Optimal Observers and Optimal Control: Improving Car Efficiency with Kalman and Pontryagin

by

Kenneth D. Sebesta

Dissertation submitted to the Faculty of the Graduate School of the
University of Luxembourg in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2010

Jury:
Professor Juergen Sachau, Co-Advisor
Professor Eric Busvelle, Co-Advisor
Professor Ugo Boscain, Chair
Professor Jean-Regis Hadji-Minaglou
Professor Hassan Hammouri
Professor Thierry-Marie Guerra

Preface

The information age has radically shifted every aspect of our lives, perhaps first and foremost our scientific ones.

Before the information age, 99% of humanity's interaction with technology was limited to a car, airplane, telephone, or television. While technology had had a dramatic impact on life— from medicine to cheap manufacturing— for the populace at large direct everyday interaction with technology was rare.

Internet has changed all that. It has brought high technology into everyday lives, to the point that it is unimaginable to do without. Science is no longer the exclusive domain of the scientist. Now we enter into the age of the *makers*.

Makers don't invent from whole cloth, they assemble. They take bits bits and pieces gathered from the far-flung corners of the world and combine them to do something new, something different. Oftentimes, something *scientific*.

Who is a scientist? Historically, science was largely performed by the nobles and well-to-do. In the 19$^{\text{th}}$ century the natural sciences— geology, astronomy, etc...— were regarded as social activities. Discovering a new plant, or planet for that matter, was as much a way to impress ones friends as it was to push back the frontiers of science.

It wasn't until the early-to-mid 20$^{\text{th}}$ century that science become a career instead of a hobby. WWI and especially WWII were times of tremendous scientific progress, pushed by the various governments' fervent desires to discover and invent the secret weapons that would win the war. With the governments' desire came funding, stability, and jobs.

Now in the early 21$^{\text{st}}$ Century, science is undergoing another transformation. Scientific principles are no longer hidden in an ivory tower, papers are no longer locked behind closed doors. Individuals are now able to quickly and efficiently access the scientific world. In a way, science has gone full loop. In past times, only the rich had the time to indulge in idle fantasies. Now, we are perhaps all rich enough for such explorations. Now, we are becoming *makers*.

Most hard science will still be done by professional scientists, just like most hard journalism is done by professional journalists. But just as the journalistic world is supplemented by the bloggers, so is the scientific world supplemented by the makers. We must not forget this burgeoning world of the amateur scientist.

To this end, our scientific output must not be hidden. It must be obvious, available, and accessible. In this spirit, this dissertation is an experiment, much the same that a PhD is an experiment. This dissertation is designed such that the results and application are easy to understand for those with a less-than-formal background.

Dedication

To Laura. The best moments of my life have been spent with you.

Acknowledgments

It is difficult to express in words the journey of science. It is an open-ended trip where the consequences of decisions are years in the making. An uphill trip littered with the refuse of cast-off ideas and rejected works. A sublime trip that is filled with the joy of accomplishment, of exploration, and of knowledge.

It is said that research is “spending six months of work in order to find out which two weeks made it all worthwhile”. My thanks go to all those who helped me recognize my two weeks.

First and foremost, I’d like to thank my advisor, Professor Juergen Sachau for giving me an invaluable opportunity to work on challenging and extremely interesting projects over the past four years. He has always made himself available for help and advice and there has never been an occasion when I’ve knocked on his door and he hasn’t given me time.

I would also like to thank my co-advisor, Professor Eric Busvelle. Without his extraordinary theoretical ideas and computational expertise, this thesis would have been a distant dream. Seldom have I encountered anyone with such clarity of vision.

Thanks are due to Professors Ugo Boscain and Jean-Regis Hadji-Minaglou for agreeing to serve on my thesis committee, and to Professors Thierry-Marie Guerra and Hassan Hammouri for taking the time to review the manuscript as my reviewers.

Special thanks to Professor Jean-Paul Gauthier, for whom nothing is impossible; and to Dr. Nicolas Boizot, who walked side by side with me on our common path, in Burgundy and Luxembourg alike.

Contents

Appendices

List of Figures

List of Tables

List of Abbreviations

Abbr.	Meaning
μC	microcontroller
μP	microprocessor
CAN	Controller Area Network
CVT	Continuously Variable Transmission
CD-EKF	Continuous-Discrete Extended Kalman Filter
DD-EKF	Discrete-Discrete Extended Kalman Filter
DEM	Digital Elevation Map
DP	Dynamic Programming
ECU	Engine Control Unit
EKF	Extended Kalman Filter
PMP	Pontryagin's Maximum Principle
UKF	Unscented Kalman Filter
NED	North-East-Down coordinate system

Introduction

Every age has its theme, and ours is energy. Our civilization is founded on the stuff. It warms our houses, keeps the lights on, runs the computers, and is present in every single aspect of our lives. It is no coincidence that, according to the Economist, three of the ten largest companies in the world are energy companies, and eight of the largest ten are involved either in transportation and/or energy.

Without energy, the author would not have written this paper, and the reader would not be reading it.

It is impossible to overestimate the impact energy has had on our lives, and perhaps only the most savage of cultures lost in the deepest jungles of the Amazon or the African heartland can even truly relate to what it means to go without.

Civilization as we know it would disappear overnight if we were to lose our energy source. Thus, it would seem wise to make every drop, so to speak, count. Unfortunately, progress along this front has been difficult. Some would have us believe that we can drill our way out of our problem. With the greatest respect[24], and in light of the ongoing *Deep Horizons* catastrophe in the Gulf of Mexico, the author disagrees.

The case has been laid out and it is clear that there is a pressing need to reduce energy waste and consumption.

Everyone talks about saving gas, but do we actually know how to do it? Of course, we can always say that "better technology" will find a solution, but that begs the question "What exactly is better technology?" Do we need better technology in order to improve fuel efficiency, or can we do that now, with the existing tools we have at hand?

We might learn something about the current state of technology vs. the current state of applied controls by examining the Grand DARPA Challenge. The Grand DARPA Challenge was a competition to build an autonomously driven vehicle that would race across the Mojave Desert, on a 240km course driving at average speeds in excess of 35km/hr. The first year, 2004, the farthest any of the 21 teams' vehicles went was a little less than 12km. Contrast this to 2005, when the challenge was not only successfully met, it was completed by a total of 5 out of 23 teams. Indeed, only one team *failed* to best the previous year's record. While technology evolves quickly, it does not evolve that quickly.

No, we have to look elsewhere for an explanation of the dramatic improvement in only 365 days.

We have to look to controls.

The field of modern optimal control is a relatively young field of mathematics. Fifty-one years ago, Lev Semenovich Pontryagin published *The Mathematical Theory of Optimal Processes*, founding the modern study of optimal control.

(Of course, like all mathematics, even if optimal control is a relatively recent branch of mathematics, the roots go back much further, all the way to Bernoulli, who first mentioned the *Principle of Optimality* in connection with the Brachistochrone Problem[1], and Newton, who solved it.)

Pontryagin inspired many of his contemporaries. Indeed, Rudolph Kalman used his work to develop the Linear Quadratic Regulator in 1960, followed later on in the year by what is now known as the discrete-discrete Kalman Filter and then in 1961 the continous-continuous Kalman Filter.

Shortly before Pontryagin published his seminal paper, Bellman introduced the idea of *Dynamic Programming*, a sort of brute force approach to optimal control of discrete-time systems. Optimal control was found to express a fundamental property of motion in natural systems. Bellman demonstrated that the natural direction for solving optimal control problems is backwards in time.

Since then, much fundamental research has been performed in the field of optimal control. Clarke[16] and Sussmann[46] have relaxed and generalized its principles to overcome some of the original limitations of the PMP. Today, the advent of high-power computers in low-power packages is opening the door to optimizing processes that just a few years ago would have been too costly or time consuming.

According to Benjamin Dessus, roughly half of the installed power base in the West is made up of cars and trucks. Obviously, there is a wealth of opportunities to use optimal control to make reductions in emissions and fuel consumption. However, all optimality studies depend on an engine efficiency model (or alternatively an engine specific fuel consumption (SFC) model). In previous studies, these models were assumed to be available ahead of time.[25, 51, 39]

Unfortunately, this is not the case in reality. Car manufacturers are notoriously tight-lipped and guard their data jealously. Furthermore, engine data alone does not tell the whole story, as it is ideal data from an ideal engine under ideal conditions. As any driver knows, the manufacturer reported fuel economy is terribly imprecise in real-world situations.

For an optimality study, it is essential to know the true system efficiency, which cannot be predicted from engine data alone. Many other car-specific factors come into play, such as transmission losses, rolling resistance, air drag, etc... Even when these values can be found in literature or from the manufacturer, they are oftentimes unreliable (e.g. coefficients of air resistance) and have limited usefulness.

A better optimality approach would take into account the real-world parameters of the individual vehicle in question. This approach should be able to adapt to changing circumstances, such as mass change when a passenger gets out of a car or an air resistance change

[1] Literally, "shortest time". The Brachistochrone Problem was one of the earliest problems posed in the calculus of variations. The problem was to "find the shape of the curve down which a bead sliding from rest and accelerated by gravity will slip (without friction) from one point to another in the least time". Newton solved it in one day. The answer: a *cycloid*.

when the driver puts down a convertible top. This approach also would even allow monitoring motor efficiency with time, opening the door to early fault warning. (A tire low-pressure warning comes to mind.)

Can we improve fuel efficiency for every single car in the world, without modifying a single one of them? Can we do this without imposing burdensome— and dangerous— driving styles on the drivers? Can efficiency be improved without sacrificing air quality?

The author believes the answer is yes.

But first we have to develop some tools along the way.

****

Dissertation overview

Goal: The goal of this PhD is to develop a system by which we can optimize fuel efficiency in individual vehicles, based on data gathered from these same. We offer targeted, specific controls trajectories for each individual car and individual trip.

- Chapter 1 is a survey of the state of the art in modeling and optimal control.
- Chapter 2 details current state of the art in sensoring and processing components in addition to describing the sensor package and data logger developed during the research project.
- Chapter 3 shows the application to the fuel efficiency mapping problem. A case study is presented demonstrating the observer. There are also tips and techniques that are of interest to those who would like to apply this work in a real-world environment.
- Chapter 4 illustrates the process for applying Pontryagin's Maximum Principle to an arbitrary vehicle. Several case studies are presented, demonstrating the controller.
- Appendix A is devoted to the subject of non-linear observers, with a special focus on the continuous-discrete time domain and fusion of asynchronous data.
- Appendix B is devoted to the subject of optimal controllers, with a special focus on Pontryagin's Maximum Principle.

In addition to the ones cited above, additional appendices contain many of the tools that had to be created and/or invented in order to turn the theory into practice. Specifically: data protocols, schematics, a short description of the CAN bus and a working CAN logger, a description of the data files, an example of using Google Earth to visualize data, Quad coordinates, etc...

Lastly, there is a worked example of Pontryagin's Maximum Principle, designed to be especially useful for engineers and scientists who have not yet been exposed to optimal control.

Notation conventions

Throughout this dissertation, the following notation conventions apply:

Description	Example	Meaning
Bold, lowercase letter	$\mathbf{x}$	vector
Bold, upppercase letter	$\mathbf{A}$	matrix
c, subscripted	$\mathbf{A}_c$	continuous time
d, subscripted	$\mathbf{A}_d$	discrete time
i, j, subscripted	$\mathbf{A}_{i,j}$	i^{th}, j^{th} element
k, subscripted	$\mathbf{x}_k$	k^{th} time step
T, superscripted	$\mathbf{A}^T$	matrix transpose
Hat, over letter	$\hat{\mathbf{x}}$	estimated value (as opposed to real)
Star, superscripted	$\mathbf{u}^*$	optimal
Minus, superscripted	$\mathbf{P}_k^-$	intermediate value between time k and time $k+1$
Dot, superscripted	$\dot{\mathbf{x}}$	time deivative

Table 1: Notation conventions

Variable conventions

Throughout this dissertation, the following variable conventions apply:

Variable	Meaning
f	system model vector valued function
h	measurement model vector valued function
$\mathbf{x}$	state vector
$\mathbf{y}$	measurement vector
$\mathbf{u}$	control vector
$\mathbf{A}$	Jacobian of f with respect to $\mathbf{x}$
$\mathbf{B}$	Jacobian of f with respect to $\mathbf{u}$
$\mathbf{H}$	Jacobian of h with respect to $\mathbf{x}$
J	cost
$\mathbf{K}$	gain matrix
$\mathbf{P}$	state noise covariance
n_x	number of state variables
n_y	number of measurement variables
n_u	number of control variables
Δt	time step
η	engine efficiency
ϕ	inverse of engine efficiency, i.e. $1/\eta$

Table 2: Variable conventions

Chapter 1

State of the (Theoretical) Art

Contents

Optimal control theory has only recently broken into the automotive industry. An important motivation for this progress is the increasing number of electronic sensors in modern cars, opening the door to more and better controls, for instance in the automatic control of the air-fuel mixture. Sophisticated control methods are currently used in a variety of applications, however, in the industry most such problems typically model very short time intervals and do not require optimal control techniques. Optimizing a passenger car's efficiency along its trajectory is still confined to the world of research.

The car efficiency optimization problem has been considered in [39, 53]. Most of the literature has chosen to use Dynamic Programming (Sec. B.2.3)[9, 25] or Non-Linear Programming (Sec. B.2.4), although recently some studies have examined the use of Pontryagin's Maximum Principle as it applies to hybrid cars.[39, 53]

The authors of [25] found that optimal control applied to semi-trucks could both increase fuel efficiency and decrease driving time, all while reducing wear and tear on the transmission.

This chapter is devoted to establishing the necessary models for (1) generating a efficiency map for an individual car and (2) creating the optimal control from this map.

1.1 State models

The optimization problem presented in this dissertation is split by necessity into two parts: observation of holistic system efficiency and application of optimal control to real-world examples. To this end, in the following two sections we present state-space models for 1) the observation problem and 2) the control problem.

The observer model represents the link between the sensors and the state, whereas the controller model needs only the state variables necessary for optimization. The upshot is that the 17-state observer model is reduced to a 2-state controller model, or even 1-state if the optimal target is only a one-dimensional speed target, and not the two dimensional pairing of position and speed.

1.2 Observer model

The observer model is designed to give an estimation of holistic fuel efficiency, based on data collected only from real-world driving.[43] It is built on a simplified model of car kinematics, ignoring complex effects such as increased rolling resistance during turns, etc... If additional measurements are available, the model can be extended in order to give an increase in accuracy, but with the associated increase in computation time.

1.2.1 Modeling

We begin by establishing the system model.

1.2.1.1 Model variables

See Table 1.1.

Variable	Description	Units
x_i	car displacement in the $\hat{\imath}$ axis	[m]
v_i	car speed in the $\hat{\imath}$ axis	$[\frac{\mathrm{m}}{\mathrm{s}}]$
a_i, a_j, a_k	car acceleration in the $\widehat{ijk}$ axes	$[\frac{\mathrm{m}}{\mathrm{s}^2}]$
F_{thrust}	thrust force along $\hat{\imath}$ axis	[N]
ψ	car heading $\in [0, 2\pi)$	[rad]
θ	car pitch $\in [-\frac{\pi}{2}, \frac{\pi}{2})$	[rad]
S_j	road straightness coefficient (horizontal plane)	$[\frac{\mathrm{rad}}{\mathrm{m}}]$
S_k	road straightness coefficient (vertical plane)	$[\frac{\mathrm{rad}}{\mathrm{m}}]$
N, E, D	Northing, Easting, and Down (local projection plane)	[m]
E_{in}	energy content input, fuel	[J]
P_{in}	power input, fuel	[W]
b_i, b_j, b_k	accelerometer biases in the $\widehat{ijk}$ axes	$[\frac{\mathrm{m}}{\mathrm{s}^2}]$
θ_e	engine angular displacement	[rad]
ω_e	engine angular speed	$[\frac{\mathrm{rad}}{\mathrm{s}}]$
C_{rr}	rolling resistance coeffecient	[-]
C_d	air drag coefficient	[-]
A	car frontal cross-section area	$[\mathrm{m}^2]$
ρ	air density	$[\frac{\mathrm{kg}}{\mathrm{m}^3}]$
M	total vehicle mass	[kg]
M_a	total accelerated mass	[kg]
g	acceleration due to gravity	$[\frac{\mathrm{m}}{\mathrm{s}^2}]$

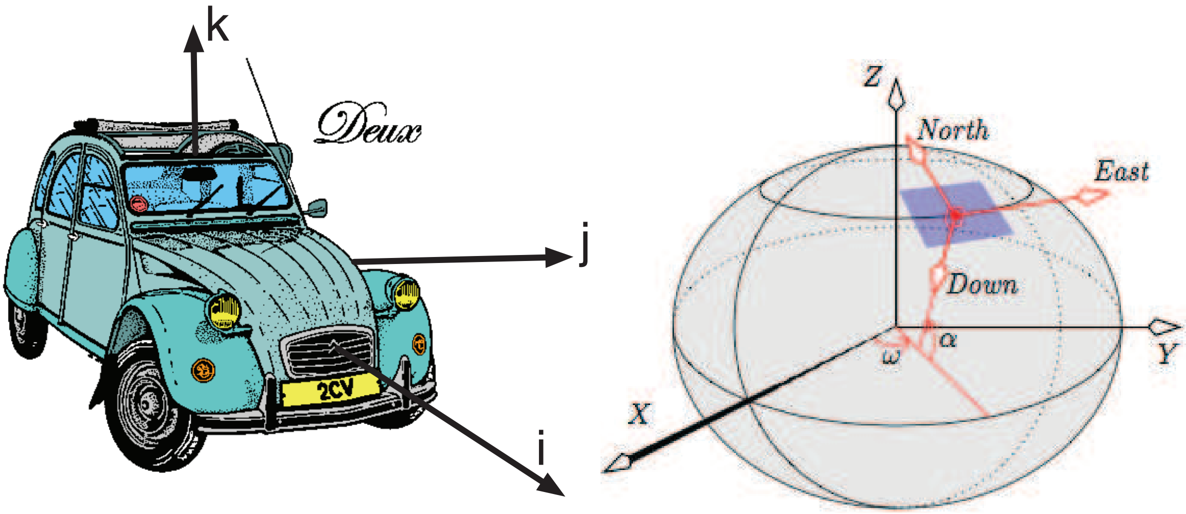
Table 1.1: Observer model variables

1.2.1.2 Reference frames

Two reference frames are used:

- An inertial reference frame $\widehat{ijk}$ linked to the car, where $+\hat{\imath}$ points in the direction of forward movement, $+\hat{\jmath}$ points 90 degrees left, and $+\hat{k}$ points straight up from the car. See Fig. 1.1a

- A global reference frame NED (North-East-Down) which is the local mercator projection of geodetic latitude-longitude-altitude coordinates. Note that in order to maintain right-handed orthogonality, *Down* represents negative altitude. Also, note that proper care must be taken while projecting geodetic coordinates into a mercator projection. [52] provides a good explanation of the process, along with Matlab code. See Fig. 1.1b



(a) Body

(b) North-East-Down

Figure 1.1: Reference frames

1.2.1.3 Kinematics

Net acceleration is proportional to the sum of forces along the road surface. See Fig. 1.2.

$$M_a \dot{v}_i = \Sigma F = F_{thrust} - F_{drag} - F_{roll} - F_{gravity} \tag{1.1}$$

$$F_{drag} = \frac{1}{2} \rho C_d A v_i^2 \tag{1.2}$$

$$F_{roll} = C_{rr}(Mg\cos(\theta) + S_k v_i^2) \tag{1.3}$$

$$F_{gravity} = Mg\sin(\theta) \tag{1.4}$$

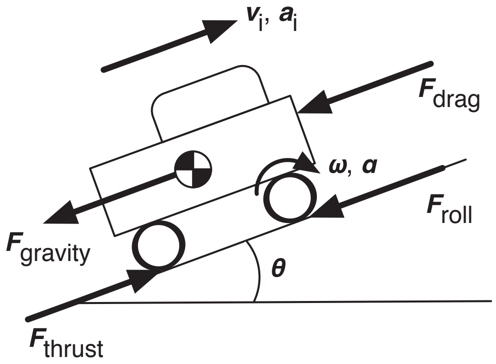


Figure 1.2: Free-body car diagram

where

$$M_a = \frac{J}{r_w^2} + M$$

M_a is the equivalent *accelerated* mass, and takes into account the angular momentum of the various spinning parts of the car, where J is the sum of all polar moment of inertias, and r_w the wheel radius. This constant represents all the various bits and pieces that must accelerate in order for the car itself to accelerate.

Note that this model does not separate braking forces (both engine braking and friction brakes) from thrust forces. For non-hybrid vehicles, this is an unnecessary distinction, as only data gathered while the engine is propelling the car forward is interesting.

The car is assumed fixed to the road, and there is a no-slip condition in both longitudinal and latitudinal directions. Furthermore, it is assumed that the car undergoes no rolling rotation about the $\hat{\imath}$-axis ($\phi, \dot{\phi} \equiv 0$). Thus, v_j, v_k=0 and any acceleration Σa_j or Σa_k represents a centripetal acceleration, i.e. the car is on a curving path:

$$\begin{aligned} a_{j,b} &= S_j v_i^2 \\ a_{k,b} &= S_k v_i^2 \end{aligned} \tag{1.5}$$

Note 1 *S_j and S_k, the road straightness coefficients, represent the inverse of road curvature. Straightness coefficients are preferable to road curvature coefficients, since straightness coefficients are finite when on a straight section of road. Realistic values for S_j seem to lie on the interval [-0.05, 0.05] and for S_k in [-0.005, 0.005]*

Likewise, the rate of change of ψ and θ is dependent on the road straightness coefficients and forward velocity:

$$\begin{aligned} \dot{\psi} &= S_j v_i \\ \dot{\theta} &= S_k v_i \end{aligned} \tag{1.6}$$

Note 2 *In the $\widehat{NED}$ coordinate system, $+\psi$ represents a yaw to the right, and $+\theta$ represents a pitch upwards.*

The N, E, D dynamic equations are written in spherical coordinates centered about the body reference frame:

$$\begin{aligned}\dot{N} &= \cos(\psi)\cos(\theta)v_i \\ \dot{E} &= \sin(\psi)\cos(\theta)v_i \\ \dot{D} &= -\sin(\theta)v_i\end{aligned} \tag{1.7}$$

The engine turns at a fixed rate with respect to the wheels, neutral gear and differential withstanding. In a normal car transmission, (See 4.4.2), this ratio is different between gears but otherwise constant.

$$\dot{\omega_e} = \dot{v}_i N_{spd} \tag{1.8}$$

As we cannot directly measure N_{spd}, we suppose that the engine and wheels are not connected, which is the same as taking $N_{spd} \equiv 0$. While this is obviously physically incorrect, since we have very good independent sensors for estimating ω_e and v_i, this approach yields satisfactory results.

Lastly, a measured acceleration will always have slight biases in each direction, not only due to imperfections in fabrication and in mounting, but also due to changes of load distribution in the car. Note also that an accelerometer in a gravity field measures the pull of gravity in addition to the net body acceleration. These considerations lead us to the following equations:

$$\begin{aligned}a_{i,m} &= a_{i,b} + \tfrac{M}{M_a} g\sin(\theta) + b_i \\ a_{j,m} &= a_{j,b} + b_j \\ a_{k,m} &= a_{k,b} - g\cos(\theta) + b_k\end{aligned} \tag{1.9}$$

1.2.1.4 Energy analysis

Instantaneous power-in is the volume of fuel consumed per second times the amount of energy per volume.

$$P_{in} = k\dot{V}_{fuel} \tag{1.10}$$

Instantaneous power-out is the thrust force times velocity:

$$P_{out} = F_{thrust} v_i \tag{1.11}$$

Efficiency is defined as

$$\eta = \frac{P_{out}}{P_{in}} \tag{1.12}$$

Power-out is also equal to torque times angular velocity:

$$P_{out} = T_e \omega_e \tag{1.13}$$

Thus, by rearranging eqns. (1.11) and (1.13) the torque is

$$T_e = F_{thrust} \frac{v_i}{\omega_e} \tag{1.14}$$

Note 3 *Result 1.14 is completely independent of transmission gear ratios.*

1.2.2 Complete observer model

Unknown parameters are included in the model in order to observe them. F_{thrust} is the most critical of them, as F_{thrust} is used to estimate P_{out}, but we do not have a direct measurement of it.

