade and vehicle parameter estimation for longitudinal control using GPS. *In:* 2001 IEEE Intelligent Transportation Systems Proceedings, August 25, 2001 - August 29, 2001, 2001 Oakland, CA, United states. Institute of Electrical and Electronics Engineers Inc., 166-171.

BELLINGER, S., KALILL, J. & TYLER, B. 2003. *System and method for estimating vehicle mass*. USA patent application.

BELLINGER, S. & SHUTTY, J. 2000. *System for controlling operation of an internal combustion engine*. USA patent application.

BEST, M. C. & GORDON, T. J. Year. Suspension system identification based on impulse-momentum equations. *In:* Dynamics of Vehicles on Roads and on Tracks. 15th IAVSD Symposium, 25-29 Aug. 1997, 1998 Netherlands. Swets &amp; Zeitlinger, 598-618.

BLANCHARD, E., SANDU, A. & SANDU, C. 2009. Parameter estimation for mechanical systems via an explicit representation of uncertainty. *Engineering Computations,* 26**,** 541-69.

BLANCHARD, E., SANDU, C. & SANDU, A. Year. A polynomial-chaos-based Bayesian approach for estimating uncertain parameters of mechanical systems. *In:* 19th Int. Conf. Design Theory and Methodology and 1st Int. Conf. Micro and Nano Systems, presented at - 2007 ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, IDETC/CIE2007, September 4, 2007 - September 7, 2007, 2008 Las Vegas, NV, United states. American Society of Mechanical Engineers, 1041-1048.

BLANCHARD, E., SANDU, C. & SANDU, A. Year. Comparison between a polynomial-chaos-based Bayesian approach and a polynomial-chaos-based EKF approach for parameter estimation with application to vehicle dynamics. *In:* 2009 ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, DETC2009, August 30, 2009 - September 2, 2009, 2010a San Diego, CA, United states. American Society of Mechanical Engineers, 893-904.

BLANCHARD, E. D., SANDU, A. & SANDU, C. 2010b. A Polynomial Chaos-Based Kalman Filter Approach for Parameter Estimation of Mechanical Systems. *Journal of Dynamic Systems, Measurement and Control,* 132**,** 061404 (18 pp.).

BREEN, M. T. 1996. *System and method for determining relative vehicle mass*. USA patent application.

CHEN, C.-T. 1999. *Linear System Theory and Design,* New York, Oxford University Press.

DRUZHININA, M., MOKLEGAARD, L. & STEFANOPOULOU, A. G. Year. Compression braking control for heavy-duty vehicles. *In:* Proceedings of 2000 American Control Conference (ACC 2000), 28-30 June 2000, 2000 Danvers, MA, USA. American Autom. Control Council, 2543-7.

DRUZHININA, M., STEFANOPOULOU, A. & MOKLEGAARD, L. 2002. Adaptive continuously variable compression braking control for heavy-duty vehicles. *Transactions of the ASME. Journal of Dynamic Systems, Measurement and Control,* 124**,** 406-14.

DU, H.-P. & ZHANG, N. 2010. Robust Active Suspension Design Subject to Vehicle Inertial Parameter Variations. *International Journal of Automation and Computing,* 7**,** 419-27.

DU, H., ZHANG, N. & LAM, J. 2008. Parameter-dependent input-delayed control of uncertain vehicle suspensions. *Journal of Sound and Vibration,* 317**,** 537-56.

DUTTA, P. & BHATTACHARYA, R. 2010. Nonlinear estimation of hypersonic state trajectories in Bayesian framework with polynomial chaos. *Journal of Guidance, Control, and Dynamics,* 33**,** 1765-1778.

ERIKSSON, A. 2009. *Implementation and Evaluation of a Mass Estimation Algorithm.* M.S., KTH - Royal Institute of Technology.

FATHY, H. K., DONGSOO, K. & STEIN, J. L. Year. Online vehicle mass estimation using recursive least squares and supervisory data extraction. *In:* 2008 American Control Conference (ACC '08), 11-13 June 2008, 2008 Piscataway, NJ, USA. IEEE, 1842-8.