The state and measurement vectors— $\mathbf{x}, \mathbf{y}$ respectively— are:

$$\mathbf{x} = \begin{pmatrix} x_i \\ v_i \\ F_{thrust} \\ \psi \\ \theta \\ N \\ E \\ D \\ S_j \\ S_k \\ E_{in} \\ P_{in} \\ \theta_e \\ \omega_e \\ b_i \\ b_j \\ b_k \end{pmatrix} \qquad \mathbf{y} = \begin{pmatrix} x_{i,m} \\ a_{i,m} \\ a_{j,m} \\ a_{k,m} \\ DEM_{\psi} \\ DEM_{alt} \\ GPS_{lat} \\ GPS_{lon} \\ GPS_{alt} \\ INJ_{duration} \\ INJ_{time} \end{pmatrix} \tag{1.15}$$

where the state equations are

$$\begin{aligned} \dot{x}_i &= v_i \\ \dot{v}_i &= \frac{1}{M_a}\left(F_{thrust} - \text{sign}(v_i)\left(\frac{1}{2}\rho C_d A v_i^2 + M C_{rr}\left(g\cos(\theta) + S_k v_i^2\right)\right) - Mg\sin(\theta)\right) \\ \dot{\psi} &= S_j v_i \\ \dot{\theta} &= S_k v_i \\ \dot{N} &= v_i \cos(\psi)\cos(\theta) \\ \dot{E} &= v_i \sin(\psi)\cos(\theta) \\ \dot{D} &= -v_i \sin(\theta) \\ \dot{E}_{in} &= P_{in} \\ \dot{\theta}_e &= w_e \end{aligned} \tag{1.16}$$

and

$$\dot{F}_{thrust} = \dot{S}_j = \dot{S}_k = \dot{P}_{in} = \dot{\omega}_e = \dot{b}_i = \dot{b}_j = \dot{b}_k = 0 \tag{1.17}$$

where the observation equations are

$$\begin{array}{rcl}
x_{i,m} & = & x_i \\
a_{i,m} & = & \dfrac{1}{M_a}\left(F_{thrust} - \text{sign}(v_i)\left(\dfrac{1}{2}\rho C_d A v_i^2 + MC_{rr}\left(g\cos(\theta) + S_k v_i^2\right)\right)\right) + b_i \\
a_{j,m} & = & S_j v_i^2 + b_j \\
a_{k,m} & = & -g\cos(\theta) - S_k v_i^2 + b_k \\
DEM_{\psi} & = & \psi \\
DEM_{alt} & = & -D \\
GPS_{lat} & = & N \\
GPS_{lon} & = & E \\
GPS_{alt} & = & -D \\
INJ_{duration} & = & E_{in} \\
INJ_{time} & = & \theta_e
\end{array} \tag{1.18}$$

and where

- $x_{i,m}$ is the measured distance at time t_k
- $a_{i,m}, a_{j,m}, a_{k,m}$ are the measured accelerations in the longitudinal, latitudinal, and vertical body-frame axes at time t_k
- DEM_{ψ}, DEM_{alt} are road headings and road altitude, respectively, at time t_k. This data is read from a DEM (Digital Elevation Map), when available. Note that it is entirely possible to get reliable heading data from the DEM without having the associated altitude data, and that furthermore these two measurements are decoupled in time.
- $GPS_{lat}, GPS_{lon}, GPS_{alt}$ are, respectively, the GPS latitude, longitude, and altitude measurements taken at time t_k and geodetically transformed into the local NED projection system.
- $INJ_{duration}$ is the duration the fuel injector is open, times the quantity of fuel per second of opening, times the amount of energy per quantity of fuel. In other words, $\Delta E_{in} \propto t$, and so this is equivalent to the measured energy input at time t_k
- INJ_{time} is the time t_k at which the injector opens, which represents a multiple of π radians (See Sec. 2.3.4.1 for more details.) of angular distance traveled by the engine since the previous measurement.

Note 4 *The GPS is not used as a speed measurement, since it was impossible to know if our particular GPS unit measured speed directly through doppler shift or if it simply reported speed as a filtered change of position with respect to time. In the former case, it would indeed have been a good measurement. If it is possible to find a GPS that uses doppler shift, then it would improve system accuracy to integrate it into the measurement model.*

1.2.3 Observability normal form

We present here a normal form of the observer, as described in [21]. The system is identical to the previous model, but transformed such that the structure follows certain rules (such as each state in a block can only depend on the states above it, and at the very least the topmost state in a block must be an output). The observability normal form is useful for demonstrating observability (cf. Sec. A.1.2), amongst other things.

We would like to write the normal form system as

$$\dot{\mathbf{z}} = \mathbf{A}\mathbf{z} + b(\mathbf{z}) \tag{1.19}$$

First we rewrite the variable names:

$$\left\{ \begin{array}{rclrcl}
 & & & \dot{x}_1 & = & x_2 \\
x_1 & = & x_i & \dot{x}_2 & = & \frac{1}{M_a}\left(x_3 + \mathrm{sign}(x_2)\left(\frac{1}{2}\rho C_d A x_2^2 + MC_{rr}\left(g\cos(x_5) + x_{10}x_2^2\right)\right)\right. \\
x_2 & = & v_i & & & \quad \left. -Mg\sin(x_5)\right) \\
x_3 & = & F_T & \dot{x}_3 & = & 0 \\
x_4 & = & \psi & \dot{x}_4 & = & x_9 x_2 \\
x_5 & = & \theta & \dot{x}_5 & = & x_{10} x_2 \\
x_6 & = & N & \dot{x}_6 & = & x_2\cos(x_4)\cos(x_5) \\
x_7 & = & E & \dot{x}_7 & = & x_2\sin(x_4)\cos(x_5) \\
x_8 & = & D & \dot{x}_8 & = & -x_2\sin x_5 \\
x_9 & = & S_j & \dot{x}_9 & = & 0 \\
x_{10} & = & S_k & \dot{x}_{10} & = & 0 \\
x_{11} & = & E_{in} & \dot{x}_{11} & = & x_{12} \\
x_{12} & = & P_in & \dot{x}_{12} & = & 0 \\
x_{13} & = & \theta_e & \dot{x}_{13} & = & x_{14} \\
x_{14} & = & w_e & \dot{x}_{14} & = & \dot{x}_2 N_{spd} \\
x_{15} & = & b_i & \dot{x}_{15} & = & 0 \\
x_{16} & = & b_j & \dot{x}_{16} & = & 0 \\
x_{17} & = & b_k & \dot{x}_{17} & = & 0
\end{array} \right. \tag{1.20}$$

where the measurements are

$$\begin{aligned}
& y_1 = x_1 \\
& y_2 = x_4 \\
& y_3 = x_6 \\
& y_4 = x_7 \\
& y_5 = -x_8 \\
& y_6 = x_{11} \\
& y_7 = x_{13} \\
& y_8 = \frac{1}{M_a}\left(x_3 - \mathrm{sign}(x_2)\left(\frac{1}{2}\rho C_d A x_2^2 + MC_{rr}\left(g\cos(x_5) + x_{10}x_2^2\right)\right)\right) + x_{15} \\
& y_9 = x_9 x_2^2 + x_{16} \\
& y_{10} = -g\cos(x_5) - x_{10}x_2^2 + x_{17}
\end{aligned} \tag{1.21}$$

In order to transform system (1.16) into the observability normal form, we propose the following change of coordinates $\Phi(\mathbf{x}) = \mathbf{z}$ and its inverse $\Phi^{-1}(\mathbf{z}) = \mathbf{x}$

$$\left\{ \begin{array}{ll}
\begin{aligned}
z_1 &= x_1 \\
z_2 &= x_2 \\
z_3 &= x_3 - Mg\sin(x_5) \\
&\quad -MC_{rr}\left(g\cos(x_5) + x_{10}x_2^2\right) \\
z_4 &= x_4 \\
z_5 &= x_9 x_2 \\
z_6 &= x_6 \\
z_7 &= x_7 \\
z_8 &= -x_8 \\
z_9 &= x_2\sin(x_5) \\
z_{10} &= x_2^2 x_{10}\cos(x_5) \\
&\quad +\tfrac{\sin(x_5)}{M_a}(z_3 - \tfrac{1}{2}\rho C_d A z_2^2) \\
z_{11} &= x_{11} \\
z_{12} &= x_{12} \\
z_{13} &= x_{13} \\
z_{14} &= x_{14} \\
z_{15} &= \tfrac{1}{M_a}\left(x_3 - \tfrac{1}{2}\rho C_d A x_2^2\right. \\
&\qquad -MC_{rr}\left(g\cos(x_5)\right. \\
&\qquad\quad \left.\left.+x_{10}x_2^2\right)\right) + x_{15} \\
z_{16} &= x_9 x_2^2 + x_{16} \\
z_{17} &= -g\cos(x_5) \\
&\qquad\quad -x_{10}x_2^2 + x_{17}
\end{aligned}
&
\begin{aligned}
x_1 &= z_1 \\
x_2 &= z_2 \\
x_3 &= z_3 + MC_{rr}\left(g\sqrt{1-(z_9/z_2)^2}\right. \\
&\quad \left.+\frac{M_a z_2 z_{10} - z_9\left(z_3 - \frac{1}{2}\rho C_d A z_2^2\right)}{M_a z_2\sqrt{1-(z_9/z_2)^2}}\right) \\
&\quad +Mg\frac{z_9}{z_2} \\
x_4 &= z_4 \\
x_5 &= \arcsin\left(z_9/z_2\right) \\
x_6 &= z_6 \\
x_7 &= z_7 \\
x_8 &= -z_8 \\
x_9 &= z_5/z_2 \\
x_{10} &= \frac{M_a z_2 z_{10} - z_9\left(z_3 - \frac{1}{2}\rho C_d A z_2^2\right)}{M_a z_2^3\sqrt{1-(z_9/z_2)^2}} \\
x_{11} &= z_{11} \\
x_{12} &= z_{12} \\
x_{13} &= z_{13} \\
x_{14} &= z_{14} \\
x_{15} &= z_{15} - \tfrac{1}{M_a}\left(z_3 - \tfrac{1}{2}\rho C_d A z_2^2\right. \\
&\quad \left.+Mg(z_9/z_2)\right) \\
x_{16} &= z_{16} - z_5 z_2 \\
x_{17} &= z_{17} + g\sqrt{1-(z_9/z_2)^2} \\
&\quad +\frac{M_a z_2 z_{10} - z_9\left(z_3 - \frac{1}{2}\rho C_d A z_2^2\right)}{M_a z_2\sqrt{1-(z_9/z_2)^2}}
\end{aligned}
\end{array}\right.$$

which gives

$$\dot{\mathbf{z}} = \begin{pmatrix} \mathbf{A}_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{A}_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{A}_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{A}_4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{A}_5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathbf{A}_6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{A}_7 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{A}_8 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{A}_9 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{A}_{10} \end{pmatrix} \mathbf{z} + \begin{pmatrix} 0 \\ -\frac{1}{M_a}\frac{1}{2}\rho C_d A z_2^2 \\ b_3(z) \\ 0 \\ b_5(z) \\ b_6(z) \\ b_7(z) \\ 0 \\ 0 \\ b_{10}(z) \\ 0 \\ 0 \\ 0 \\ b_{14}(z) \\ b_{15}(z) \\ b_{16}(z) \\ b_{17}(z) \end{pmatrix} \tag{1.22}$$

where

$$\mathbf{A}_1 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & \frac{1}{M_a} \\ 0 & 0 & 0 \end{pmatrix}, \quad \mathbf{A}_5 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$$

$$\mathbf{A}_2 = \mathbf{A}_6 = \mathbf{A}_7 = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$$

$$\mathbf{A}_3 = \mathbf{A}_4 = \mathbf{A}_8 = \mathbf{A}_9 = \mathbf{A}_{10} = 0$$

In addition, we have

$$\mathbf{y}_{t_k} = \begin{pmatrix} 1 & 0 & 0| & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0| & 1 & 0| & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0| & 1| & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0| & 1| & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0| & 1 & 0 & 0| & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0| & 1 & 0| & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0| & 1 & 0| & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0| & 1| & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0| & 1| & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0| & 1 \end{pmatrix} \mathbf{z} \tag{1.23}$$

Eqns. (1.22) and (1.23) are the model's normal form. Note that $x_2 > v_{i,min} > 0$, that $\theta = x_5 \in [-\frac{\pi}{4}; \frac{\pi}{4}]$, and that $\cos(\arcsin(\alpha)) = \sqrt{1-\alpha^2}$ (because $|\alpha| < \frac{\pi}{2}$). Therefore the function is well defined. Again, it bears repeating that the system is no longer observable when $x_2 = 0$, and certainly has numerical stability problems as x_2 becomes tiny. This will be taken into account in the following sections.

This change of variables brings the model into the form

$$\dot{\mathbf{z}} = \begin{pmatrix} \mathbf{A}_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{A}_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{A}_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{A}_4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{A}_5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathbf{A}_6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{A}_7 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{A}_8 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{A}_9 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{A}_{10} \end{pmatrix} \mathbf{z} + \begin{pmatrix} 0 \\ -\frac{1}{M_a}\frac{1}{2}\rho C_d A z_2^2 \\ b_3(\mathbf{z}) \\ 0 \\ b_5(\mathbf{z}) \\ b_6(\mathbf{z}) \\ b_7(\mathbf{z}) \\ 0 \\ 0 \\ b_{10}(\mathbf{z}) \\ 0 \\ 0 \\ 0 \\ b_{14}(\mathbf{z}) \\ b_{15}(\mathbf{z}) \\ b_{16}(\mathbf{z}) \\ b_{17}(\mathbf{z}) \end{pmatrix} \qquad (1.24)$$

where

$$A_1 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & \frac{1}{M_a} \\ 0 & 0 & 0 \end{pmatrix}, \quad A_5 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$$

$$A_2 = A_6 = A_7 = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$$

$$A_3 = A_4 = A_8 = A_9 = A_{10} = 0$$

1.3 Controller model

The controller model is very basic, as nothing more complicated is needed to model distance traveled and energy consumed. Thus, the only controls are the engine speed, the transmission gear ratio, and the desired torque. Calculating power input requires only the addition of the engine efficiency to the model. The rest of the variables in eq. (1.15) are not necessary for minimizing the energy usage.

In addition, we must define the cost, J, to be minimized across the controller trajectory. While for this work absolute energy consumption was chosen, this cost could easily include other factors, such as particulate output, $C0_2$ emissions, etc... (In which case, it goes without saying that the controller model would have to be remade in order to model these new costs.)

1.3.1 Modeling

1.3.1.1 Model variables

See Table 1.2.

Variable	Description	Units
v_i	car speed in the $\hat{i}$ axis	$[\frac{\mathrm{m}}{\mathrm{s}}]$
ω_e	engine angular speed	$[\frac{\mathrm{rad}}{\mathrm{s}}]$
T_e	engine torque	[Nm]
N_{spd}	transmission ratio	[-]
η	engine efficiency	[-]

Table 1.2: Controller model variables

1.3.1.2 Cost

A minimizing controller must have a cost, J, to minimize.

A heat engine transforms the fuel energy into work energy and waste heat. In the case of a car engine, a part of the work energy is lost due to various frictions and thermodynamic inefficiencies in a car (air resistance, rolling resistance, transmission losses, etc...) a part of the energy is used to run the various car accessories, and the balance is transformed into kinetic and/or potential energy.

We seek to minimize fuel consumption, that is to say, minimize E_{in}

$$J = E_{in} = \int_0^T P_{in} dt \tag{1.25}$$

T can be fixed-horizon or not, i.e. if we wish we can apply a time criteria to the optimization or not, depending on the goals of a particular driving phase. See Sec. 4.2 for further details.

1.3.1.3 Kinematics

We start with the basic net forces equation developed in eq. (1.1), given in its refined form:

$$\dot{v}_i = \frac{1}{M_a}\left(F_{thrust} - \mathrm{sign}(v_i)\left(\frac{1}{2}\rho C_d A v_i^2 + M C_{rr}\left(g\cos(\theta) + S_k v_i^2\right)\right) - Mg\sin(\theta)\right)$$

By combining eqns. (1.8) and (1.14) into the above, we arrive at

$$\dot{v}_i = \frac{1}{M_a}\left(T_e N_{spd} - \mathrm{sign}(v_i)\left(\frac{1}{2}\rho C_d A v_i^2 + M C_{rr}\left(g\cos(\theta) + S_k v_i^2\right)\right) - Mg\sin(\theta)\right) \tag{1.26}$$

We keep v_i in the model, instead of transforming it into $\dfrac{\omega}{N_{spd}}$ because it is a continuous variable, whereas N_{spd} is not.

1.3.2 Complete controller model

The state vector is

$$\mathbf{x} = \begin{pmatrix} x_i \\ v_i \end{pmatrix} \tag{1.27}$$

and the dynamic equations are given by:

$$\dot{\mathbf{x}} = \begin{pmatrix} v_i \\ \frac{1}{M_a}\left(T_e N_{spd} - \text{sign}(v_i)\left(\frac{1}{2}\rho C_d A v_i^2 + M C_{rr}\left(g\cos(\theta) + S_k v_i^2\right)\right) - Mg\sin(\theta)\right) \end{pmatrix} \tag{1.28}$$

While the engine efficiency, η, is a function of many parameters, most of these are intrinsic to the engine construction. Most of the remainder are a function of the engine control system. Practically, the only two parameters that are left to the user to be controlled are the engine speed and torque. As luck would have it, these are the most important in determining engine efficiency. As such, we suppose η to be only a function of ω_e and T_e, and define a function ϕ and its polynomial approximation (to be used in Chap. 4).

$$\phi(\omega_e, T_e) = \frac{1}{\eta(\omega_e, T_e)} \approx a_0 + a_1\omega_e + a_2 T_e + a_3\omega_e T_e + a_4\omega_e^2 + a_5 T_e^2 + \ldots \tag{1.29}$$

and thus we can calculate

$$P_{in} = \frac{\omega_e T_e}{\eta(\omega_e, T_e)} = \omega_e T_e \phi(\omega_e, T_e) \tag{1.30}$$

1.4 Discussion

The key parameter in the minimization problem is η. This is the parameter that links the controls with the cost. The model for η is difficult to estimate, as it depends on several other estimated parameters, one of which, F_{thrust} is very noisy and itself difficult to measure/estimate. Thus the observation problem hinges on the best possible estimation of F_{thrust} possible.

This is in itself provokes a cascade of dependencies. A close examination of F_{thrust} shows that it depends heavily on θ, and to a less extent several constants ($C_r r$, C_d, ρ, etc...) and the car's velocity (v_i). While v_i is easy to measure and estimate (from GPS and odometry data), θ is somewhat difficult to measure in a dynamic system without a gyroscope. Of course, at a complete stop, θ can be inferred from the direction of the gravity vector, but when the car is accelerating it becomes difficult to have a clear measurement.

Other state parameters, such as N, E, are important only insofar as they contribute to improving the estimation of F_{thrust}. The only exceptions to this are the engine speed and energy input states, $(\theta_e, \omega_e, E_{in}, P_{in})$. The engine speed is important as it is a control. The input energy is important because it is the quantity to be minimized. However, for both of these the model and measurements are excellent, so their impact on the model is not as critical as F_{thrust}.

Chapter 2

State of the (Practical) Art

Contents

As frequently seen in this dissertation, controlling and observing have different needs, and thus onboard equipment can be split along these two lines.

Monitoring equipment is composed of a sensor package, a μC (microcontroller), and an SD card for data storage.

The controlling equipment is a μP (microprocessor) that calculates the optimal control trajectory for a given driving route. It bases its controls on information from previously observed sensor data and GPS routing software.

In an industrial setting, a car manufacturer would already have access to all this sensor data, and indeed in a more global sense to the efficiency map. However, the challenge is just that: to recreate the efficiency map with other techniques than depending on the manufacturer.

This project's sensor suite was inspired by the MPGuino[17], a basic mpg (mile per gallon) calculator based on the arduino[1], that has inspired a large following due to its simplicity and ease of use. The MPGuino calculates fuel consumption based on a calibrated fuel flow and car speed.

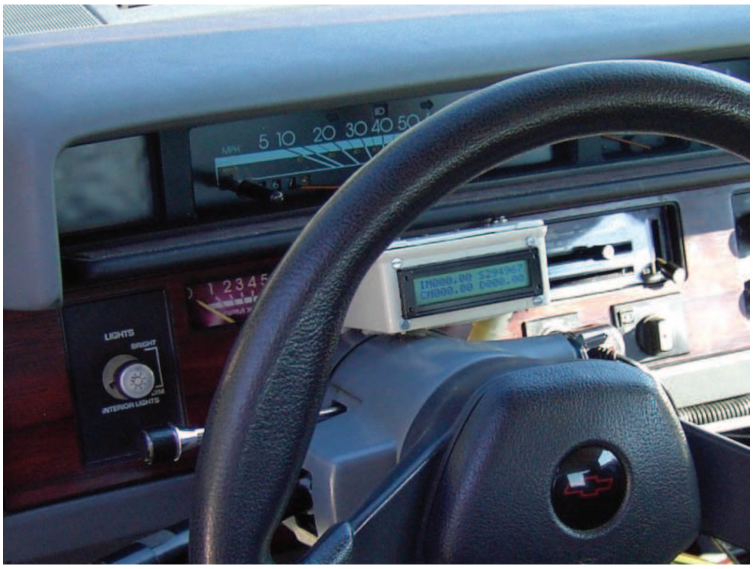


Figure 2.1: MPGuino

Unfortunately, the arduino has several structural defects which prevent it from being used for our needs, not least of all the choice of an basic model ATMega chip which has insufficient memory and peripherals.

There are several commercially available data loggers available, but these have the disadvantage of:

- being closed source
- being expensive
- having limited expansion

2.1 Microprocessors

A microprocessor puts most or all of a computer's CPU onto a single integrated circuit. The microprocessor has seen incredible progress since its introduction forty years ago. They see use in every single facet of life, from portable music players to computer peripherals to aerospace control systems.

This project uses a microprocessor to compute the optimal control.

2.2 Microcontrollers

Microprocessors are much like microcontrollers. In fact, a microcontroller can be seen as a microprocessor core with peripheral support, such as analog-to-digital converters, serial port protocols, etc... Today, a microcontroller has an impressive array of sensors and speaks a number of bus protocols.

Due to the addition of the peripherals, in general they do not perform at quite such dizzying speeds as pure microprocessors. However, they can throttle back to the point that they are only consuming nW of power.

While there are still speed gains to be made, and floating point math is to be avoided due to the lack of a floating processing unit, it is fully sufficient to use a microcontroller as the foundation of our sensor suite.

As such, an Atmel ATMega644p forms the core of the system. Chip speed is set by an 8Mhz internal RC oscillator. The microcontroller was programmed in C. The chip was chosen as it is relatively easy to develop, has a low price, requires no external components (thus saving development time), and has an active user community (www.avrfreaks.net).

Future versions will be built around the Atmel XMEGA, which employs the same instruction set as the ATMega series, but has much more advanced peripherals and a much faster (32MHz) and more stable (<.1%) RC oscillator. This allows true 32-bit timers, along with a much more accurate timestamp.[1]

Project source code can be found online at www.eissq.com[44]

2.3 Sensor package

The sensor package is made up of 4 sensors:

1. Three-axis accelerometer
2. GPS
3. Fuel injector timer (for measuring fuel flow and engine position)
4. Odometer

Note 5 *While neither a gyroscope nor a compass is used in the sensor package due to the requirement for a low-cost design, the observer detailed below was nonetheless designed with the possibility of adding these sensors. If gyroscope or compass data is available, it can be directly used without modification.*

[1] This version exists today, but testing has not yet been completed.

2.3.1 GPS

GPS, first started by the American military in the 1970s, has in the last 10 years revolutionized geolocalization. Its weaknesses are largely due to a non-gaussian error (due in part to the fact that only satellites overhead can be seen) and the inability to penetrate buildings and earth, meaning that a GPS fix is oftentimes unavailable in urban canyons, and always unavailable in tunnels. GPS also suffers from a lack of (relative) precision, only being accurate to sub-meter scales.

GPS works on the principle of triangulation. The GPS receiver is based on the very simple idea that it can calculate absolute distance from signal propagation time, i.e. the difference in time between when a signal was sent from the satellite and when it was received by the receiver. With the signal propagation time the receiver can calculate distance to the the satellite's known position. Geometrically, the intersection of any three lines is sufficient to place the receiver in 3-dimensional space so the receiver seeks out three known positions in order to have a fix. (In theory. In reality, the GPS receiver must solve an equation with 4 unknowns, as an absolute time solution must also be found before the receiver can calculate a signal's Δt. Thus, in practice a GPS receiver must have 4 satellites visible in the constellation before it can return a 3-D fix.)

Trivial as the idea may seem, the realities of compensating for time compression due to a satellite's relativistic speed with respect to the Earth, and time dialation due to proximity to Earth's gravity well are, to greatly understate, complicated.

The US is constantly upgrading the GPS network with new and better satellites, and many of GPS's inherent problems will be done away with by the promising new EU-sponsored Galileo system, but at the time of this dissertation Galileo was still many years away due to inter-EU squabbling. If it ever flies, it will herald in a new age of global positioning, as the accuracy and precision are far higher than GPS.

For our research, we used the San Jose Technology FV-M8 GPS. This particular model was very popular with hobbyists and researchers because of its 5Hz update rate and low price (<$50). While better GPSes have appeared on the market since, this one fulfilled needs.

However, if a new GPS unit were to be selected, we would take care to use one that output the raw GPS data, instead of only the filtered data. The filtered data poses problems as by its very nature the filtered data always lags behind the real-time state. If the raw GPS data were available, the position could be calculated as part of the observer model, allowing the Kalman Filter to do the work of filtering noise. This would help capture the true dynamics of the system, as manufacturers are loath to release the proprietary data of how they regulate their filters.

Care must be taken with GPS data to properly interpret and filter the measurement data. As show in Fig. 2.2, the GPS can have significant error, even after it has had complete 3-D fixes with $>$ 4 satellites. Ostensibly, the GPS delivers an estimation of the tracking accuracy, the HDOP, but in our experience this value is to be taken skeptically, as we have already seen the GPS get "dynamically stuck" at a clearly wrong series of coordinates for extended periods of time, all while the GPS reports a very low HDOP.

2.3.2 Accelerometer

Current accelerometer technology is MEMS. From Wikipedia:

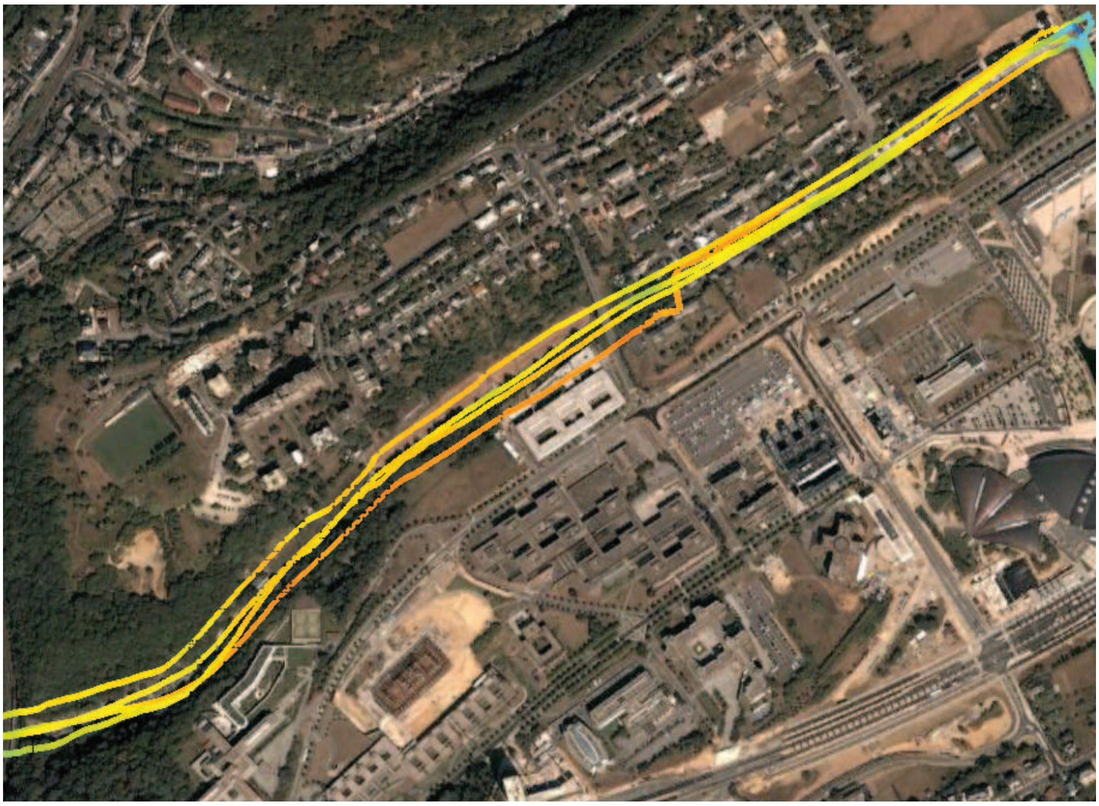


Figure 2.2: GPS repeatability errors (Plotted with GET, Appendix F)

> "Conceptually, an accelerometer behaves as a damped mass on a spring. When the accelerometer experiences an acceleration, the mass is displaced to the point that the spring is able to accelerate the mass at the same rate as the casing. The displacement is then measured to give the acceleration."

Most accelerometers work *in-plane*, so they can only measure acceleration in one dimension. A 3-D accelerometer is made of 3 in-plane accelerometers mounted in one package, each pointing along a cartesian axis.

Consumer-grade accelerometers are of the piezoelectric type. When a force is applied to a piezoelectric crystal, a voltage is induced. By transducing this voltage the sensor can output an acceleration value. Currently, more and more accelerometers have direct digital outputs, sparing the designer the difficulty of creating a precision analog circuit.

The switch from analog to digital signals saves considerable time in the design and implementation of the circuit. Digital signals reduce— or completely eliminate— the need for intensive noise-rejecting circuit simulation and design.

Indeed, at low data transmission speeds, the circuit pathway is not important. In fact, the only significant consideration is proper capacitance and grounding of the Vcc and GND lines, respectively.

We used the STMicro LIS3LV02D three-axis digital accelerometer. We chose this accelerometer in particular because it has digital outputs, in addition to having class-leading resolution (12-bits).

The accelerometer uses the TWI bus (sometimes called by its trademarked name I2C) to communicate with the microcontroller at 500kbps, more than sufficient for the 40Hz refresh rate. Data is timestamped for future use.

Note 6 *The accelerometer can also be used to measure road quality, which would give a better estimation of rolling resistance.[3]*

2.3.3 Odometer

The total distance the car has traveled is essential to high-quality observability. The GPS does not have sufficient reliability— not accuracy, nor resolution, nor speed— for this. *Odometry* is the use of data from sensors to measure a change of position. All cars have some form of odometer or another. The oldest cars have mechanical odometers that are driven by a spinning cable attached to a transmission gear, but modern cars use electric signals, usually derived from the ABS system.

The odometer sensor in cars is usually a toothed wheel that rotates past a hall sensor. This contactless sensoring system is very robust, does not wear out, is unaffected by dirt and grime, and gives high-quality signals. Its largest drawback is that the sensor teeth must go past the hall sensor quickly enough to trigger it. This speed threshold tends to be around 3-5 kph.

Many, but not all, modern cars have a Vss (vehicle speed sensor) signal, which is a PWM signal sent over a dedicated line. This signal is calibrated to give x pulses per y kilometers, and is often used by a stereo's speed-sensitive volume control feature or a GPS navigation system's dead-reckoning module, although cars can also use it to drive the speedometer.

Note 7 *The aspiring tinkerer who would like to install a data logger such as is described here is advised not to use the ABS signal directly, as it is a very high impedance signal and there is great risk of perturbing the ABS system, with unknown consequences to safety.*

In this case, we take the 12V Vss PWM signal and use a voltage divider to bring signals down to acceptable levels for the μC.

While this is in general a low-noise signal, care must be taken. Data is as accurate as the wheel radius model. For simplicity, we assume that wheel radius is a constant, however this is a simplification, as not only does it decrease with increasing car weights (flattening of the wheel), but it also varies based on air pressure (function of atmospheric pressure, tire temperature, and centripetal acceleration).

In addition to these inaccuracies, jitter in the μC real-time clock can influence speed observations.

In one test case, the car's Vss was non-functional, so we were forced to develop an interface to the car's CAN bus. This is detailed in Appendix D.

2.3.4 Fuel injector timer

The μC measures the time that the fuel injector opens, and for how long it stays open. The result is that we have two measurements with one sensor: engine angular distance (explained in Sec. 2.3.4.1) and fuel volume injected (explained in Sec. 2.3.4.2).

We timestamp the rising and falling edges of the signal, adding an opto-isolator in order to ensure that the μC can never have an adverse effect on the vehicle's ECU. All negative voltage components are blocked by a diode, thus protecting the circuitry from reverse voltage spikes. In order to protect the car ECU from short circuits, a current limiting resistor is placed immediately at the voltage shunt.

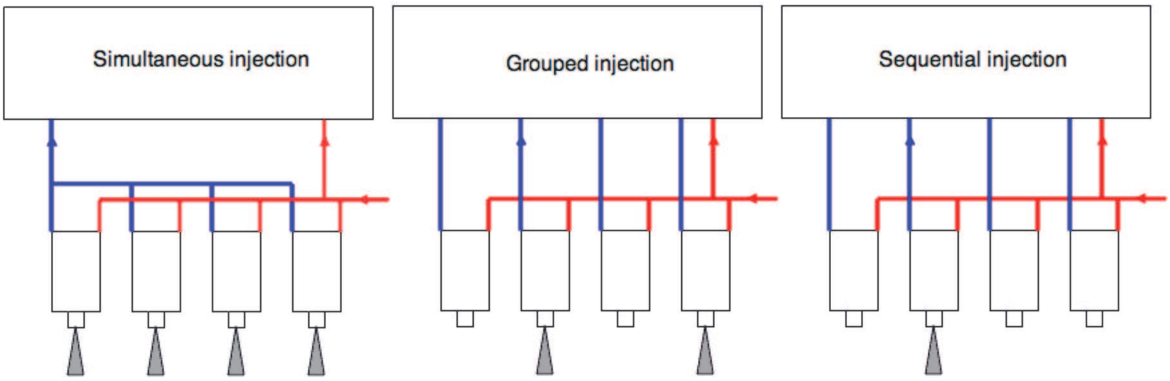
2.3.4.1 Engine angular distance/velocity

Properly controlling the engine demands measuring its position with a great deal of accuracy. The engine position is measured through multiple sensors in order for the ECU to properly time fuel injection, spark, etc... Sensors such as the camshaft position sensor are robust and accurate, and can be excellent choices for measuring angular position.

However, the engine angular position can also be measured by sensing voltage pulses in the fuel injection or in the spark plug wires. This method lacks somewhat in resolution, as it can only give position in multiples of π (or thereabouts, as explained in the following).

The position can also be measured optically with a colored dot on a rotating element of the engine, for instance the main engine pulley. However, installing the optical reader is probably more trouble than it's worth since there are so many other, excellent sources.

While we cannot regularly sample the engine's angular displacement, we can know at each fuel injection impulse that the engine has returned to its previous orientation. How many revolutions that is depends on the car. Contrary to expectations, in a 4-cycle engine a single fuel injector can inject multiple times during one complete combustion cycle.



(a) Simultaneous (b) Grouped (c) Sequential

Figure 2.3: Fuel injection schemes

- Simultaneous (Fig 2.3a): In this scheme, all injectors fire at the same time, and thus each injector pulse indicates that the engine has rotated $\dfrac{4\pi}{\#\text{ of cylinders}}$ radians.
- Grouped (Fig 2.3b): In this scheme, injectors fire in groups. Depending on the number of cylinders, this could be any factor of the total number of cylinders. For instance, some Jaguars V-12 engines use four groups of three injectors, whereas some older Citroen

engines have two groups of two. Each injector pulse indicates that the engine has rotated $\frac{4\pi}{\#\text{ of groups}}$ radians.

- Sequential (Fig 2.3c): Each fuel injector fires independently of the others. This is the most common in modern cars, but by no means ubiquitous. Each injector pulse indicates that the engine has rotated 4π radians.

Note 8 *Some engines can switch between injector grouping styles, depending on engine conditions. For instance, sequential injection may be used for low engine speeds, switching to simultaneous injection at high speeds. The same principle is used in changing from light loads to heavy loads. Care must be taken to identify if an engine is capable of switching injection schemes. If so, additional data will be required in order to identify the injector scheme. This information can come from the observer or from the OBD-II interface, to name two possibilities.*

2.3.4.2 Fuel flow meter

There are varying ways of measuring fluid flow. It can be directly measured by impeller blade, turbine, or even ultrasound. Except for the last one, all these methods require direct contact with the fluid being measured, which in the case of gasoline is difficult due to its reactivity, and the last one itself is prohibitively expensive. In any case, each of these requires installing extra equipment in the vehicle, which would understandably be unacceptable to many owners.

Having already encountered these problems, the online community explored the technique of using the fuel injector pulse width to measure fuel economy. This technique is accurate to ±1%. [17], and is thus well within the bounds of reasonable error.

Note 9 *All these techniques suffer from the fact that they measure volume, and not mass. This is a non-negligible difference, but unfortunately short of measuring temperature in the fuel tank and compensating based on a particular blend of gasoline's coefficient of thermal expansion, there is little that can be done about this. Fortunately, this can be disregarded as the relative efficiencies are of interest here. They will stay the same and so the impact will not be major.*

This method infers fuel flow by measuring fuel injector pulse duration. A fuel injector is effectively a needle (*pintle*) that plugs a small hole (*discharge orifice*). When current flows through an electromagnetic coil, the needle is pulled back from the hole, allowing fuel to flow through. When current stops flowing, a spring pushes the needle snug with the hole, stopping fuel flow. (See Fig. 2.4)

While measuring the voltage pulse applied to the injector neglects some factors, such as opening and closing speed, opening and closing hysteresis, variations in fuel pressure, temperature, density, etc..., the reality is that simply measuring the beginning and end of the positive voltage purposes are quite consistent. There is certainly an unknown bias, but these can be measured and other errors are absorbed into the model and have no effect on the final control.

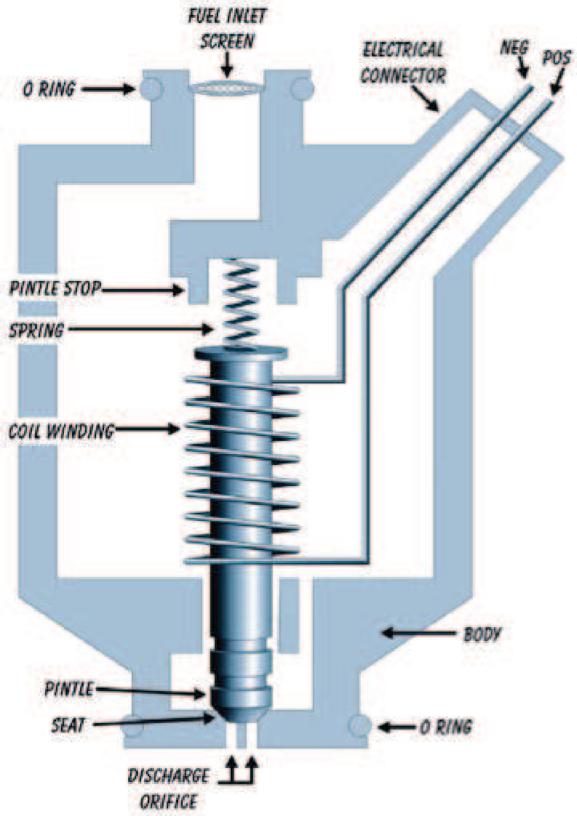


Figure 2.4: Fuel injector cutaway

Note 10 *With the advent of peak-injectors, which use a sudden current surge to rapidly open the injector and then use PWM to keep it open without burning the coil, this technique will have to be modified.*

Furthermore, this technique does not work with direct injection diesel engines, as the fuel pressure is no longer constant, violating the principle that fuel quantity injected is proportional to injector opening duration.

A possible solution to both problems could be to install an additional sensor installed in order to measure fuel flow in the fuel lines.

Another very strong possibility is to measure the oxygen sensor and MAF(mass airflow sensor) instead of the fuel injector timing pulse, as this method can compensate for changes in density, changes in fuel blend (95/98 octane, summer/winter, ethanol/gasoline, etc...) , and works as well with diesel fuel as gasoline.

2.4 Other sensors

2.4.1 Gyroscope

Inertial navigation systems[22], as they pertain to car navigation, have been studied in countless research papers.[47] In most of these cases, a gyroscope is used, however we opted not to use one due to costs. Commercially available gyroscopes fall into three categories:

1. Mechanical gyroscopes— These are built around a spinning (gyrating) mass. They are large, bulky, require separate power supply, expensive, and prone to failure.

2. MEMS— Micro Electro-Mechanical Systems. These are not true gyroscopes. They are composed of two three-axis accelerometers mounted to a digital signal processor or microcontroller of some kind. These gyroscopes integrate angular acceleration to arrive at angular position. The reference vector is given by gravity.

   This is generally effective, as average acceleration can be assumed to be zero in the long run, but has serious shortcomings when instantaneous acceleration is large. Furthermore, accelerometers tend to be noisy and have low resolution, so noise on the level of several milli-gravity is normal. Inertial orientation as resulting from the double integration is therefore particularly prone to error.

3. Ring laser— The *nec plus ultra* of gyroscopes, these are extremely expensive, and thus out of range a project that hopes to have wide-scale acceptance.

2.4.2 Compass

Accurate digital compasses are starting to see wider acceptance, finding their way into low-end consumer electronics. While in the end, they were not necessary for the desired level of accuracy, they are still a promising technology and can only serve to improve results, once the price comes down a little more.

Note 11 *The author feels that it should be possible to geolocate a car with only a DEM (Digital Elevation Map), compass, and odometer. This exercise in pattern matching is left as a difficult, but challenging, exercise to the reader.*

2.5 Additional equipment

2.5.1 Data storage

All data is stored on an SD card. MMC and SD cards have an SPI compatibility mode. (Note, the higher capacity (>2GB) SDHC cards do **not**.) This compatibility mode, well documented online, gives a basic access to the SD card over the SPI bus, albeit at reduced speeds.

In order to reduce computational complexity, some liberties are taken with the SD card's FAT16 file system. Specifically, a large file of several GB (2GB is the maximum size permitted in FAT16), initialized to zeros, is saved on the card. Then, data is logged sequentially in 512 byte blocks, starting at the beginning of the file. Because this file is not fragmented, the sequential writes always write *within* the memory-space allocated to the file.

Data retrieval is simple. The original file simply needs to be copied to the computer, and then the unwritten space pared off. Data accumulates at several megabytes a minute, so a 2GB card can be used for several months for daily commuter usage.

Further details can be found in Appendix C.1.4.

2.5.2 Digital Elevation Map

The DEM (Digital Elevation Map) is a mapping of points along the road. In this particular case, a shapefile formatted document was provided by the Luxembourgish mapping agency. This document included all roads in Luxembourg, with an altitude precision of 1m.

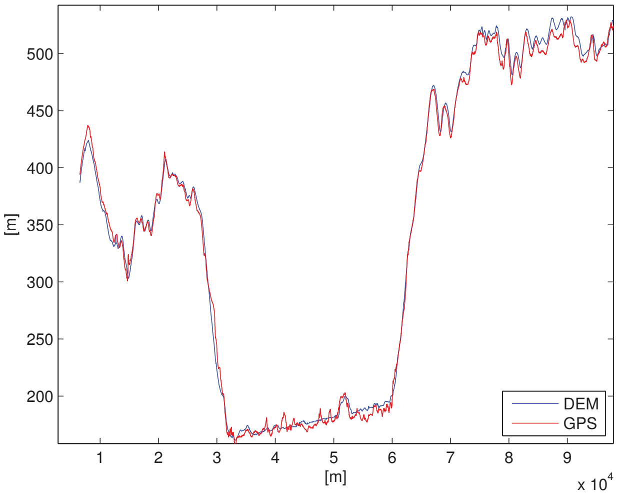


Figure 2.5: DEM vs. GPS along a trajectory

Unfortunately, today DEM data is difficult to come by, to the point of being impractical. In most cases, this data is sold by mapping companies at exorbitant prices, precluding wide-scale usage. (The authors were fortunate in Luxembourg to have the support of the national mapping administration, and thus have free access to quality DEM data.)

While the NASA developed SRTM data [34] is widely available, it is of limited resolution and, by its nature, misses nuances such as the difference between bridges and tunnels.

As free, high-quality mapping data [18] becomes more and more available, it can be used to better the model.

Figure 2.6 shows the advantages that the DEM can bring to correcting the GPS. Notice in specific the 20m discrepency at the 7000m mark. Clearly this sort of GPS error can wreak havoc with the slope estimation and as a consequence in the force estimation if not accounted for.

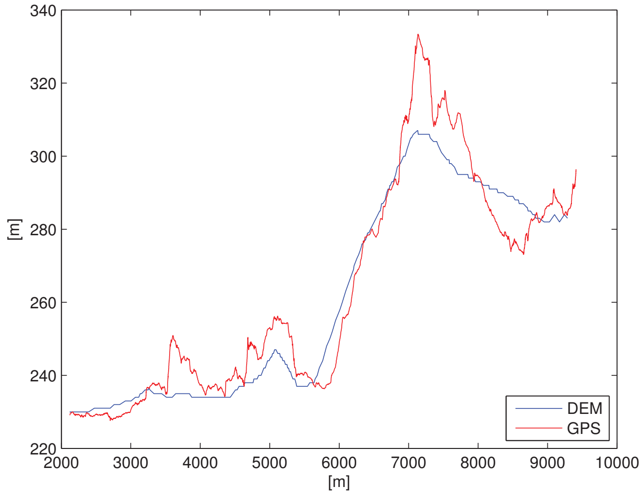


Figure 2.6: Disagreement between DEM vs. GPS along a trajectory

Chapter 3

Efficiency mapping

Contents

A cornerstone of any vehicle optimal control study is an engine efficiency model (or alternatively an engine specific fuel consumption (SFC) model). Unfortunately, this information is very difficult to come by. In previous studies, these models were assumed to be available ahead of time.[25, 51, 39]

In reality, this is not the case. Car manufacturers are notoriously tight-lipped and guard their data jealously. Furthermore, engine data alone does not tell the whole story, as it is ideal data from an ideal engine under ideal conditions. As any driver knows, the manufacturer reported fuel economy are vary considerably in real-world situations.

For an optimality study, it is essential to know the true system efficiency, which cannot be predicted from engine data alone. Many other car-specific factors come into play, such as transmission losses, rolling resistance, air resistance, etc... The values, when they can be found, are oftentimes unreliable (e.g. coefficients of air resistance) and have limited usefulness.

A better optimality approach would take into account the real-world parameters of the individual vehicle in question. This approach should be able to adapt to changing circumstances, such as a mass change when a passenger gets out of a car or an air resistance change when a convertible top being is put down. This approach also would allow monitoring motor efficiency with time, opening the door to an early indication of fault development.