FREMD, R. 1987. *Apparatus for measuring the mass of a motor vehicle*. USA patent application 874857.

GARNER, L. E. 2002. *Calculus,* Boston, MA, Pearson Custom Publishing.

GENISE, T. A. 1996. *Control method/system including determination of an updated value indicative of gross combination weight of vehicles*. USA patent application.

GERMANN, S. & ISERMANN, R. Year. Modelling and control of longitudinal vehicle motion. *In:* Proceedings of 1994 American Control Conference - ACC '94, 29 June-1 July 1994, 1994 New York, NY, USA. IEEE, 1-5.

GHANEM, R. G. & SPANOS, P. D. 1991. *Stochastic finite elements: A spectral approach,* New York, Springer-Verlag.

GRIESER, J. 2005. *Method for determining an estimate of the mass of a motor vehicle*. USA patent application.

GUBNER, J. A. 2006. *Probability and Random Processes for Electrical and Computer Engineers,* New York, Cambridge University Press.

HAC, A. B. 2009. *Dynamic Estimation of Vehicle Inertial Parameters and Tire Forces from Tire Sensors*. USA patent application.

HAN, K. J., KIM, I. K., JO, H. Y. & HUH, K. S. 2009. Development and experimental evaluation of an online estimation system for vehicle mass. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering,* 223**,** 167-176.

HAYAKAWA, K., OSAWA, M., YOSHIDA, H. & OSHIMA, M. 2002. *Vehicle mass calculation device*. USA patent application.

HUANG, J. & LIN, W. C. Year. EKF-based in-vehicle estimation of relative CG height. *In:* 2008 ASME Dynamic Systems and Control Conference, DSCC 2008, October 20, 2008 - October 22, 2008, 2009 Ann Arbor, MI, United states. ASME, 295-302.

HUH, K., LIM, S., JUNG, J., HONG, D., HAN, S., HAN, K., JO, J. H. & JIN, J. M. 2007. Vehicle Mass Estimator for Adaptive Roll Stability Control. *SAE Technical papers,* paper 2007-01-0820.

INMAN, D. J. 2001. *Engineering Vibration,* Upper Saddle River, NJ 07458, Prentice-Hall.

IOANNOU, P. A. & SUN, J. 1996. *Robust adaptive control,* Upper Saddle River, NJ 07458, Prentice-Hall.

KHALIL, H. K. 2002. *Nonlinear Systems,* Upper Saddle River, NJ 07458, Prentice-Hall, Inc.

KIM, C. & RO, P. I. 2000. Reduced-order modelling and parameter estimation for a quarter-car suspension system. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering,* 214**,** 851-864.

KLATT, A. 1985. *Method and aparatus to automatically determine the weight of mass of a moving vehicle*. USA patent application 557,789.

KUBUS, D., KROGER, T. & WAHL, F. M. Year. On-line estimation of inertial parameters using a recursive total least-squares approach. *In:* 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, 22-26 Sept. 2008, 2008 Piscataway, NJ, USA. IEEE, 3845-52.

LEE, H., PARK, K., HWANG, T., NOH, K., HEO, S.-J., JEONG, J. I., CHOI, S., KWAK, B. & KIM, S. 2009. Development of enhanced ESP system through vehicle parameter estimation. *Journal of Mechanical Science and Technology,* 23**,** 1046-1049.

LEIMBACH, K.-D., VEIL, H. & HUMMEL, S. 2001. *Method and system for determining a vehicle mass*. USA patent application 09/110,810.

LEIMBACH, K.-D., VEIL, H. & HUMMEL, S. 2002. *Process and device for determining a vehicle's mass*. USA patent application 09/254,351.

LI, J. & XIU, D. 2009. A generalized polynomial chaos based ensemble Kalman filter with high accuracy. *Journal of Computational Physics,* 228**,** 5454-69.

LIN, Y. & KORTUM, W. 1991. Identification of system physical parameters for vehicle systems with nonlinear components. *Vehicle System Dynamics,* 20**,** 354-365.