3.1 Model

We recall the model developed in Section 1.2.2, using the variables highlighted in Table 1.1.

$$
\begin{array}{rcl}
\dot{x}_i &=& v_i \\
\dot{v}_i &=& \frac{1}{M_a}\left(F_{thrust} - \text{sign}(v_i)\left(\frac{1}{2}\rho C_d A v_i^2\right.\right. \\
&& \left.+MC_{rr}\left(g\cos(\theta) + S_k v_i^2\right)\right) \\
&& \left.-Mg\sin(\theta)\right) \\
\dot{F}_{thrust} &=& 0 \\
\dot{\psi} &=& S_j v_i \\
\dot{\theta} &=& S_k v_i \\
\dot{N} &=& v_i\cos(\psi)\cos(\theta) \\
\dot{E} &=& v_i\sin(\psi)\cos(\theta) \\
\dot{D} &=& -v_i\sin(\theta) \\
\dot{S}_j &=& 0 \\
\dot{S}_k &=& 0 \\
\dot{E}_{in} &=& P_{in} \\
\dot{P}_{in} &=& 0 \\
\dot{\theta}_e &=& w_e \\
\dot{w}_e &=& 0 \\
\dot{b}_i &=& 0 \\
\dot{b}_j &=& 0 \\
\dot{b}_k &=& 0
\end{array}
\qquad
\begin{array}{rcl}
x_{i,m} &=& x_i \\
a_{i,m} &=& \frac{1}{M_a}\left(F_{thrust} - \text{sign}(v_i)\left(\frac{1}{2}\rho C_d A v_i^2\right.\right. \\
&& \left.\left.+MC_{rr}\left(g\cos(\theta) + S_k v_i^2\right)\right)\right) + b_i \\
a_{j,j} &=& S_j v_i^2 + b_j \\
a_{k,m} &=& -g\cos(\theta) - S_k v_i^2 + b_k \\
DEM_{\psi} &=& \psi \\
DEM_{alt} &=& -D \\
\theta_m &=& \theta \\
GPS_{lat} &=& N \\
GPS_{lon} &=& E \\
GPS_{alt} &=& -D \\
INJ_{duration} &=& E_{in} \\
INJ_{time} &=& \theta_e
\end{array}
\tag{3.1}
$$

3.1.1 Values used

Table 3.1 gives typical model parameter values. The car parameters are taken from published values for the 2004 Smart Roadster Coupé. Note that mass is considered to include an average 70kg driver with 5kg of clothing and personal material.

Parameter	Value
g	9.805
ρ	1.2
C_{rr}	0.010
C_d	0.39
M	920
M_a	948
A	1.93
r_w	0.2954

Table 3.1: Model parameters

3.1.2 Car model observer

3.1.2.1 Car model observability

It is reasonable to suppose that certain measurements will not be available, depending on sensor state and the current environment. For instance, the GPS fix can be lost in a tunnel or in urban canyons. We present here an observability study for the various loss of sensors. These results summarize a theoretical study of the differentiable system observability property, but can also be inferred from intuitive considerations.

- Loss of GPS: Clearly, without the GPS we lose observability for N, E, D[1]$, \psi$. However, observability is preserved for the rest of the model.
- Loss of accelerometer: As the model is completely observable with only the GPS and the fuel injector sensor, this only impacts the observability of the accelerometer bias. By inference, we see that the model works with only a two-axis accelerometer available.
- Loss of Vss: No impact on observability.
- Loss of DEM: No impact on observability.
- Loss of engine speed sensor: (Even though one sensor measures both engine speed and fuel flow, for questions of observability we separate the two, as it is possible to have independent sensors for both.) No impact on observability only if the transmission ratio is known at all times.

[1] D is a nuanced case. Without the GPS, it is impossible to fix the car's position sufficiently well on the map in order to be able to use a DEM.

- Loss of fuel flow sensor: Clearly, observability of energy flow is lost. Without energy flow, efficiency cannot be calculated, and so this is the most important sensor for our study.

Of course, it goes without saying that as we lose sensors we lose accuracy. For obvious reasons, we only consider losing one sensor at a time. The above observability study should not be used as a rationale for decreasing the number of sensors, but instead only for validating an approach which does not, for whatever reason, use all desired sensors.

3.1.2.2 Choice of observer

The choice of observer is paramount. The model is of a large dimension (17 states) and the final result, the efficiency η, involves dividing by a derivative of a measured state.

At first attempt, the UKF (Cf. A.3.2) was explored as a possible solution, but there were grave difficulties integrating the measurements into the model. In [33], Moullion provides a reasonable explanation of why this behavior is seen. He explains that the UKF's correction compares a physical mean, the measured value, with a sigma-space mean, the value calculated by the model non-linearity. Depending on this nonlinearity, there can be a strong deviation from the true mean, and it is difficult to transform the physical mean into the sigma-space.

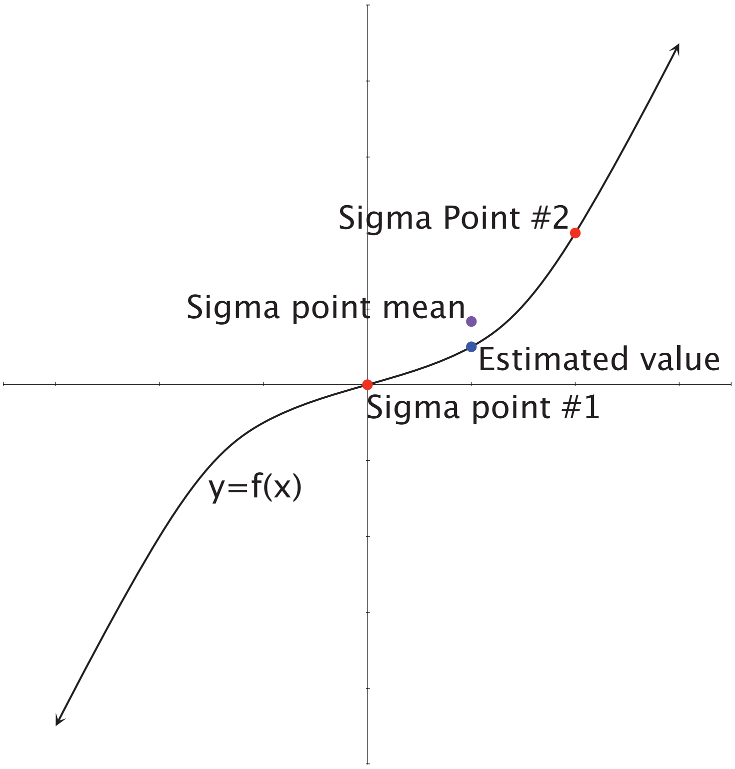


Figure 3.1: Sigma point averaging

If the true value equals the estimated value, we would expect that the system would apply no correction. However, as can be seen in Fig. 3.1, the sigma mean value $\hat{\mu}$ is not equal to the estimated value, and so there will be an error ϵ and in consequence a correction.

Specifically, the problem with the UKF came from the need to estimate θ as accurately as possible in order to likewise estimate F_{thrust} as accurately as possible. (Cf. Sec. 1.4) Since the acceleration measurements are the most important measurements for estimating θ, this difference between the sigma space mean and the cartesian space mean provokes an estimation error that reverberates through the entire model and rapidly makes the results useless.

It is outside the scope of this dissertation to develop a generalized approach to converting measurements into the appropriate coordinate system in order to fully apply UKF. Thus in the end, in spite of the UKF's superior stability, its computational costs and sensitivity to the measurement model rules it out.

The classical EKF (Cf. A.3.1), although theoretically not as robust as the UKF, still gives satisfactory performance. As such, henceforth we base our observer on this observer. (In Sec. A.4, we include a discussion of the Adaptive High-gain Extended Kalman Filter[45, 8] as a possible continuation of the EKF approach.)

We recall that this model is in continuous-discrete time, as described in Section A.2.3

$$\left\{ \begin{array}{rcl} \dot{\mathbf{x}} & = & f(\mathbf{x}(t), \mathbf{u}(t), t) \\ \mathbf{y}_k & = & h(\mathbf{x}_k, \mathbf{u}_k, k) \end{array} \right.$$

We recall that this is a CD-EKF, as described in Section A.3.1.3, and using the shorthand described in (A.11)

Prediction:

$$\begin{array}{rcl} \hat{\mathbf{x}}_k^- & = & \hat{\mathbf{x}}_{k-1} + \displaystyle\int_{t_{k-1}}^{t_k} \frac{d\mathbf{x}}{dt} dt \\ \hat{\mathbf{P}}_k^- & = & \hat{\mathbf{P}}_{k-1} + \displaystyle\int_{t_{k-1}}^{t_k} \frac{d\hat{\mathbf{P}}}{dt} dt \end{array}$$

where $\dfrac{d\hat{\mathbf{x}}}{dt} = f(\mathbf{x}, \mathbf{u})$ and

$$\frac{d\hat{\mathbf{P}}}{dt} = (\mathbf{A}_c \hat{\mathbf{P}} + \hat{\mathbf{P}} \mathbf{A}_c^T + \mathbf{Q}_c) \quad \Big| \quad \hat{\mathbf{P}}|_{t_{k-1}} = \hat{\mathbf{P}}_{k-1} \tag{3.2}$$

and $\mathbf{Q}_c$ is a positive-definite matrix in $\mathbb{R}^{n_x \times n_x}$

Correction:

$$\begin{array}{rcl} \mathbf{K}_k & = & \hat{\mathbf{P}}_k^- \mathbf{H}_k^T (\mathbf{H}_k \hat{\mathbf{P}}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \\ \hat{\mathbf{x}}_k & = & \hat{\mathbf{x}}_k^- + \mathbf{K}_k \left(\mathbf{y}_{t_k} - h(\hat{\mathbf{x}}_k^-, \mathbf{u}) \right) \\ \hat{\mathbf{P}}_k & = & (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \hat{\mathbf{P}}_k^- \end{array} \tag{3.3}$$

where $\mathbf{R}_k$ is a positive-definite matrix in $\mathbb{R}^{n_y \times n_y}$

3.1.2.3 Jacobian

The CD-EKF requires the Jacobian $\mathbf{A}_c$ of the transition equations, and the Jacobian $\mathbf{H}_k$ of the measurement equations.

The Jacobian $\mathbf{A}_c$ can be numerically calculated, but finding the formal Jacobian for the system described by eqns. (3.1) is not complicated. Even if Jacobians seem complicated to some, they are straightforward to calculate, can be validated numerically and/or formally, and form the foundation of the EKF, one of the most widely-used observers in modern science.

$\mathbf{A}_c$ is a $(\mathbf{17} \times \mathbf{17})$ matrix with zeros everywhere, except at the following indices:

$$\mathbf{A}_c = \begin{pmatrix}
0 & A_{1,2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & A_{2,2} & A_{2,3} & 0 & A_{2,5} & 0 & 0 & 0 & 0 & A_{2,10} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & A_{4,2} & 0 & 0 & 0 & 0 & 0 & 0 & A_{4,9} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & A_{5,2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_{4,10} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & A_{6,2} & 0 & A_{6,4} & A_{6,5} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & A_{7,2} & 0 & A_{7,4} & A_{7,5} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & A_{8,2} & 0 & 0 & A_{8,5} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_{11,12} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_{13,14} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix} \tag{3.4}$$

where

$$\begin{aligned}
A_{1,2} &= 1 \\
A_{2,2} &= -\mathrm{sign}(v_i)\frac{1}{M_a}(\rho C_d A v_i + 2C_{rr} M S_k v_i) \\
A_{2,3} &= \frac{1}{M_a} \\
A_{2,5} &= \frac{M}{M_a} g(C_{rr}\sin(\theta) - \cos(\theta)) \\
A_{2,10} &= -\frac{M}{M_a} C_{rr} v_i^2 \\
A_{4,2} &= S_j \\
A_{4,9} &= v_i \\
A_{5,2} &= S_k \\
A_{5,10} &= v_i
\end{aligned}
\qquad
\begin{aligned}
A_{6,2} &= \cos(\psi)\cos(\theta) \\
A_{6,4} &= -v_i \sin(\psi)\cos(\theta) \\
A_{6,5} &= -v_i \cos(\psi)\sin(\theta) \\
A_{7,2} &= \sin(\psi)\cos(\theta) \\
A_{7,4} &= v_i \cos(\psi)\cos(\theta) \\
A_{7,5} &= -v_i \sin(\psi)\sin(\theta) \\
A_{8,2} &= -\sin(\theta) \\
A_{8,5} &= -v_i \cos(\theta) \\
A_{11,12} &= 1 \\
A_{13,14} &= 1
\end{aligned}$$

Likewise for the output equations Jacobian, **H**:

$$
\begin{array}{rcl}
\mathbf{H}_{dist} & = & \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \\
\mathbf{H}_{acc} & = & \begin{pmatrix} 0 & H^{acc}_{1,2} & H^{acc}_{1,3} & 0 & H^{acc}_{1,5} & 0 & 0 & 0 & 0 & H^{acc}_{1,10} & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & H^{acc}_{2,2} & 0 & 0 & 0 & 0 & 0 & 0 & H^{acc}_{2,9} & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & H^{acc}_{3,2} & 0 & 0 & H^{acc}_{3,5} & 0 & 0 & 0 & 0 & H^{acc}_{3,10} & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \\
\mathbf{H}_{\psi} & = & \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \\
\mathbf{H}_{\theta} & = & \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \\
\mathbf{H}_{GPS} & = & \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \\
\mathbf{H}_{S_j} & = & \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \\
\mathbf{H}_{INJ} & = & \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \\
\mathbf{H}_{DEM} & = & \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}
\end{array}
$$

where

$$
\begin{array}{rcl}
H^{acc}_{1,2} & = & -\frac{1}{M_a}\operatorname{sign}(v_i)(\rho C_d A v_i + 2C_{rr} M S_k v_i) \\
H^{acc}_{1,3} & = & \frac{1}{M_a} \\
H^{acc}_{1,5} & = & \frac{M}{M_a} C_{rr} g \sin(\theta) \\
H^{acc}_{1,10} & = & -\frac{M}{M_a} C_{rr} v_i^2 \\
H^{acc}_{1,15} & = & 1
\end{array}
\qquad
\begin{array}{rcl}
H^{acc}_{2,2} & = & 2S_j v_i \\
H^{acc}_{2,9} & = & v_i^2 \\
H^{acc}_{2,16} & = & 1
\end{array}
\qquad
\begin{array}{rcl}
H^{acc}_{3,2} & = & -2S_k v_i \\
H^{acc}_{3,5} & = & g\sin(\theta) \\
H^{acc}_{3,10} & = & -v_i^2 \\
H^{acc}_{3,17} & = & 1
\end{array}
$$

See Section A.5 for an explanation of how to create $\mathbf{H}_k$. See Algorithm 1 for pseudo-code of how $\mathbf{H}_k$ is used in practice.

3.1.3 Observer pseudo-code

The original code was written in Matlab, but is easily adaptable to any language.

Algorithm 1: Observer loop

```
while forever do
    while no new measurement do
        integrate $\frac{d\hat{\mathbf{x}}(t)}{dt}$ and $\frac{d\hat{\mathbf{P}}}{dt}$
    end
    $\hat{\mathbf{x}}_k^- \leftarrow \hat{\mathbf{x}}(t_k)$
    $\hat{\mathbf{P}}_k^- \leftarrow \hat{\mathbf{P}}(t_k)$

    Measurement
    $\mathbf{y}_{t_k} \leftarrow$ event(s) at time $t_k$
    $\Delta t \leftarrow t_k - t_{k-1}$

    Correction
    construct new measurement Jacobian, $\mathbf{H}_k$
    construct new measurement covariance, $\mathbf{R}_k$
    $\mathbf{K}_k \leftarrow \hat{\mathbf{P}}_k^- \mathbf{H}_k^T (\mathbf{H}_k \hat{\mathbf{P}}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1}$
    $\hat{\mathbf{P}}_k \leftarrow (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \hat{\mathbf{P}}_k^-$
    $\hat{\mathbf{x}}_k \leftarrow \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{y}_{t_k} - h(\hat{\mathbf{x}}_k^-, \mathbf{u})$

    Update loop
    $\hat{\mathbf{x}}(t_k) \leftarrow \hat{\mathbf{x}}_k$
    $\hat{\mathbf{P}}(t_k) \leftarrow \hat{\mathbf{P}}_k$
    $k \leftarrow k + 1$
end
```

We use a Runge-Kutta-4 integrator for integrating the continuous model and covariance differential equation systems. Note, however, that the interval itself is Δt, and as such changes every iteration, not for numerical reasons as in classical adaptive step versions of RK, but because of the asynchronous measurements.

There is not much to gain by using a variable-step integrator, such as Runge-Kutta-4-5. Due to the nature of the observer loop, there tend to be multiple integration intervals between each update of a particular variable. For example, while awaiting a new acceleration measurement, the system will have been integrated multiple times across smaller intervals as other measurements, such as fuel burn, arrive.

3.1.4 Tuning

It can be quite complicated to tune a model with 17 states. In the following, several strategies are provided for tuning the observer.

3.1.4.1 Tuning state model covariances

- x_i: the kinematic model is exact, the model noise is thus low.

- v_i: the kinematic model is exact, the model noise is thus low.

- F_{thrust}: the model is obviously incorrect— $\dot{F}_{thrust}$ is supposed equal to 0— and moreover is also subjected to very rapid swings. The model error is quite high.

  Note 12 *It is possible to improve the frequency response of $\dot{F}_{thrust}$ by using a local polynomial model, of the type $\dot{F}_{thrust} = \xi_1$, $\dot{\xi}_1 = \xi_2$, $\dot{\xi}_2 = \xi_3$, ... This type of model better represents variables that are likely to have rapid swings in state, as the modeled variable can vary like an* n*-order polynomial.*

  The cost of this improved response is the addition of dimensions to the state model, with a commensurate increase in processing costs. In this particular case, satisfactory results were obtained with the basic model.

- ψ: The car heading is dependent on a kinematic process, and in the long run the model is exact. (The car will always be on the road, always pointing in the direction the road goes.) The error is low.

- θ: The car pitch is dependent on a kinematic process, and in the long run the model is exact. (The car will always be sitting with all four wheels on the ground, always parallel to the road surface). The error is low.

- N, E, D: Northing, Easting, and Down are dependent on a simplified kinematic model (Coriolis effects are neglected), and thus the model is quite good. In the author's experience, a very low covariance noise value gives good results, as will be shown below.

- S_j: the horizontal road straightness coefficient has a poor model, $\dot{S}_j = 0$, but due to the very small nature of S_j itself the tuning value will be low on an absolute scale. See Note 12 for suggestions on how the model can be improved.

- S_k: the vertical road straightness coefficient has a poor model, $\dot{S}_k = 0$, but due to the very small nature of S_k itself the tuning will be low on an absolute scale. Moreover, we can be certain that S_k changes slowly (otherwise driving a car would be like riding on a roller-coaster) so S_k should be smaller than S_j. See Note 12 for suggestions on how the model can be improved.

- E_{in}: The model for E_{in} is based on the conservation of energy, and is thus exact. The model noise is low.

- P_{in}: The model for P_{in} is poor: $\dot{P}_{in} = 0$. Moreover, P_{in} is capable of high frequency swings (at the speed at which the driver depresses the pedal) and thus model noise is quite high. See Note 12 for suggestions on how the model can be improved.

- θ_e: The engine angular distance model is very good, indeed it is kinematically exact. As a consequence the model noise is very low.

- ω_e: The engine angular velocity should be considered to be a very noisy model, as 1) the engine angular distance measurement is very good so there is little risk to placing a high degree of confidence in the measurement and not in the model, and 2) when the transmission shifts or is put into neutral the engine changes speed very rapidly. See Note 12 for suggestions on how the model can be improved.

- b_i, b_j, b_k: The accelerometer biases should be considered to react glacially slowly, since the accelerometer bias is estimated consistently across multiple trips. There will be slight variations with different loading patterns, however. I.e. a car will tilt when passengers get in or out, thus provoking slightly different biases depending on passenger weight and distribution, so the biases must be allowed some room for movement.

3.1.4.2 Tuning measurement model covariances

- Accelerometer noise: in the $\hat{k}$ direction (upwards-downwards from the car), the accelerometer is far noisier than in the $\hat{\jmath}$ (side-to-side) and $\hat{\imath}$ (forward-backwards) directions. This is due to road noise from bumps, uneven surfaces, etc...

- GPS noise: GPS error is internally estimated by the GPS receiver, and expressed as HDOP (Horizontal Degree Of Precision), a value that is part of the data sent by the GPS unit. If using the NMEA protocol, this data is sent in the `$GPGGA` and `$GPGSA` packets.

  Unfortunately, HDOP was not reliable in absolute terms, as the estimated error was frequently underestimated. After losing a GPS fix and then regaining it at a later time, the GPS position error could be orders of magnitude greater than that predicted by HDOP.

  We suggest using a relatively high base value for the GPS noise covariance, and increasing it as a function of increasing HDOP.

- Vss noise: the Vss is a very clean signal, but the quantization of this signal can lead to errors. (See Fig. 3.2.) These quantization errors decrease with increasing time resolution, but unfortunately even at a time resolution of 125μs the error was higher than the author would have liked. Real-world results can be seen in Fig. 3.3. While this error can somewhat be smoothed out, it would be better to increase the sampling speed.

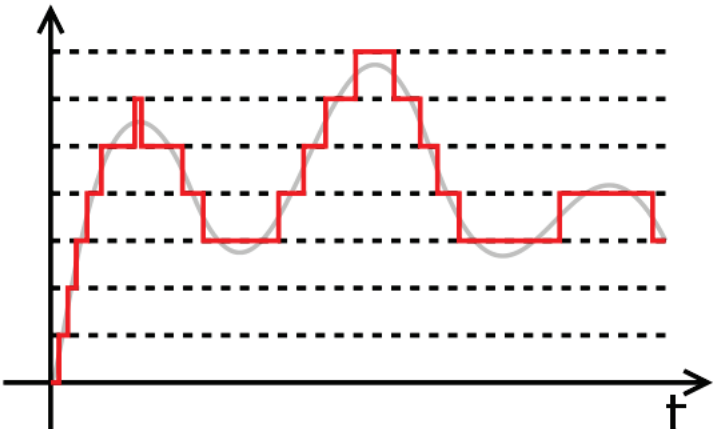


Figure 3.2: Quantized signal

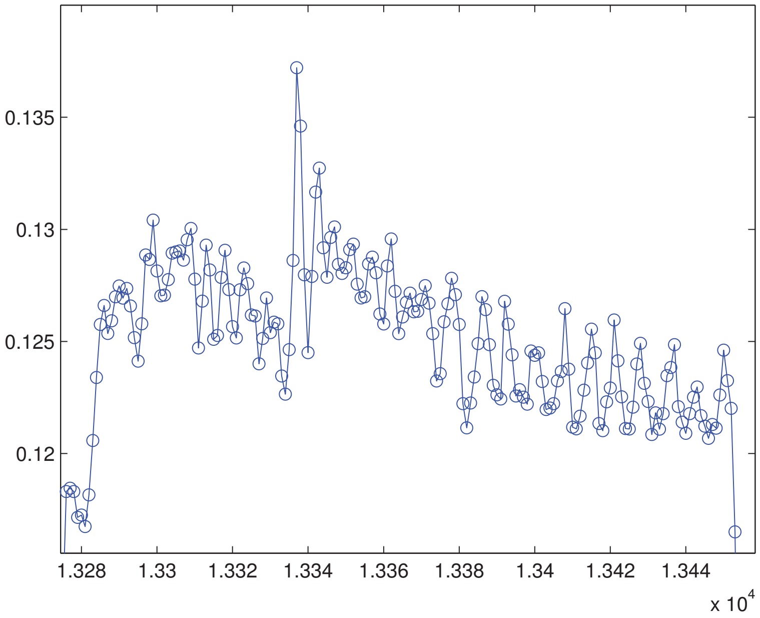


Figure 3.3: Vss signal

Due to limitations in hardware it was not possible to sample at a higher rate, although future versions of the data logger certainly could. 10μs resolution is probably the minimum required in order to eliminate quantization errors.

Quantization errors aside, it seems that a 1% odometer error is reasonable.

- Engine angular distance noise: this has the same quantization error problem as the Vss, although there is for all intents and purposes no error whatsoever in angular distance traveled. (Obvious, since the injector only fires at the same point in the cycle every time.) This value can be very low and still yield good results.
- If a DEM is present, the noise at the precise node is very low, but determining if one is at a node, and if not how far away one is from said node, can be difficult. We had good experience with a value roughly 1000 times smaller than the GPS.

3.1.4.3 Values used

We present here the values used in the above-described observer.

3.1.4.4 Normalization

Another approach to observer tuning is to normalize all variables. States and measurements that are considered noisier, or less reliable, are given higher relative values, and vice-versa for states and measurements that are more reliable.

Variable	Value
$Q(1,1)$	0.1
$Q(2,2)$	0.01
$Q(3,3)$	90000
$Q(4,4)$	0.01
$Q(5,5)$	1.5625×10^{-8}
$Q(6,6)$	0.0001
$Q(7,7)$	0.0001
$Q(8,8)$	0.00001
$Q(9,9)$	10^{-6}
$Q(10,10)$	$1/v_i^2 \times 10^{-9}$
$Q(11,11)$	0.01
$Q(12,12)$	50000
$Q(13,13)$	0.01
$Q(14,14)$	500
$Q(15,15)$	10^{-17}
$Q(16,16)$	10^{-17}
$Q(17,17)$	10^{-17}

Table 3.2: System covariance values

Variable	Value
R_{dist}	0.05
$R_{acc}(1,1)$	1
$R_{acc}(2,2)$	4
$R_{acc}(3,3)$	50
R_{ψ}	0.03
R_{θ}	0.03
$R_{GPS}(1,1)$	(GPS HDOP)$^2 \times 100$
$R_{GPS}(2,2)$	(GPS HDOP)$^2 \times 100$
$R_{GPS}(3,3)$	(GPS HDOP)$^2 \times 1000$
R_{S_j}	10^6
$R_{INJ}(1,1)$	1
$R_{INJ}(2,2)$	0.016
R_{DEM}	1

Table 3.3: Measurement covariance values

We used a non-normalized approach as it gives an absolute feel to the problem physics, but the reader might prefer the normalized approach as it is more abstract and focuses more on the computational math, and less on the application.

3.2 Data gathering

Data was gathered through normal car driving. Not requiring special actions on the part of drivers is an essential facet of the approach.

However, due to this constraint, a large part of the rpm vs. torque relationship is not explored. Specifically, low torque at high rpm is a relatively rare occurrence in normal driving. In fact, most normal drivers do not drive at high rpms (>4000rpm) no matter what the circumstances, and when they do it is only for brief periods of heavy acceleration, i.e. high torque. Nonetheless, these points are not strictly necessary, as the peak efficiency will not occur at these high rpm-low torque relationships.

Moreover, the optimal procedure will push the driver to explore high efficiency areas, at the expense of other zones, improving our model in the area that most needs it. In short, the most important area to characterize first is the zone of greatest efficiency.

In our experience, an hour of highway driving, along with several starts, was sufficient to fill a significant part of the curve, and have a reasonable curve fitting.

3.3 Results

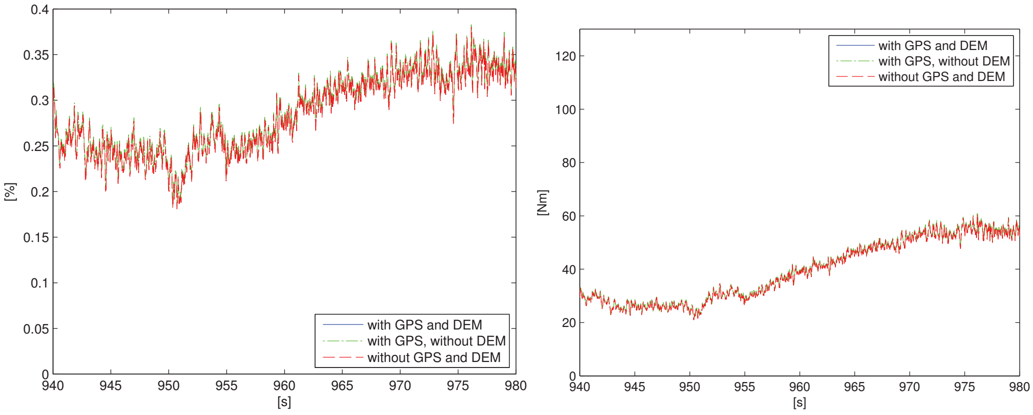
Several cars were selected for sampling. The specific car represented here is a Smart Roadster Coupé, with a 60kW engine that has been chip remapped at some previous point— when, how and with what was unknown to the owner— to a higher output. This provides a good case study as even the manufacturer does not know the performance of this particular engine. The additional cars give similar results (Sec. 3.6), although a detailed study is not presented.

Each result considers three cases: 1) GPS and DEM data were provided to the observer, 2) only GPS data is provided, and 3) neither GPS nor DEM data is provided. Let us point out that in this third degraded mode observability is no longer preserved for variables N, E, D, ψ. However, observability for the rest of the system is preserved, and thus we can still calculate efficiency.

Note that efficiency is only plotted for moments when the car's engine is propelling the car.

3.3.1 Highway

Figure 3.4 shows results gathered from data logged on a highway.



(a) Efficiency vs. time

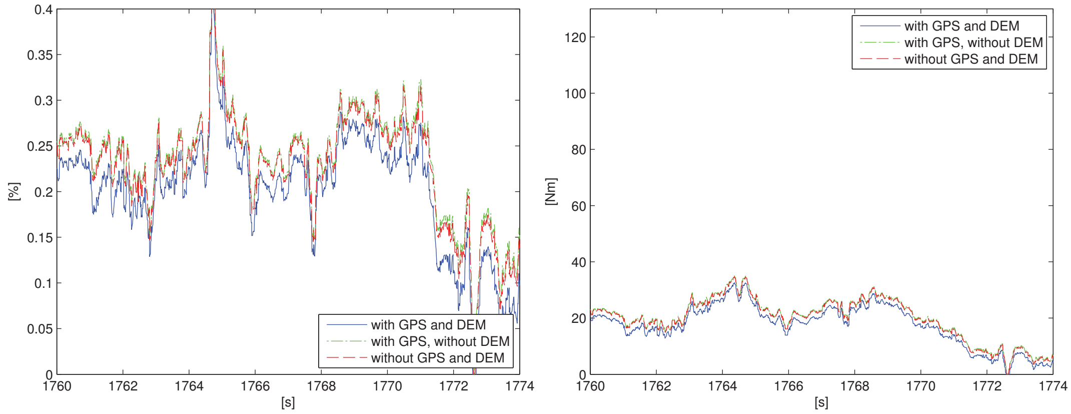
(b) Torque vs. time

Figure 3.4: Highway data

Remark that efficiency grows with increasing torque, which is to be expected.[2]

3.3.2 In-town

Figure 3.5 shows results gathered from data logged in a city.



(a) Efficiency vs. time

(b) Torque vs. time

Figure 3.5: In-town data

Again, remark that efficiency grows with increasing torque, which is to be expected. Also remark that the system is noisier, with transient efficiencies reaching unreasonable values. This is certainly due to the fact that in-town driving is more chaotic, with rapid shifts from acceleration to deceleration phases.

Nonetheless, the results are consistent with the previous highway example.

3.3.3 Maximum acceleration

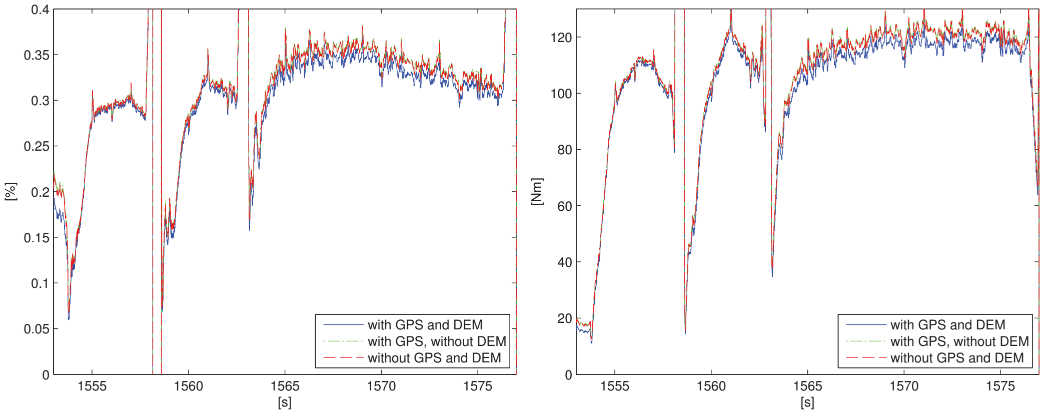
Figure 3.6 shows results when the car is undergoing maximum acceleration.

Efficiency is globally higher in maximum acceleration than the previous two examples. This is because there are less pressure losses behind the throttle body and thus the engine is not working as hard to intake air. However, efficiency here is already slightly suffering versus peak efficiency because at maximum power the car's engine management system injects additional fuel in order to help cool the cylinders. This additional fuel is not completely burned, and thus the engine has a lower fuel efficiency.

3.3.4 Efficiency map

An engine efficiency map, Fig. 3.7, is made by averaging together efficiency values as functions of rpm and torque. Note that the efficiency graph shows a continuous and smooth progression from low to high efficiency, and shows a sharp drop off to the right as the rpm

[2] The reason for this might not be immediately obvious to the lay eye. Efficiency is a function of many variables, but one important variable is pumping losses. As the throttle plate restricts the airflow into the engine, it creates a zone of low pressure behind the throttle body. The engine must work harder to create this vacuum, and so there is more energy lost due to this pressure differential.



(a) Efficiency vs. time

(b) Torque vs. time

Figure 3.6: Maximum acceleration data

reaches the redline (≈ 6300rpm, not shown in the figure). This is raw data from hours of driving, in urban and highway environments, and has not been post-processed in any way, except to average the data in each grid point, and that a grid point is not drawn if the standard deviation is too high (±5%).[3]

A sample efficiency map, Fig. 3.8, is given for reference.

3.3.5 Discussion

As can be seen, the data graphed in the three test cases coincide so well it is often difficult to distinguish between results with all sensors and results without GPS and/or DEM. Where these sensors make the biggest difference is in urban environments, with lots of sharp turns and acceleration/deceleration phases.

This shows that the efficiency model is very robust, in spite of the fact that the altitude estimation becomes unsatisfactory. Therefore, DEM and GPS data do not seem to be strictly necessary. This is most likely due to the fact that the absolute altitude data is not necessary. As the odometer permits estimating the car's acceleration, θ remains observable. More to the point, the altitude data is heavily filtered, as it is less reliable, and thus the estimation of F_{thrust} depends most heavily on the acceleration measurement.

If needed, one can eliminate these two sensors, reducing hardware costs and simplifying the programming task. However, despite the robustness of the observer, it is still preferable to use GPS and DEM when possible. While not dramatic, the results are clearly worse without the altitude data, and so as a consequence the efficiency map will be less precise.

It is interesting to note in these plots that the theoretical property of observability is clearly confirmed in practice. That is to say, despite the noise the whole state necessary for calculating η can be retrieved from the measurements and in consequence the observer can be said to perform to satisfaction. Even in the more complicated urban cycle, error remains

[3] A too high standard deviation means a too large variation of ω_e, T_e in the grid point, which means that it might not accurately reflect reality. As there are already many valid grid points, it does not harm the overall fitting to discard these points.

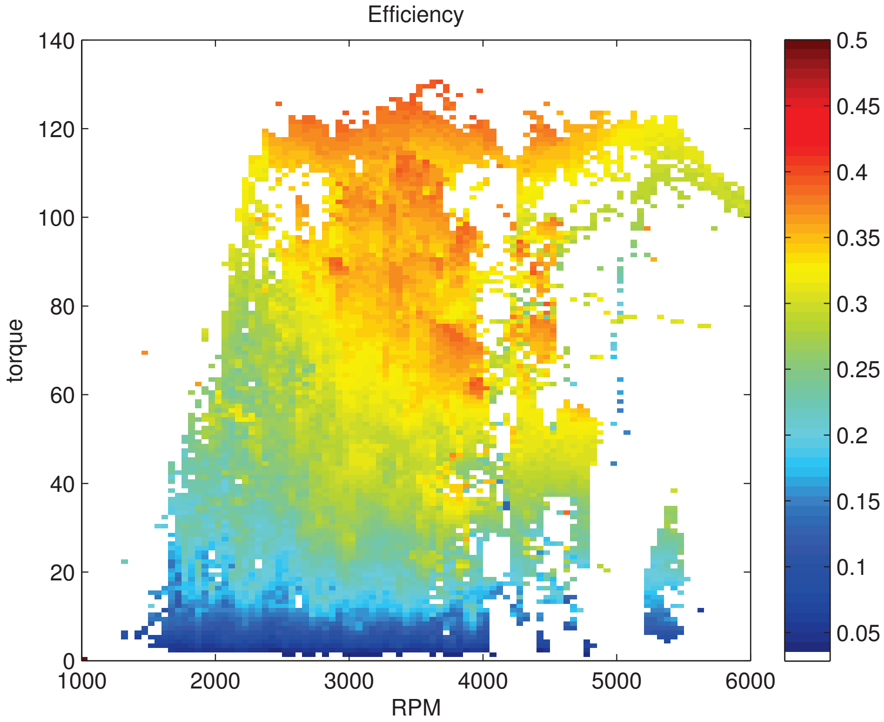


Figure 3.7: Efficiency map for Smart Roadster Coupé

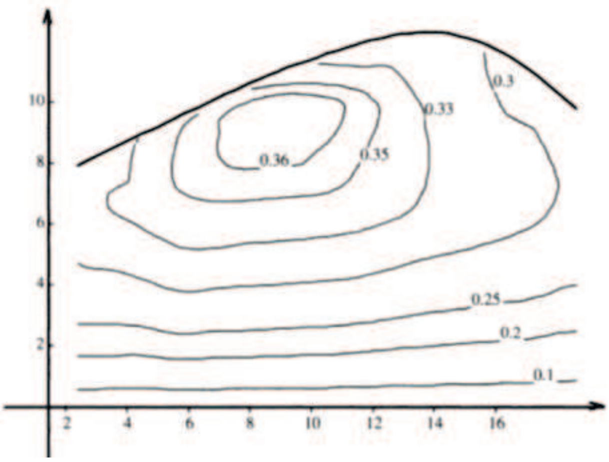


Figure 3.8: Sample efficiency map

sufficiently small.

It must be noted that the efficiency values are globally too high. (Thermal efficiency is generally in the 25-30% range, and this is before taking drivetrain losses into account.) This is not a major concern, as it indicates a bias in one or another of the car parameters. For instance, if C_{rr}, the coefficient of rolling resistance, is estimated too high then as a result

F_{thrust} will be overestimated, which in turn leads to a higher calculation of P_{out} and finally in consequence a globally higher efficiency. Or, if the amount of fuel injected per cycle is underestimated, then there is a percentage bias that affects the entire curve. However, as long as any optimality study uses these same values, it will lead to a similar optimal control.

3.4 Curve fitting

Analytic optimal control approaches require an efficiency or specific fuel consumption model represented by an equation that depends on the system state. We used the mesh presented in Fig. 3.7 as the raw data for the curve fitting. The curve fitting is not very sensitive to the mesh granularity, and so a mesh resolution that tracks relatively closely to a 100x100 grid is a good compromise between data point averaging and resolution. This led to grid squares 50[rpm] wide by 2[Nm] tall. Averages were only included as explained in the previous section: there must be a minimum number of points making up the average, and the standard deviation must be sufficiently small (±5% efficiency).

It is clear that that the empty areas seen in the unexplored torque-rpm-efficiency relationship will have a tendency to allow potentially undesirable fitting effects. The lack of data at these points can lead to a fitting with impossibly high efficiencies, especially in the case of a polynomial fitting.

These effects can be handled by 1) forcing the graph to conform to expectations (a car engine will always have one or two efficiency islands, and in modern passenger cars designed to run efficiently at 2000-4000rpm it is reasonable to suppose that there will never be an additional efficiency island to the extreme right), 2) cutting the fitting off at any minimum and setting all η outside this minimum to 0, or 3) seeking more data by performing additional experiments to better map this area.

While #3 would seem to make the most sense, in fact it would be seen as unacceptable by the average driver, and so this approach should be overlooked as impractical. #1 is interesting because it is easy programatically. It suffices to line the righthand border with a column of grid points with zero efficiency. #2 is most likely the best, though, as it allows the curve fitting to most faithfully model the area of greatest interest.

3.4.1 Efficiency fitting

Here we seek to fit the curve $\eta(\omega_e, T_e)$. Keeping in mind that the goal is to create an optimal control for minimizing fuel consumption, it is desirable to keep the model as simple as possible, by limiting the fitting polynomial order to the greatest practical extent. The resulting curve parameters are not very sensitive to the mesh granularity, nor to polynomial class.

Curves were fit by a classical least-squares algorithm, and the coefficient of determination, R^2 is given. In least-squares fittings, the problem is to find the coefficient vector **c** in

$$\mathbf{F}(\omega_e, T_e)\mathbf{c} = \eta$$

where **F** is a matrix with the rows made by the polynomial function we are looking to fit (in the cases below, that can be $\eta(\omega_e, T_e)$, $\phi(\omega_e, T_e) = \frac{1}{\eta}$, or $P_{in}(\omega_e, T_e)$) evaluated at the corresponding grid point.

In order to ensure numerical stability, it is always good practice to first normalize and center the system:

$$\mathbf{F}_n(\omega_e, T_e)\mathbf{c}_n = \eta_n$$

where

$$\begin{aligned} \mathbf{m}_F &= \text{mean of columns of } \mathbf{F} \text{ (as a row vector)} \\ \sigma_F &= \text{std. deviation of columns of } \mathbf{F} \text{ (as a row vector)} \\ \mathbf{F}_{n,(i,j)} &= \frac{\mathbf{F}_{i,j} - \mathbf{m}_{F,i}}{\sigma_{F,i}} \end{aligned}$$

and

$$\begin{aligned} m_\eta &= \text{mean}(\eta) \\ \sigma_\eta &= \text{std. deviation}(\eta) \\ \eta_{n,i} &= \frac{\eta_i - m_\eta}{\sigma_\eta} \end{aligned}$$

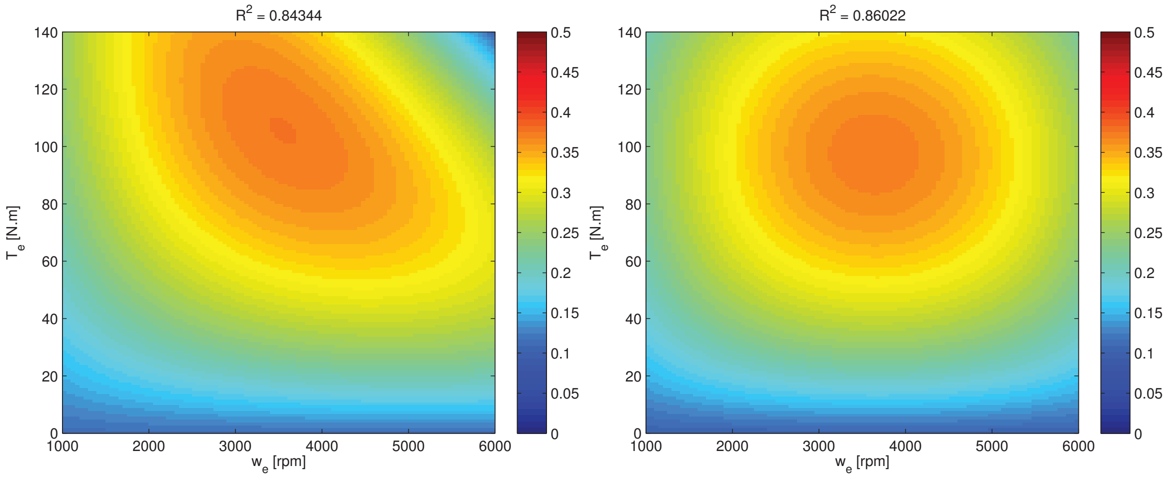
If the original coefficients are needed, they are given by

$$\mathbf{c} = (\mathbf{F}^T\mathbf{F})^{-1}\mathbf{F}^T(\mathbf{F}_n\mathbf{c}_n\sigma_\eta + \mathbf{m}_\eta)$$

where $\mathbf{m}_\eta$ is a column vector where every element has the value m_η

The classes of polynomials presented below are chosen based on the highest degree of an individual variable, not the polynomial's degree itself. This approach was chosen because the results naturally grouped together in such a way.

3.4.1.1 Polynomials of highest degree 2



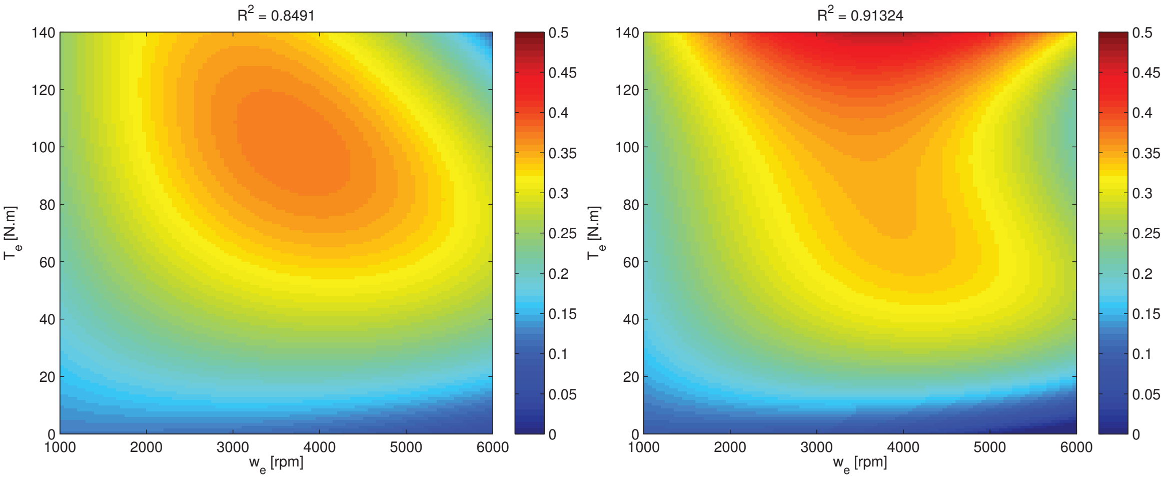
(a) $\eta = \alpha_0 + \alpha_1\omega_e T_e + \alpha_2\omega_e^2 T_e + \alpha_3\omega_e T_e^2$

(b) $\eta = \alpha_0 + \alpha_1\omega_e T_e + \alpha_2\omega_e^2 T_e + \alpha_3\omega_e T_e^2 + \alpha_4\omega_e^2 T_e^2$

Figure 3.9: 2nd order η fittings

This is the class of polynomials where the highest degree in a single variable is 2. It would seem that the simplest model is already sufficient, although the $\mathbf{R}^2$ value could be seen as a bit low in the simplest cases.

3.4.1.2 Polynomials of highest degree 3



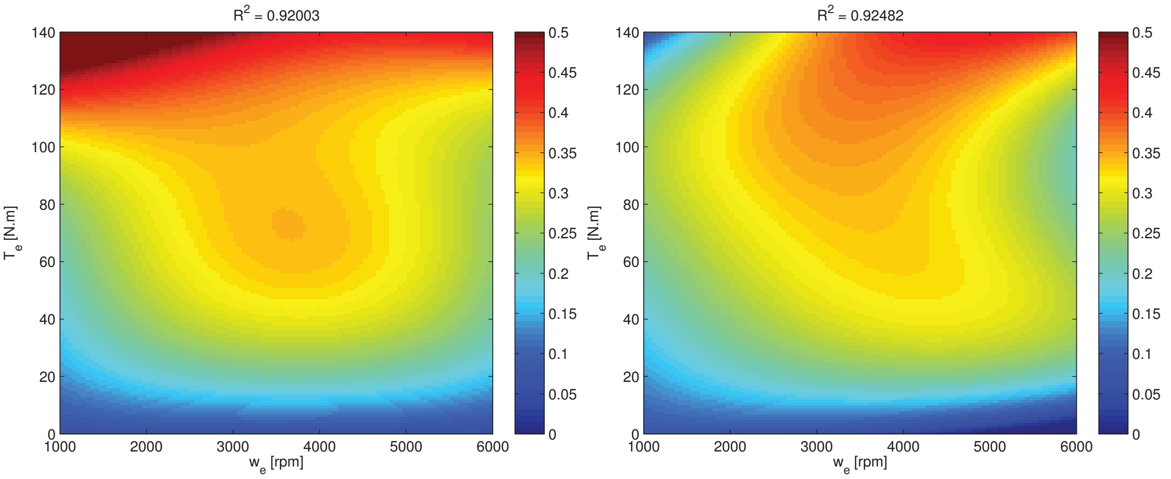
(a) $\eta = \alpha_0 + \alpha_1\omega_e T_e + \alpha_2\omega_e^2 T_e + \alpha_3\omega_e T_e^2 + \alpha_4\omega_e^3$

(b) $\eta = \alpha_0+\alpha_1\omega_eT_e+\alpha_2\omega_e^2T_e+\alpha_3\omega_eT_e^2+\alpha_4\omega_e^3+\alpha_5\omega_eT_e^3$

Figure 3.10: 3rd order η fittings

Figure 3.10a shows a slightly or somewhat better residual than the 2nd order, but not significantly so. Figure 3.10b has a higher fitting coefficient, but the efficiency island turns into a ridge. While this might indeed be the case, it is difficult to tell from the real data, so with all other things being equal we prefer the simpler model.