LINGMAN, P. & SCHMIDTBAUER, B. Year. Road slope and vehicle mass estimation using Kalman filtering. *In:* Dynamics of Vehicles on Roads and on Tracks. 17th IAVSD Symposium, 20-24 Aug. 2001, 2003 Netherlands. Swets &amp; Zeitlinger, 12-23.

MARZOUK, Y. & XIU, D. 2009. A Stochastic Collocation Approach to Bayesian Inference in Inverse Problems. *COMMUNICATIONS IN COMPUTATIONAL PHYSICS,* 6**,** 826-847.

MARZOUK, Y. M., NAJM, H. N. & RAHN, L. A. 2007. Stochastic spectral methods for efficient Bayesian solution of inverse problems. *Journal of Computational Physics,* 224**,** 560-86.

MASSEL, T., DING, E. L. & ARNDT, M. Year. Estimation of vehicle loading state. *In:* Proceedings of the 2004 IEEE International Conference on Control Applications, 2-4 Sept. 2004, 2004 Piscataway, NJ, USA. IEEE, 1260-5.

MOON, T. K. & STIRLING, W. C. 2000. *Mathematical methods and algorithms for signal processing,* Upper Saddle River, New Jersey 07458, Prentice-Hall.

OHSAKU, S. & NAKAI, H. 2000. *Sprung Mass Estimation Apparatus*. U.S.A. patent application 09/333287

PENCE, B. L., FATHY, H. K. & STEIN, J. L. Year. Sprung mass estimation for off-road vehicles via base-excitation suspension dynamics and recursive least squares. *In:* 2009 American Control Conference (ACC-09), 10-12 June 2009, 2009 Piscataway, NJ, USA. IEEE, 5043-8.

PENCE, B. L., FATHY, H. K. & STEIN, J. L. Year. A base -excitation approach to polynomial chaos-based estimation of sprung mass for off-road vehicles. *In:* 2009 ASME Dynamic Systems and Control Conference, DSCC2009, October 12, 2009 - October 14, 2009, 2010 Hollywood, CA, United states. American Society of Mechanical Engineers, 857-864.

PHILLIPS, D. H. & RICHARDSON, R. W. 1997. *System for measuring total weight and weight distribution of a vehicle*. USA patent application 615,449.

POULARIKAS, A. D. 1999. *The Handbook of Formulas and Tables for Signal Processing,* Boca Raton, CRC Press LLC.

PRIYANDOKO, G., MAILAH, M. & JAMALUDDIN, H. 2009. Vehicle active suspension system using skyhook adaptive neuro active force control. *Mechanical Systems and Signal Processing,* 23**,** 855-68.

RAJAMANI, R. 2006. *Vehicle Dynamics and Control,* New York, Springer Science+Business Media, Inc.

RAJAMANI, R. & HEDRICK, J. K. 1995. Adaptive observers for active automotive suspensions: theory and experiment. *IEEE Transactions on Control Systems Technology,* 3**,** 86-93.

REINER, K., RIEKER, H. & STOLL, J. 1990. *Device for determining the mass of a motor vehicle*. USA patent application 436028.

RIEKER, H., SCHUETZNER, P. & STOLL, J. 2002. *Device for determining the weight of a motor vehicle*. USA patent application 09/236,224.

RISTIC, B., MASKELL, S. & GORDON, N. 2004. *Beyond the Kalman filter: particle filters for tracking applications.,* Boston, Artech House.

RITZEN, E. 1998. *Adaptive Vehicle Weight Estimation.* MS, Linköping University.

ROZYN, M. & ZHANG, N. 2010. A method for estimation of vehicle inertial parameters. *Vehicle System Dynamics,* 48**,** 547-565.

SAAD, G., GHANEM, R. & MASRI, S. Year. Robust system identification of strongly non-linear dynamics using a polynomial chaos based sequential data assimilation technique. *In:* 48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, April 23, 2007 - April 26, 2007, 2007 Waikiki, HI, United states. American Institute of Aeronautics and Astronautics Inc., 6005-6013.

SANDU, A., SANDU, C. & AHMADIAN, M. 2006. Modeling multibody systems with uncertainties. Part I: Theoretical and computational aspects. *Multibody System Dynamics,* 15**,** 369-391.