3.4.1.3 Polynomials of highest degree 4



(a) $\eta = \alpha_0 + \alpha_1\omega_e T_e + \alpha_2\omega_e^2 T_e + \alpha_3\omega_e T_e^2 + \alpha_4\omega_e^2 T_e^2 + \alpha_5 T_e^4$

(b) $\eta = \alpha_0 + \alpha_1\omega_e T_e + \alpha_2\omega_e^2 T_e + \alpha_3\omega_e T_e^2 + \alpha_4\omega_e^3 + \alpha_5 T_e^3 + \alpha_6\omega_e T_e^3 + \alpha_7 T_e^4$

Figure 3.11: 4th order η fittings

As seen in Fig. 3.11 there is a high degree of correlation between the polynomial model and the data. It would seem that Fig. 3.11a is good candidate for an efficiency map, as with only 6 parameters it seems to capture the essential efficiency island.

3.4.1.4 Conclusion

In the end, it would seem that simple polynomials with highest degree 2 are already sufficient versus higher order fittings. In the interests of simplicity, we will choose this fitting model. It has the added advantage of being very robust in having a well defined efficiency peak. However, if increased resolution is necessary, Fig. 3.11 shows promise.

Another possible fitting approach would be to estimate the polynomial coefficients directly using the EKF. However, this is not as reliable in this case, as one could miss special features in the efficiency graph. It appears preferable to build a non-parametric efficiency map, and then curve-fit it. Another reason for this approach is explained at the end of [13].

It must be added that these curves cannot be completely used as published, as they do not show the engine's limitations. While these can only be mapped by a dynamometer test, for all intents and purposes they can be inferred by performing a series of WOT (wide open throttle) tests. Figure 3.7, for instance, is bounded quite nicely by the results from several WOT tests that have been averaged into the data. Likewise for Fig. 3.20. Figure 3.12 shows how the efficiency curve could look like with limitations on ω_e and T_e.

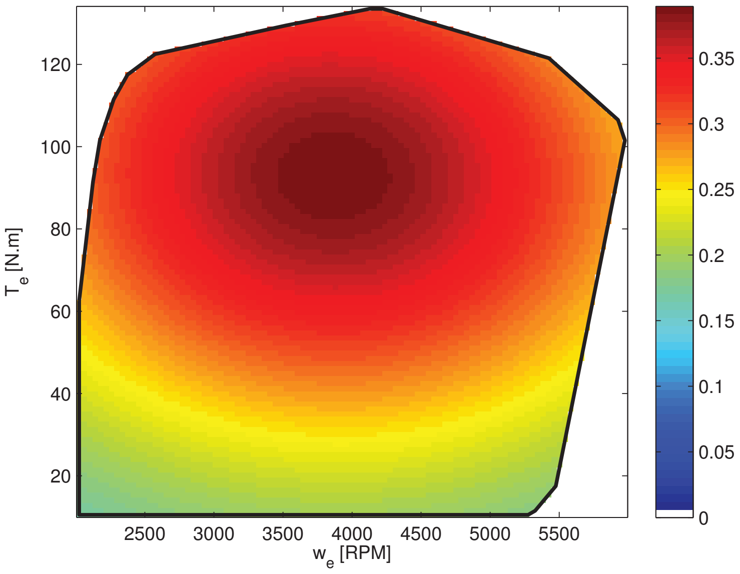
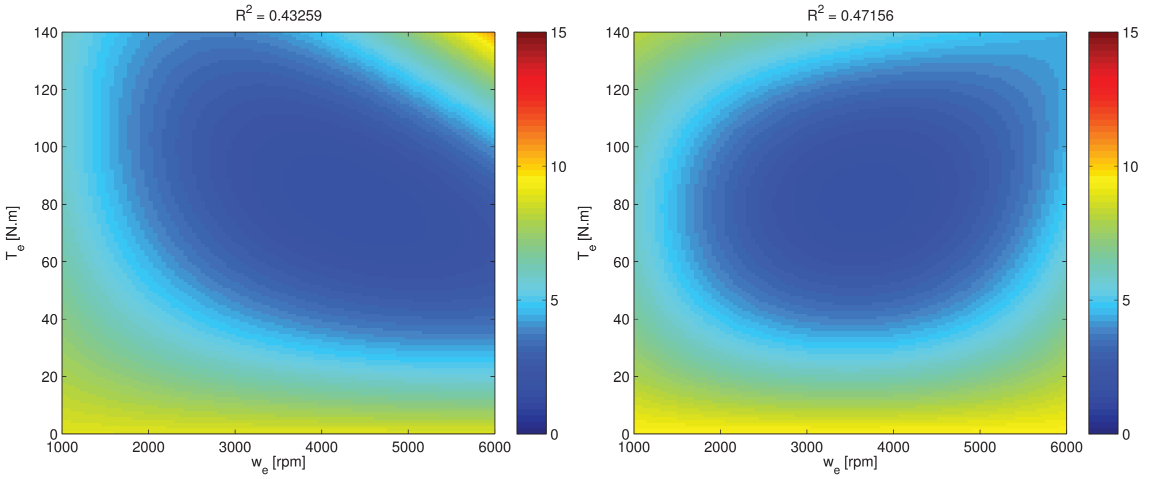


Figure 3.12: Example efficiency map with bounds

3.4.2 Specific fuel consumption fitting

Here we seek to identify the inverse, $\phi(\omega_e, T_e) = 1/\eta$. This is a useful fitting because, as explained in Sec. 1.3.2, the controller model uses the inverse of η so it is natural to try to fit η^{-1}

3.4.2.1 Polynomials of highest degree 2

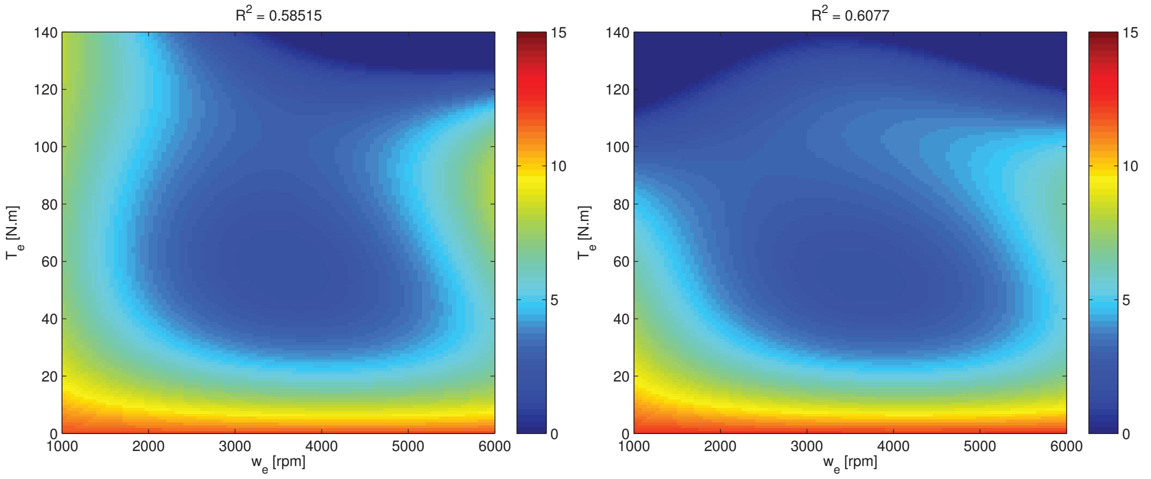


(a) $\eta^{-1} = \alpha_0 + \alpha_1\omega_e T_e + \alpha_2\omega_e^2 T_e + \alpha_3\omega_e T_e^2$ (b) $\eta^{-1} = \alpha_0 + \alpha_1\omega_e T_e + \alpha_2\omega_e^2 T_e + \alpha_3\omega_e T_e^2 + \alpha_4\omega_e^2 T_e^2$

Figure 3.13: 2nd order η^{-1} fittings

The residual for these second order polynomials is quite poor. This is certainly because we are trying to fit the inverse of a function that was quite well described by a polynomial with positive coefficients. It follows, then that this function will resist fitting by a low number of coefficients.

3.4.2.2 Polynomials of highest degree 3



(a) $\eta^{-1} = \alpha_0 + \alpha_1\omega_e T_e + \alpha_2\omega_e^2 T_e + \alpha_3\omega_e T_e^2 + \alpha_4\omega_e^2 T_e^2 + \alpha_5\omega_e T_e^3$ (b) $\eta^{-1} = \alpha_0 + \alpha_1\omega_e T_e + \alpha_2\omega_e^2 T_e + \alpha_3\omega_e T_e^2 + \alpha_4\omega_e^2 T_e^2 + \alpha_5 T_e^3 + \alpha_6\omega_e T_e^3$

Figure 3.14: 3rd order η^{-1} fittings

Figure 3.14 shows similar behavior to the above. The R^2 value is quite poor.

3.4.2.3 Polynomials of highest degree 4

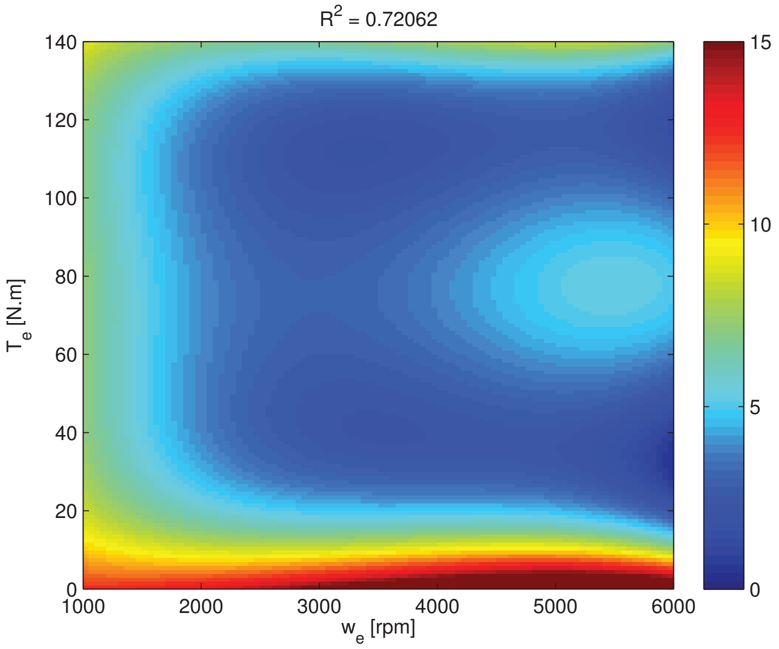


Figure 3.15: 4th order fitted curve, $\eta^{-1} = \alpha_0 + \alpha_1 \omega_e T_e + \alpha_2 \omega_e^2 T_e + \alpha_3 \omega_e T_e^2 + \alpha_4 \omega_e^2 T_e^2 + \alpha_5 \omega_e^3 + \alpha_6 w_e T_e^3 + \alpha_7 w_e T_e^4 + \alpha_8 T_e^4$

Even with 4^{th} order variables, the polynomial shown in Fig. 3.15 still has a great deal of difficulty capturing the essential dynamics.

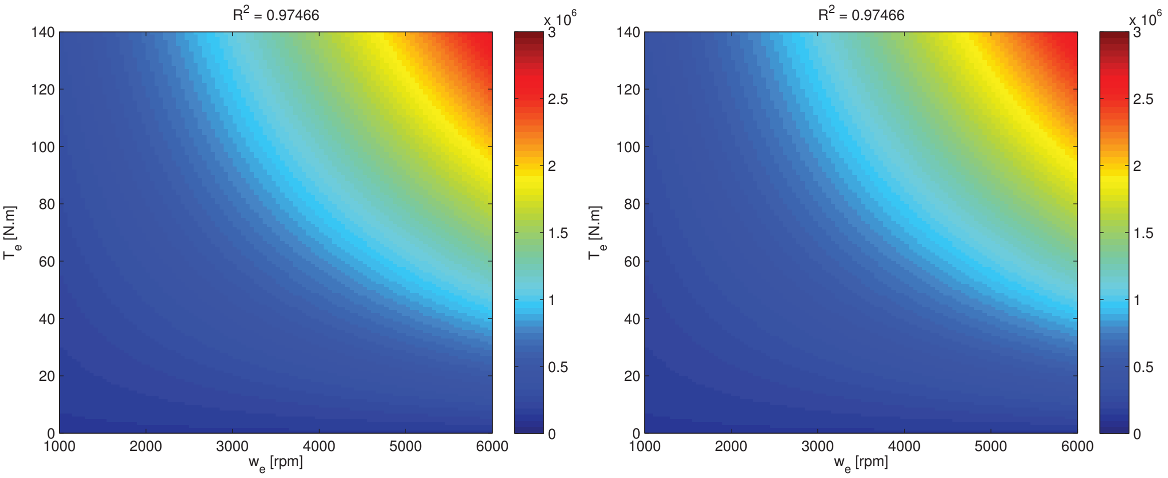
3.4.2.4 Conclusion

In fact, the poor fit is not as drastic as it might seem. By examining $1/\phi$ we can see how well the curve fitting corresponds to the original data shape. If the overall shape of the function is correct, then we can use ϕ in the optimal calculations and expect to see similar results.

3.4.3 P_{in} fitting

Here we seek to identify the entire power input function, $P_{in}(\omega_e, T_e)$. This could be a very useful fitting since this describes the entire efficiency expression as it appears in the Hamiltonian (Cf. Ch. 4).

3.4.3.1 Polynomials of highest degree 2



(a) $P_{in} = \alpha_0 + \alpha_1\omega_e T_e + \alpha_2\omega_e^2 T_e + \alpha_3\omega_e T_e^2$ (b) $P_{in} = \alpha_0 + \alpha_1\omega_e T_e + \alpha_2\omega_e^2 T_e + \alpha_3\omega_e T_e^2 + \alpha_4\omega_e^2 T_e^2$

Figure 3.16: 2rd order P_{in} fittings

The residual for these second order polynomials is quite good. We see that there is little to be gained by adding additional terms, and so a simple polynomial seems to fit well.

3.4.3.2 Polynomials of highest degree 3

Figure 3.17 does not show a significant improvement from the previous examples.

3.4.3.3 Polynomials of highest degree 4

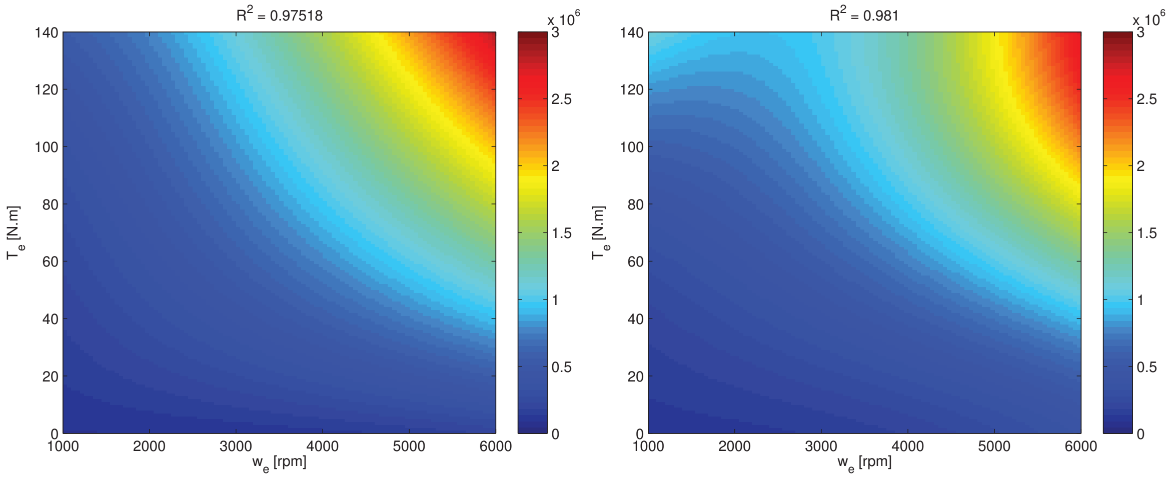
These higher order polynomials seem to show interesting behavior, but it is difficult to tell which is the "true" model. In these cases, it is important to avoid over-fitting the data.

3.4.3.4 Conclusions

Higher order models do not seem to add anything intrinsic to the general form of the fitting, and so we conclude that lower order models are sufficient.

3.5 Validation

The Smart was tested on a dynamometer in order to validate the proposed method. The test was performed by putting the dynamometer into constant speed operation, and varying



(a) $P_{in} = \alpha_0 + \alpha_1\omega_e T_e + \alpha_2\omega_e^2 T_e + \alpha_3\omega_e T_e^2 + \alpha_4\omega_e^2 T_e^2 + \alpha_5 T_e^3 + \alpha_6\omega_e^3$

(b) $P_{in} = \alpha_0 + \alpha_1\omega_e T_e + \alpha_2\omega_e^2 T_e + \alpha_3\omega_e T_e^2 + \alpha_4\omega_e^2 T_e^2 + \alpha_5 T_e^3 + \alpha_6\omega_e^3 + \alpha_7\omega_e T_e^3$

Figure 3.17: 3rd order P_{in} fittings

the torque produced by the Smart's engine.

The constant speed steps were taken in units of 100rpm, over a range from 2500rpm to 6000rpm. The torque was varied between 25Nm and 80Nm.

As the reader can see in Fig. 3.19, the curve fitted results correspond nicely to the observed results in Fig. 3.9b. While there is a scaling effect versus the experimental data results, this is easily corrected by calculating some of the parameters rather than using the reported manufacturer values, e.g. C_{rr}, C_d, fuel injection calibration, etc... It is also possible that the dynamometer calibration be somewhat off, as a dynamometer equipment is usually used for showing a relative change (useful when hot-rodding a car) instead of absolute accuracy.

What is more important is that the maximum efficiency occurs at roughly the same point in the two graphs, and the overall form is identical.

3.6 Additional results

As evidence that the above experiment is in no way specific nor tuned to the Smart Roadster, we present the efficiency graphs for a 2005 Peugeot 206RC. The engine is of a completely different construction, has a different number of cylinders, different displacement, and most of all is naturally aspirated, whereas the Smart Roadster is turbocharged.

It is interesting to note that in these graphs the same basic patterns emerge again, as expected. There is a maximum efficiency island at medium torque and rpm settings, although in the Peugeot's naturally aspirated engine it seems to occur at a slightly lower rpm than in the Smart's turbocharged engine.

Note that results are repeatable enough that once the efficiency map is established, the engine model can be used to estimate other car parameters. For instance, it would be possible to estimate rolling resistance and thus give a sort of low-pressure tire warning. It is also possible to estimate mass, or to see the change that aerodynamic modifications make in the

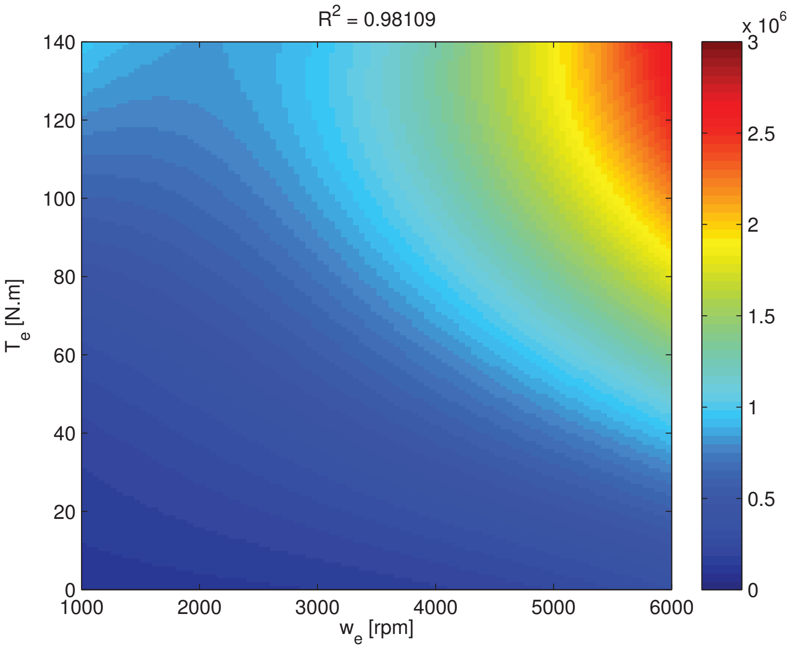


Figure 3.18: 4th order fitted curve, $P_{in} = \alpha_0 + \alpha_1\omega_e T_e + \alpha_2\omega_e^2 T_e + \alpha_3\omega_e T_e^2 + \alpha_4\omega_e^2 T_e^2 + \alpha_5 T_e^3 + \alpha_6\omega_e^3 + \alpha_7\omega_e T_e^3 + \alpha_8 T_e^4$

drag coefficient. It could furthermore be used to track slow deteriorations in the motor's general health, serving as an early warning system for engine maintenance.

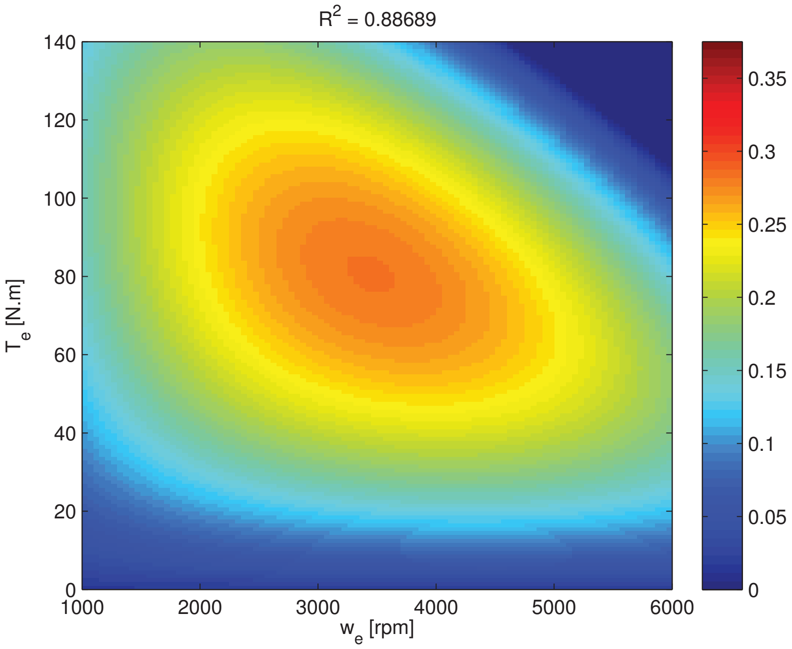


Figure 3.19: Efficiency map as calculated from dynamometer data. $\eta = \alpha_0 + \alpha_1\omega_e T_e + \alpha_2\omega_e^2 T_e + \alpha_3\omega_e T_e^2 + \alpha_4\omega_e^3 + \alpha_5\omega_e T_e^3$

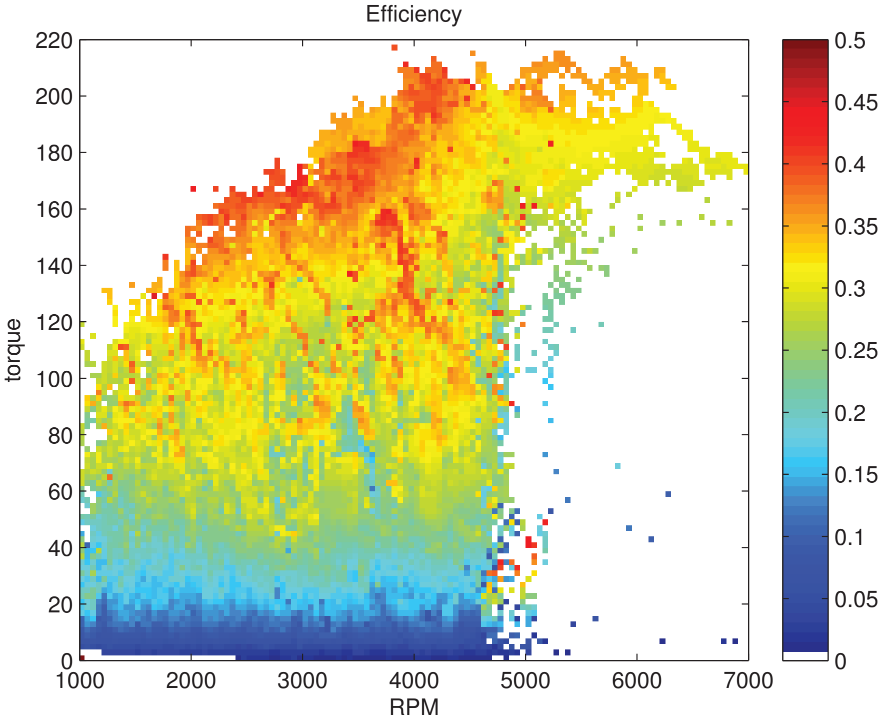


Figure 3.20: Efficiency map, Peugeot 206RC

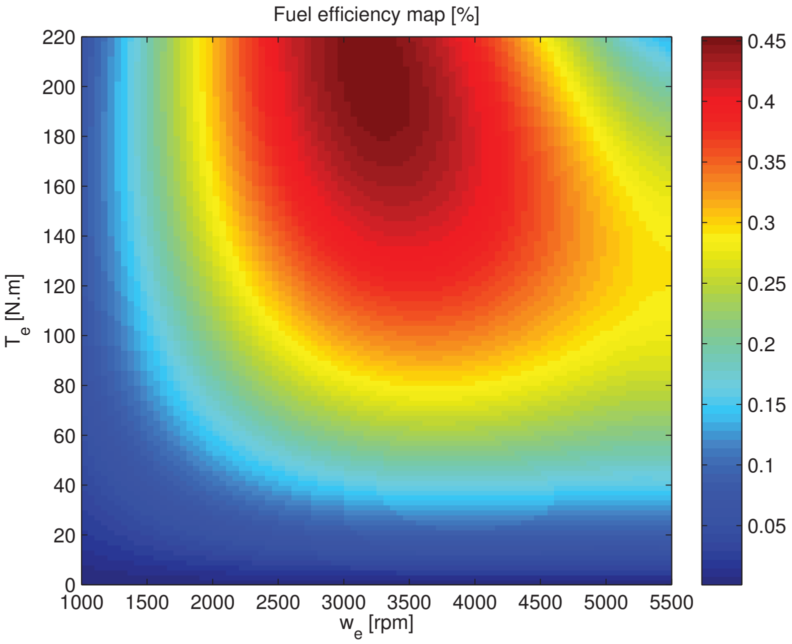


Figure 3.21: Efficiency map as 4^{th} order fitted curve, Peugeot 206RC.

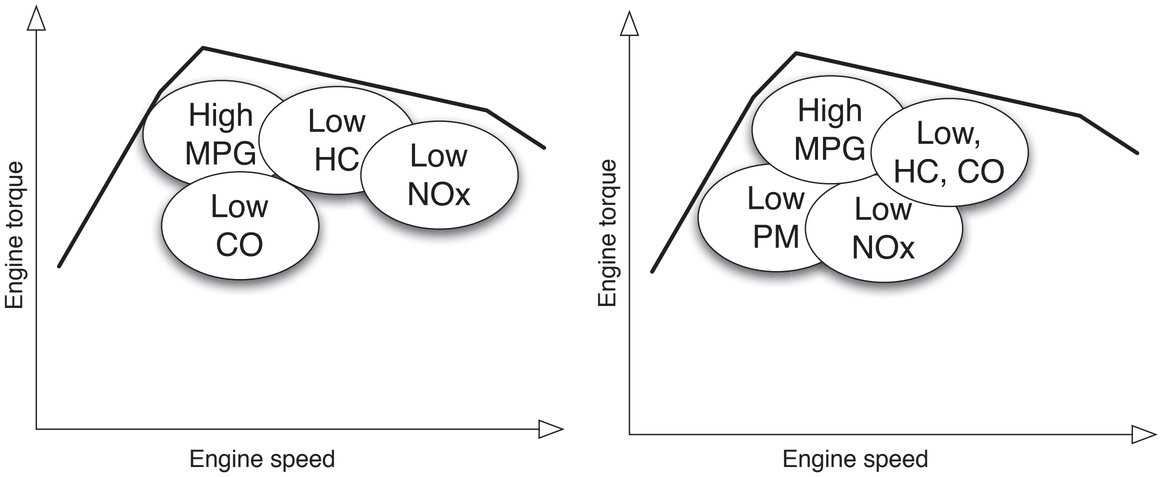
Chapter 4

Applying Pontryagin's Maximum Principle

Contents

4.1 Defining optimality

On an internal combustion engine's torque-speed map, the zone of maximum efficiency does not necessarily coincide with the zones of least emissions.[27] In certain cases, these zones can be far apart, and any optimal study destined for mass utilization must choose the optimal cost with care.



(a) Gasoline, manifold injection

(b) Diesel, direct injection

Figure 4.1: Engine operation zones

In Fig. 4.1, we distinguish between the Diesel cycle and the Otto cycle engines. HC indicates *hydrocarbon* emissions, NOx represents *nitrous-oxide* emissions, PM represents *particulate matter*, and CO *carbon monoxide*.

MPG, for the non-native reader, refers to miles per gallon.[1]

In particular, there is a definite tradeoff between NOx emissions and fuel consumption. This is the consequence of the intersection of chemistry with thermodynamics. One of the most basic elements of thermodynamics explains that the higher the compression, the higher the Carnot efficiency. On the other hand, basic chemical analysis shows that the higher the compression, the higher the quantity of NOx formed as a result.

The challenge for a holistic control strategy is thus to simultaneously reduce emissions while increasing fuel economy.

[1] While the measurement is expressed in archaic units, it is a superior way of expressing fuel economy to its pseudo-inverse "L per 100km". Due to the human propensity to understand, and favor, large numbers over small, it is sociologically better to link a greater good with a greater quantity. The difference between 2.9L/100km and 3.0L/100km appears miniscule to a potential buyer, but it represents years of research and refinement. Whereas the difference between 78mpg and 81mpg is clear to even the most mathematically challenged among us. The author hopes to one day see car efficiency given in kpl.

4.2 Fixed time versus non-fixed time optimization

Optimal control approaches, such as PMP, can treat the optimal dynamic problem in fixed-time or non-fixed time. In the former case, the problem consists of minimizing the cost for a trajectory between 0 and T, T being fixed. In the second case, T is a parameter to be optimized (and thus the minimum cost will be less than or equal to the fixed-time case). The PMP conditions are different in the two cases, which justifies treating them separately.

From a practical point of view it also makes sense to differentiate between the two problems. If the goal is to get to a certain point (e.g. drive a certain distance, accelerate to a certain speed), the problem is clear when T is fixed. As long as a reasonable value for T is given, it is likely that the solution is also reasonable, even without taking into account the state constraints (maximum engine speed, road speed limit, etc.) If T is non-fixed, it is more difficult to imagine the solution ahead of time. For instance, minimizing the cost can lead to T being small (the cost being the integral of a positive value between 0 and T), but inversely the smaller T is the more time spent at the upper limits of the motor's performance envelope, with the associated efficiency penalty, and at higher speeds, with the associated aerodynamic losses. (After all, the most fuel efficient way to get anywhere is not to use fuel: pushing the car, for example.) Clearly, it comes down to a sort of mathematic compromise.

In the case where T is non-fixed, the problem's meaning depends to a great deal on the goal (position or speed), as will be seen in the case studies, Sec. 4.4, 4.5.

4.3 Global vs. local optimization

It makes sense to think about practical optimization as a series of slices which can be joined together to produce a "more optimal" trajectory. While this trajectory is no longer properly called the optimal trajectory for an entire car trip, the practicality of actually applying the trajectory make the theoretical considerations somewhat less imperative. The practical aspects lead us to an approach where we try to establish a set of rules that drivers can follow naturally for different phases of the trip. Once the system has identified η as a function of T, ω_e and optimized the trajectory, the objective is to transmit the information in some way that the driver can understand. ("Accelerate briskly rather than gently— but don't floor it either.")

A natural separation that springs to mind is to distinguish between acceleration phases, such as merging into traffic, and cruise phases, such as going long distances on highways and autoroutes.

Fuel economy pptimization in cities is a more arduous task, and has less impact in absolute terms, as it first requires a city traffic model, which is outside the scope of this PhD.[2] Moreover, fuel optimization has less to say about in-city driving, as the number of kilometers driven in-city are far fewer than those driven in more rural environments. Note, however, that optimization that takes into account emissions, such as NOx, SOx, particles, etc..., is still very interesting in urban environments, and with the advent of hybrid cars these optimizations become much more practical.

[2] However, the sensor suite and observer developed earlier in the dissertation are already being used by other researchers in order to gather data for modeling traffic flow.

Case studies

We will now study several cases with multiple goals and then calculate the corresponding optimal trajectory. This allows us to validate our approach, knowing that the real-world application requires a more exhaustive study.

The problem is that optimal controls are very restrictive: the slightest slip, the tinniest error and optimality is lost. Our goal, then, is to optimize driving styles in a human way, in a way that does not require that the driver have unearthly powers of prediction and calculation. With the right rules, the driver will be able to drive more economically while only having to look a short distance into the future.

Note 13 *The "pulse and glide" technique is not treated here. This approach to high-efficiency driving involves a period of acceleration (the pulse) followed by a period in which the engine is shut off (the glide). While this approach is possibly more fuel efficient, it is too dangerous to be considered here, as in modern cars shutting off the engine removes the source of vacuum for the brake boost, leading to a situation where the brakes are no longer effective after one or two uses.*

Fitting polynomial: For the case study, we have chosen to use the fitting $\phi = 1/\eta$. This choice is motivated by the fact that it is an easily differentiable polynomial, which eases the analytic task. While the results presented in Sec. 3.4.2 show that this is not the best fit for the ensemble of points, it is nonetheless a good fit for the area of peak efficiency, and has the necessary behavior, i.e. a pronounced peak efficiency followed by clear dropoffs to the sides.

The fitting polynomial and coefficients are presented in Tab. 4.1.

Coefficient	Value
α_0	9.4360
α_1	-0.018136
α_2	-0.068797
α_3	3.7201e-06
α_4	2.1965e-05
α_5	3.6204e-04

Table 4.1: Coefficients for $\phi = \alpha_0 + \alpha_1\omega_e + \alpha_2 T_e + \alpha_3\omega_e T_e + \alpha_4\omega_e^2 + \alpha_5 T_e^2$

4.4 One dimension: acceleration on the straight and level

We suppose that the ground is perfectly flat, and that the car is only allowed to have a forward velocity. As such

$$\left\{ \begin{array}{rcl} \dot{x}_i & = & v_i \\ \dot{v}_i & = & \dfrac{1}{M_a}\left(T_e N_{spd} - \left(\dfrac{1}{2}\rho C_d A v_i^2 + M C_{rr} g\right)\right) \end{array} \right. \tag{4.1}$$

where ω_e, T_e, x_i, v_i are all ≥ 0[3]

The target is $v(T) = v^*$, where $t_f = T$ is supposed non-fixed. We make no assumption about the distance traveled. The problem is thus to accelerate to the given speed while using the least amount of energy necessary. The problem of T fixed, interesting from a practical point of view as explained above, will be considered later.

We now apply use the PMP theorem (see Annex B.2.5). Combining the cost eq. (1.30) with eq. (B.13), we can write the Hamiltonian, H, as

$$H = \lambda \dot{\mathbf{x}} + \lambda^0 J = \lambda_x v_i + \lambda_v \frac{1}{M_a}\left(T_e N_{spd} - \left(\frac{1}{2}\rho C_d A v_i^2 + M C_{rr} g\right)\right) + \lambda^0 \omega_e T_e \phi(\omega_e, T_e) \quad (4.2)$$

Since x_i does not appear in the target, as here the problem is only in one dimension, the costate transversality conditions impose $\lambda_x(T) = 0$, and since $\dot{\lambda}_x = -\frac{\partial H}{\partial x_i} = 0$, $\lambda_x = 0$ for all t.

In the maximal version of the PMP, $\lambda^0 \leq 0$. However, when $\lambda^0 \neq 0$ we can normalize and take $\lambda^0 = -1$, as the costate vector's magnitude is unimportant. This leaves the case of $\lambda^0 = 0$. If $\lambda^0 = 0$ then in order for the Hamiltonian H to be constant $\dot{v}_i$ must be 0, since the vector $< \lambda^0, \lambda > \neq \vec{0}$, but a constant v_i clearly cannot be an optimal trajectory. Thus we can conclude that $\lambda^0 \neq 0$ and thus we take $\lambda^0 = -1$. This yields the reduced equation

$$H = \lambda_v \frac{1}{M_a}\left(T_e N_{spd} - \left(\frac{1}{2}\rho C_d A v_i^2 + M C_{rr} g\right)\right) - \omega_e T_e \phi(\omega_e, T_e) \quad (4.3)$$

From the Hamiltonian transversality conditions (Sec. B.2.5.1), we know that $H(T) = 0$ and for an optimal control $H(t)$ =constant.

Furthermore, from the fact that along the trajectory, $\dot{v}_i \geq 0$, and $\omega_e T_e \phi(\omega_e, T_e) \geq 0$, then for $H(T) = 0$ to hold, the costate $\lambda_v \geq 0$. We can deduce that $\lambda_v = 0$ is an *absurd* solution, as it requires that $\omega_e T_e \phi(\omega_e, T_e) = 0$, so therefore $\lambda_v > 0$.

4.4.1 CVT case

A CVT (Continuously Variable Transmission) is a special type of transmission that is seeing increasing use, especially on hybrid and high efficiency vehicles. While the exact nature of the transmission depends on the particular manufacturer, the objective is to be able to have an arbitrary output speed for any given input speed. This allows the engine to always run as close as possible to peak efficiency for a given power demand.

The CVT model is particularly interesting as it appears that this is a major trend in modern cars. In the CVT case, we suppose that ω_e and T_e are the controls, $\mathbf{u}$.

Examining the Hamiltonian more closely, and recalling that $N_{spd} = \frac{\omega_e}{v_i}$ is non-constant in a CVT equipped car, we can see that $\dot{\lambda}_v > 0$ since

$$\dot{\lambda}_v = -\frac{\partial H}{\partial v_i} = \lambda_v \frac{1}{M_a}\left(\frac{\omega_e T_e}{v_i^2} + (\rho C_d A v_i)\right) > 0$$

[3] In the case of ω_e, x_i, v_i, this is immediately obvious, as they can only increase. T_e, on the other hand, is more subtle. If T_e were negative, this would imply a braking phase. In a pure acceleration phase, clearly braking would be less efficient, hence $T_e \geq 0$

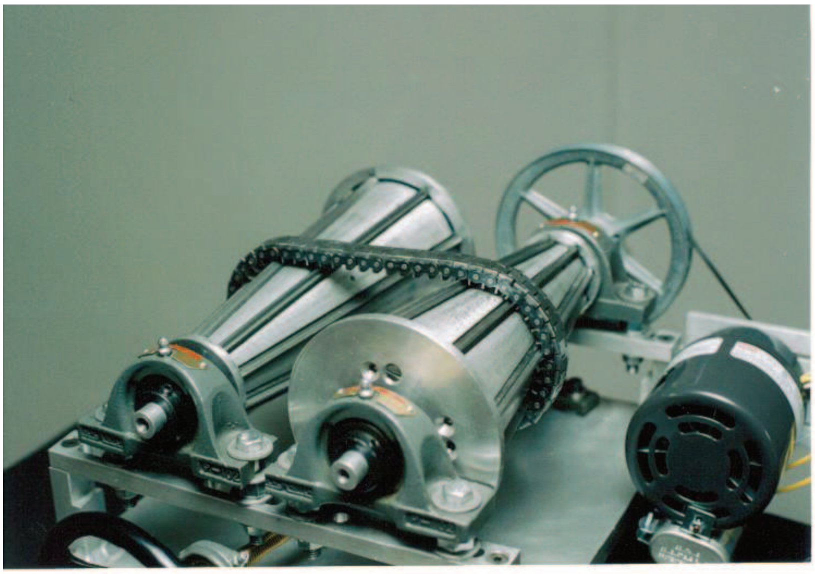


Figure 4.2: Prototype Lexus RX 400h CVT

Maximizing $H(t)$ with respect to the controls then reduces to maximizing only the part of the Hamiltonian that depends on ω_e and T_e

$$\omega_e T_e \left(\frac{1}{\eta(\omega_e, T_e)} + \frac{\lambda_v}{M_a v_i} \right)$$

which is a 3rd degree polynomial in (ω_e, T_e) and only depends on v_i and λ_v The problem is reduced, thus, to a shooting problem. (Remember that $\phi = 1/\eta$, so the numerical shooting problem can depend on ϕ or $1/\eta$, depending on the approach.)

As discussed in Sec. B.3, we can resolve the shooting problem with numerical methods.

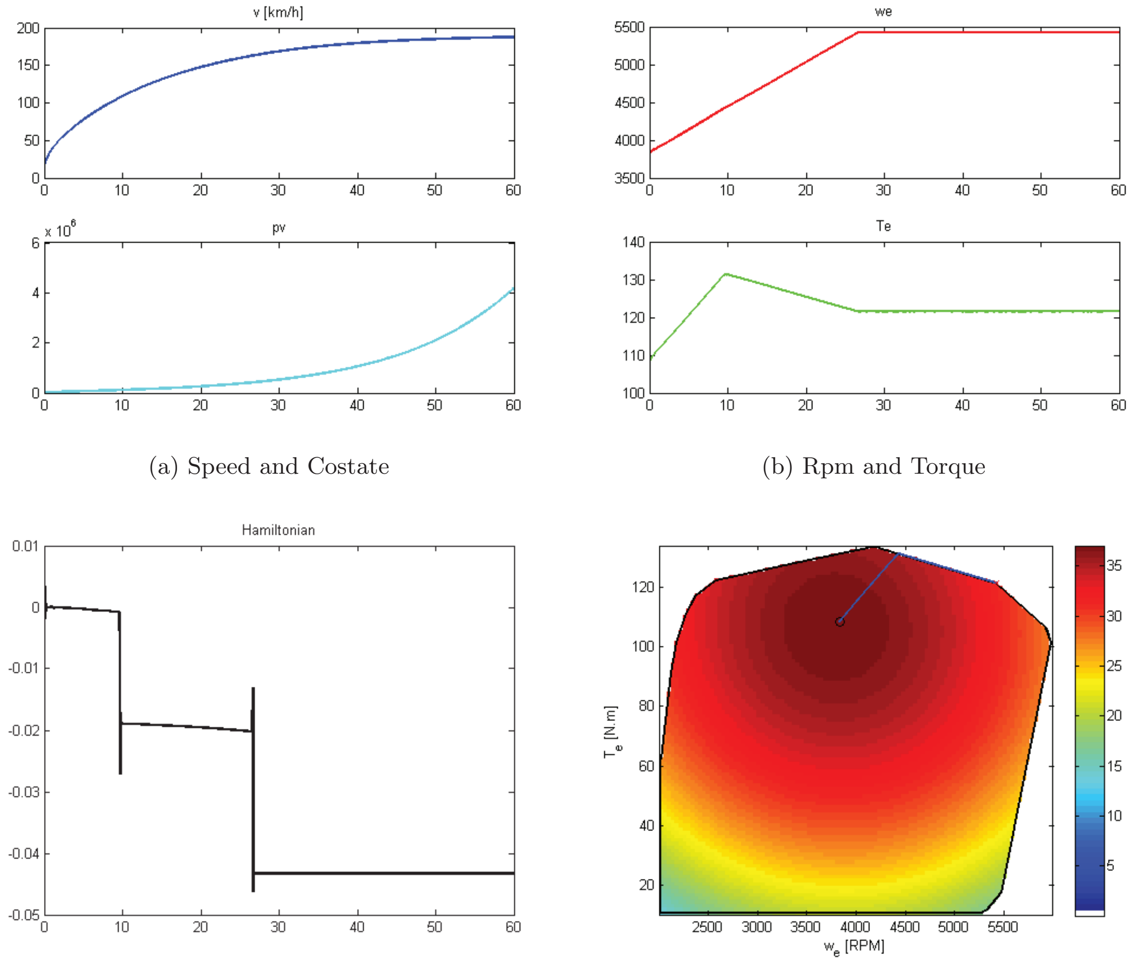
4.4.1.1 T, non-fixed horizon

As can be seen in Fig. 4.3, the results are somewhat realistic— putting aside for the moment that the Smart Roadster has a 6-spd gearbox, and not a CVT. It takes around 10 seconds to reach 100km/hr, which is consistent with the car's real-world performance.[4] The car attains the maximum velocity of 180km/hr after about 1 min. Again, consistent with real-world performance. Nonetheless, due to the CVT assumption this demonstration is not very realistic.

Note that $\lambda_v(0)$ was determined by resolving:

$$\max_{(\omega_e, T_e) \in \mathfrak{U}_{\text{adm}}} H(\mathbf{x}, \mathbf{u}, \lambda) = 0$$

[4] Manufacturer's data indicates 0-100 in 10.9s, but this is not a straight-up apples-to-apples comparison. In the maximum acceleration test, the real Smart loses a significant amount of time in gear shifts ($\approx$ 3 s), but makes up by accelerating harder between shifts.



(a) Speed and Costate

(b) Rpm and Torque

(c) Hamiltonian

(d) Control trajectory

Figure 4.3: Optimization on the straight and level: CVT, T_f non-fixed

for a given initial speed $v > 0$. (Remember that here the Hamiltonian by definition $= 0$, but also that along an optimal trajectory the control $\mathbf{u}$ must maximize H. In other words, any control which is not optimal will not only not maximize the Hamiltonian, but it will allow it to be other than 0.)

Note that the Hamiltonian was not completely preserved, due to the numerical integration. There are two ways to solve this problem. The first is to use a numerical algorithm that preserves the Hamiltonian.[40] This type of integration method is effective but difficult to put into practice. Another method, based on the theory of observers, was developed by Busvelle et al. and gives highly satisfactory results for a low numerical cost.[14] This problem has not been considered as critical since the errors are within tolerance.

The trajectory in the control/efficiency space can be seen in Fig. 4.3d. Note that at $t \approx 10$s the control reaches its limits and is no longer singular. Conceptually, this results makes sense, since initially the car is at a low speed, where efficiency losses are greater than drag losses. As the car accelerates, however, it becomes more important to accelerate harder, leaving the area of maximum efficiency, in order to minimize the time spent with the higher drag losses.

4.4.2 Multi-speed transmission case

The multi-speed case is interesting as it represents the vast majority of cars that have a limited number of fixed gears. In this case, our case study uses a six-speed gear-box.

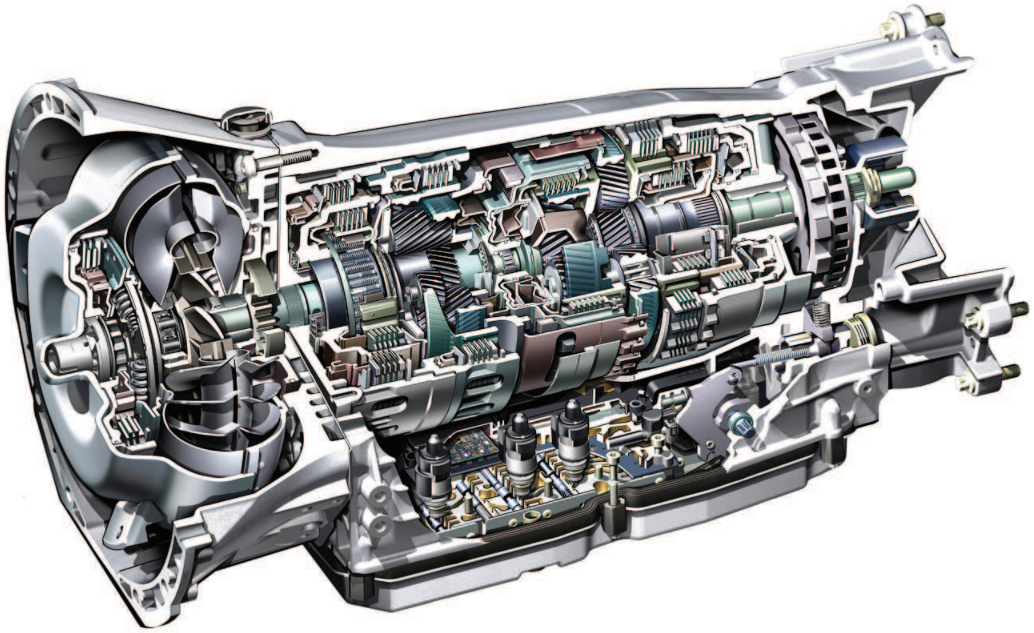


Figure 4.4: Classic multi-speed transmission

The Hamiltonian is written as before:

$$H = \lambda_v \frac{1}{M_a} \left(T_e N_{spd} - \left(\frac{1}{2} \rho C_d A v_i^2 + M C_{rr} g \right) \right) - \omega_e T_e \phi(\omega_e, T_e)$$

but this time the control is made of the pairing (N_{spd}, T_e) where N_{spd} is the ratio ω_e / v which in the case of a six-speed transmission can have six distinct values.[5] The two controls are not equivalent to a static feedback since $\omega_e = N_{spd} v$

[5] Not taking reverse into account, of course. Indeed, there is a theoretically interesting problem which is not examined in this PhD. Imagine a car on a perfectly sinusoidal road (in the vertical sense). The car starts off pointing "uphill" and the optimal target is to reach the top while minimizing fuel consumption. No one would doubt that the car's most efficient trajectory would never require rolling backwards in neutral to a "downhill" part before reengaging the transmission in order to benefit from some small forward velocity and the associated higher engine efficiency at moderate speeds.