SEGHERS, D. Local graph-based probabilistic representation of object shape and appearance for model-based medical image segmentation.

SHIMP, S. K. 2008. *Vehicle sprung mass identification using an adaptive polynomial-chaos method.* Masters of Science Masters Thesis, Virginia Polytechnic and State University.

SIMON, D. 2006. *Optimal State Estimation: Kalman, H infinity, and nonlinear approaches,* Hoboken, NJ, John Wiley & Sons, Inc.

SMITH, A., MONTI, A. & PONCI, F. Year. Indirect measurements via polynomial chaos observer. *In:* 2006 IEEE International Workshop on Advanced Methods for Uncertainty Estimation in Measurement, 20-21 April 2006, 2006 Piscataway, NJ, USA. IEEE, 6 pp.

SONG, X., AHMADIAN, M., SOUTHWARD, S. & MILLER, L. R. 2005. An adaptive semiactive control algorithm for magnetorheological suspension systems. *Journal of Vibration and Acoustics, Transactions of the ASME,* 127**,** 493-502.

SOUTHWARD, S. C. Year. Real-time parameter id using polynomial chaos expansions. *In:* ASME International Mechanical Engineering Congress and Exposition, IMECE 2007, November 11, 2007 - November 15, 2007, 2008 Seattle, WA, United states. American Society of Mechanical Engineers, 1167-1174.

TAL, R. & ELAD, S. 1999. *Method for Determining Weight of a Vehicle in Motion*. USA patent application No. 034570 filed on 03/04/1998

VAHIDI, A., DRUZHININA, M., STEFANOPOULOU, A. & HUEI, P. Year. Simultaneous mass and time-varying grade estimation for heavy-duty vehicles. *In:* Proceedings of 2003 American Control Conference, 4-6 June 2003, 2003a Piscataway, NJ, USA. IEEE, 4951-6.

VAHIDI, A., STEFANOPOULOU, A. & PENG, H. Year. Experiments for online estimation of heavy vehicle's mass and time-varying road grade. *In:* 2003 ASME International Mechanical Engineering Congress, November 15, 2003 - November 21, 2003, 2003b Washington, DC., United states. American Society of Mechanical Engineers, 451-458.

VAHIDI, A., STEFANOPOULOU, A. & PENG, H. 2005. Recursive least squares with forgetting for online estimation of vehicle mass and road grade: theory and experiments. *Vehicle System Dynamics,* 43**,** 31-55.

WEINER, N. 1938. The homogenous chaos. *American Journal of Mathematics,* 60**,** 897-963.

WENZEL, T. A., BURNHAM, K. J., BLUNDELL, M. V. & WILLIAMS, R. A. Year. Dual extended Kalman filter for vehicle state and parameter estimation. *In*, 2006. Taylor and Francis Ltd., 153-171.

WINSTEAD, V. & KOLMANOVSKY, I. V. Year. Estimation of road grade and vehicle mass via model predictive control. *In:* 2005 IEEE International Conference on Control Applications (CCA), 28-31 Aug. 2005, 2005 Piscataway, NJ, USA. IEEE, 1588-93.

XIU, D. 2007. Efficient collocation approach for parametric uncertainty analysis. *COMMUNICATIONS IN COMPUTATIONAL PHYSICS,* 2**,** 293-309.

XIU, D. & KARNIADAKIS, G. E. 2002. The Wiener-Askey polynomial chaos for stochastic differential equations. *SIAM Journal on Scientific Computing,* 24**,** 619-44.

YAMADA, N., ISHIGURO, T., KATO, H., HAYAKAWA, K., OSAWA, M. & HIBINO, R. 2006. *Vehicle weight determining device*. USA patent application 10/150,944.

YANASE, M. 2005. *Method and apparatus for estimating mass of vehicle, and method and apparatus for estimating gradient employing the method*. USA patent application 10/609619.

ZHU, G. G., TAYLOR, D. O. & BAILEY, T. L. 2000. *Recursive vehicle mass esitmation*. USA patent application 09/065366.

ZHU, G. G., TAYLOR, D. O. & BAILEY, T. L. 2002. *Recursive vehicle mass estimation system*. USA patent application 09/748050.