However, this is only because there is a critical point at which the car's rolling resistance and aerodynamic drag losses become so large that to roll backwards is *de facto* less efficient. Obviously, if there were no drag

Therefore, the Hamiltonian as a function of N_{spd} is

$$H = \lambda_v \frac{1}{M_a}\left(T_e N_{spd} - \left(\frac{1}{2}\rho C_d A v_i^2 + M C_{rr} g\right)\right) - N_{spd} v T_e \phi(N_{spd} v, T_e)$$

and so using the chain rule

$$\dot{\lambda}_v = -\frac{\partial H}{\partial v_i} = \lambda_v \frac{1}{M_a}\rho C_d A v_i + N_{spd} T_e \phi(N_{spd} v, T_e) + N_{spd}^2 v T_e \frac{\partial}{\partial \omega_e}\left[\phi(\omega_e, T_e)\right]$$

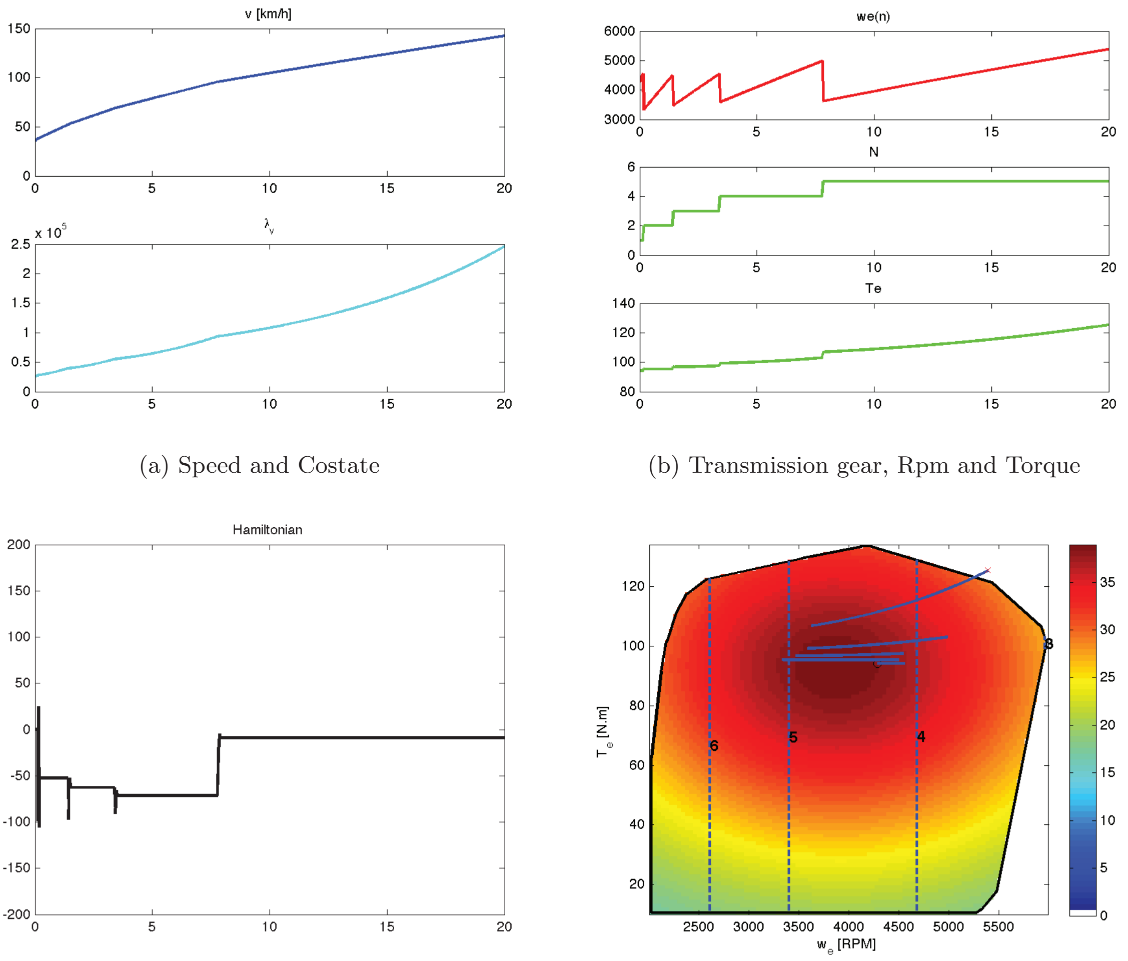
While passing from the domain of the physical controls in the (ω_e, T_e) space to the domain in (N_{spd}, T_e), the constraints become dependent on the state. The maximum principle thus must be one which takes into account the state constraints. This is a more complicated theorem because the costate vector can have discontinuities that must be calculated. In this case, however, it is not clear that this will be necessary. In effect, the intuition is that the controls will gravitate around the zones of maximum efficiency, at the expense of the zones where the efficiency drops off rapidly, which occurs especially at the state constraints. Thus, as discussed in Sec. B.2.5.3, the optimal solution can be found without considering the state constraints, and then compared with the true constraints. If the solution does not violate the physical bounds in (ω_e, T_e), then this is the optimal solution sought after. If this is not the case, however, then there is no other choice than to apply the Pontryagin Maximum Principle with state limits, such as described in [10] and in Pontryagin's book [37].

4.4.2.1 T, non-fixed horizon

Figures 4.5 show the results from PMP with the same target as the previous CVT section. The car goes from 36km/hr to 100km/hr in 9.3s, which represents a medium-to-high acceleration of around 0.2g. The gears are changed at around 4500rpm, which is consistent with this kind of spirited acceleration. Note, however, that these trajectories correspond to a problem that is somewhat non-intuitive: reach a given speed while using the least amount of fuel possible, with time, T, on a non-fixed horizon. Since the cost is "paid" the instant the car reaches the final speed, this implies a relatively short total acceleration time.

In Fig. 4.5d, the control is superimposed on the control/efficiency graph. The control trajectories as a function of the gears can be clearly seen. At 120km/hr, the optimal solution is close to leaving the state constraint space and thus becomes no longer optimal. It is relatively easy to guess what the optimal trajectory will be after this point (by the Bellman dynamic programming principle as elaborated in Sec. B.3.3, but the PMP as it is applied cannot be used to calculate an optimal trajectory greater than 120km/hr. However, in most cases throughout Europe and the world the speed is limited to 120km/hr and so this is inconsequential.

losses whatsoever the optimal trajectory is to coast back and forth, only applying the control torque when the energy can be "bought" at the cheapest price, that is to say the point where v_i gives the highest efficiency. Defining this critical point would be an interesting study, and might have real-world applications for systems that have very low drag losses and very steep efficiency curves. Trains come to mind, such as perhaps an electric train in a coal mine.



(a) Speed and Costate

(b) Transmission gear, Rpm and Torque

(c) Hamiltonian

(d) Control trajectory vs. efficiency

Figure 4.5: Optimization on the straight and level: 6-speed transmission, T_f non-fixed

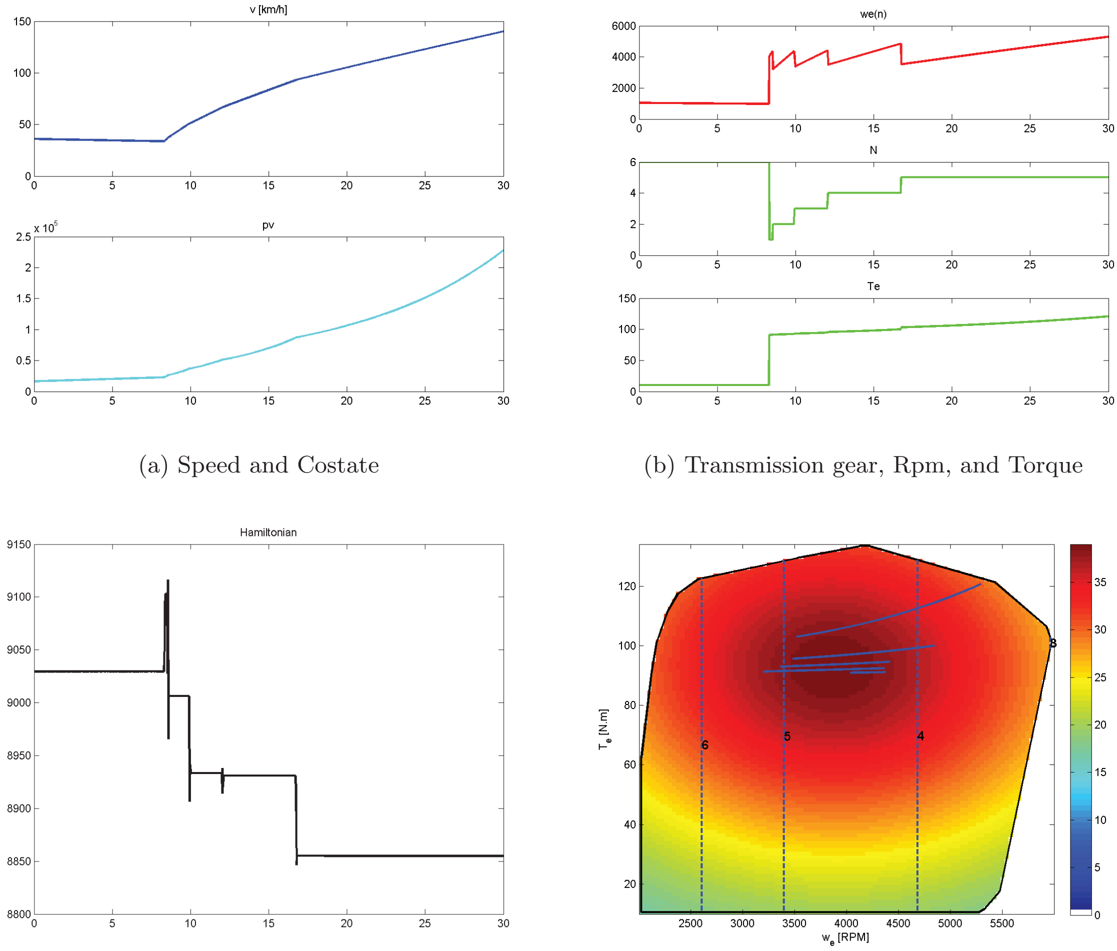
4.4.2.2 T, fixed horizon

This time, the goal is to reach the target speed v^* in a time T_f, while minimizing fuel consumption. To be specific, the goals are 140km/hr in 30s, which corresponds to a relatively normal acceleration while merging onto an autoroute.

The Hamiltonian is identical to the non-fixed horizon case, but this time the transversality criteria no longer give that $H = 0$ for all time t, although H constant along any optimal trajectory still (and always) holds. This is then a classical shooting problem, where $\lambda_v(0)$ must be determined such that $v(T_f) = v^*$. Since this is a shooting problem in only one dimension, it is straightforward to solve, for instance by the secant or bisection methods.

Results are shown in Fig. 4.6. At first glance, they seem surprising.

In the beginning, the control clearly stays at maximum N_{spd}, minimum T_e, which could be seen as almost the same thing as putting the car into neutral. Indeed, if the model had had



(a) Speed and Costate

(b) Transmission gear, Rpm, and Torque

(c) Hamiltonian

(d) Control trajectory vs. efficiency

Figure 4.6: Optimization on the straight and level: 6-speed transmission, T_f fixed

a neutral speed, it almost certainly would have coasted along until the acceleration phase at which point it behaves remarkably like the non-fixed time example. This is consistent with our expectations, since the most fuel efficient way to accelerate to a certain speed, disregarding the distance taken, is going to be to wait until the right moment and then accelerate.

The analogy taken to the extreme would be the case of a driver who must drive at 140km/hr at noon tomorrow. Obviously, the most fuel efficient strategy will be to wait until the last possible moment, using not a drop of gas, and then accelerate with the optimal profile in non-fixed horizon.

4.4.3 Real-world tests

As with the earlier sections, the real-word tests were performed on a Smart Roadster Coupé with a 6-spd manual transmission. Since the fixed-horizon case gives an identical acceleration profile to the non-fixed horizon case, we test only the latter.

A convenient section of road was chosen as the test route. The 2km length was relatively flat and straight, and allowed unimpeded acceleration to 90kph.

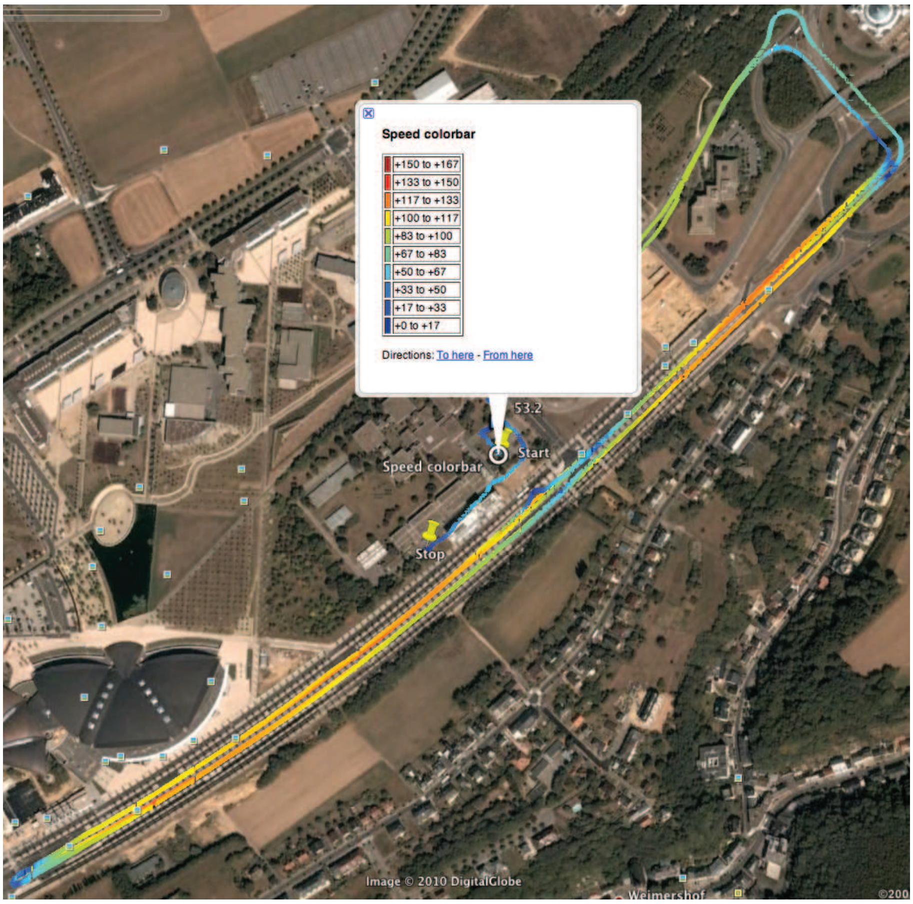


Figure 4.7: Acceleration test route (Plotted with GET, Appendix F)

While not ideal, as there were cars parked on either side of the road, we were able to perform conclusive tests that show that Pontryagin's Maximum Principle does indeed lead to a lower fuel consumption. Results can be seen in Figs. 4.8 and 4.9 and are summed up in Tab. 4.2.

Driving style	Energy used	Difference
Soft driving	2929.0kJ	-
Pontryagin trajectory	2516.4kJ	-14.1%

Table 4.2: Classic driving style vs. optimal control

There are several remarks to be made:

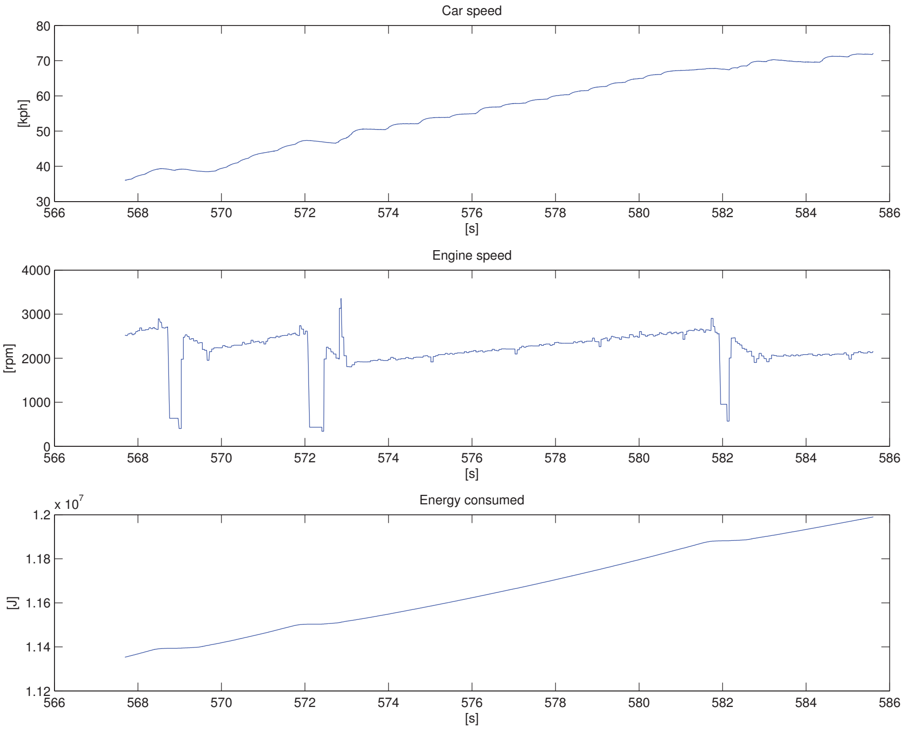


Figure 4.8: Slow acceleration test

- The improvement was seen on a driving style that would normally have been considered to be quite efficient: slow starts, easy application of throttle, no sudden movements, etc...

- These results are in 1-D only, so it makes a certain amount of sense that there is such a stark difference between optimal control and “douce” driving. The “douce” driving style takes the car much further before reaching the target speed. So while the acceleration phase is not optimal, a soft driving approach will not be as bad in a 2-D case as this data might imply.

- It is surprisingly difficult to follow the optimal trajectory, if for no other reason then that the gas pedal is a very rudimentary interface. The optimal PMP controls require relatively constant torque, which is difficult— if not impossible— to manage without instrumentation.

4.5 Two dimensions: distance on the straight and level

In this section, we search the optimal trajectory for a target speed and distance, which perhaps reflects reality better as a more common desire is to go a certain distance then it is to simply reach a certain speed. We define the target $N = \{x_i(T) = x_i^*, v_i(T) = v_i^*\}$ and suppose that x_i^* is big enough, on the order of several kilometers, in order to determine the long term optimal strategy.

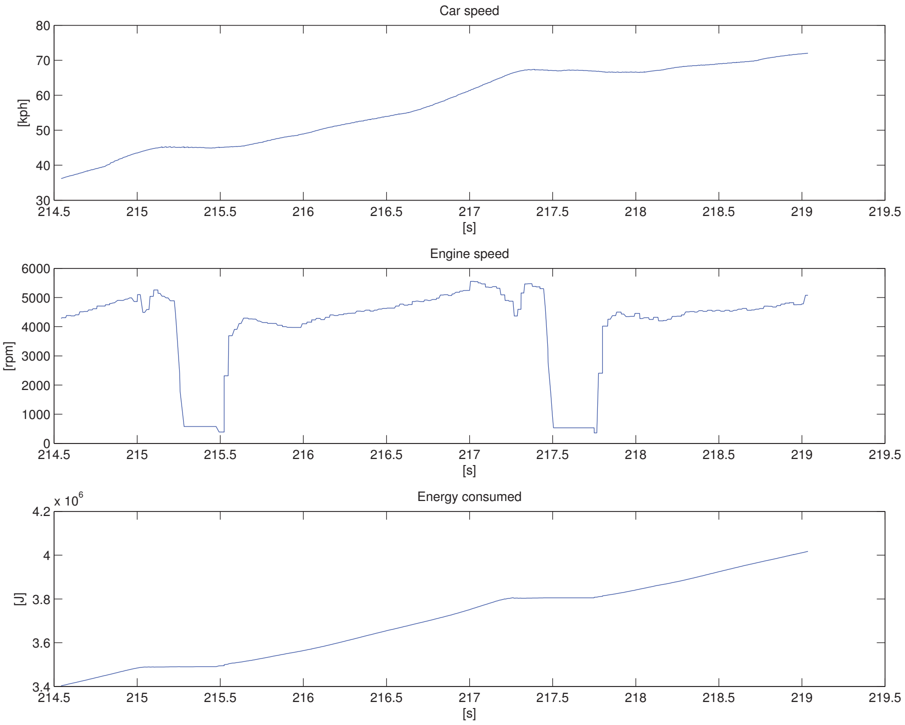


Figure 4.9: PMP optimal control acceleration test

In this case, we can no longer ignore the relation $\dot{x_i} = v$ in the now two-dimensional system (implying that the augmented system, as described in App. B.2.5, is dimension 4).

4.5.1 CVT

Returning to the CVT case— it allows for easier computations and/or formal calculations, and above all facilitates the theoretical work and taking into account the constraints— and recalling that $N_{spd} = \dfrac{\omega_e}{v_i}$ is non-constant in a CVT equipped car, we see that the Hamiltonian is now written as in eq. (4.2)

$$H = \lambda_x v_i + \lambda_v \frac{1}{M_a}\left(T_e \frac{\omega_e}{v_i} - \left(\frac{1}{2}\rho C_d A v_i^2 + M C_{rr} g\right)\right) - \omega_e T_e \phi(\omega_e, T_e)$$

The adjoint vector system is thus

$$\begin{aligned} \dot{\lambda_x} &= -\frac{\partial H}{\partial x} &&= 0 \\ \dot{\lambda_v} &= -\frac{\partial H}{\partial v} &&= \lambda_v \frac{1}{M_a}\left(\frac{\omega_e T_e}{v_i^2} + (\rho C_d A v_i)\right) - \lambda_x \end{aligned} \tag{4.4}$$

from which we see that λ_x is constant.

Suppose $\lambda_v(0)$ is fixed. Since the initial system conditions (but only system, *not* augmented) are now fixed, that is to say $v(0) = v_{i,0}$ and $x(0) = 0$, we can determine $\lambda_x(0)$ with the following condition:

$$\max_{N_{spd},T_e} H(v_{i,0}, 0, \lambda_v, \lambda_x, N_{spd}, T_e) = 0$$

We now have all the initial conditions for the augmented system $(v_i(0), x(0), \lambda_v(0), \lambda_x(0))$ and can now integrate this system. If v_i stays positive, we thus obtain a time T_f for which $x(T_f) = x^*$. For this time T_f, we will also have calculated $\lambda_v(T)$. Thus for a given $\lambda_v(0)$, we can calculate $\lambda_v(T)$ at the final time T_f

The transversality conditions implies that $\lambda_v(T_f) = 0$. We thus come, again, at a shooting problem, where the objective is to find $\lambda_v(0)$ such that $\lambda_v(T) = 0$

Results are seen in Fig. 4.10. The controls stay at their lower limits for the entire trajectory, and thus the car starts off at $v_i(0)$ and advances under its own inertia, eventually slowing down to a steady state speed for this minimal power condition.

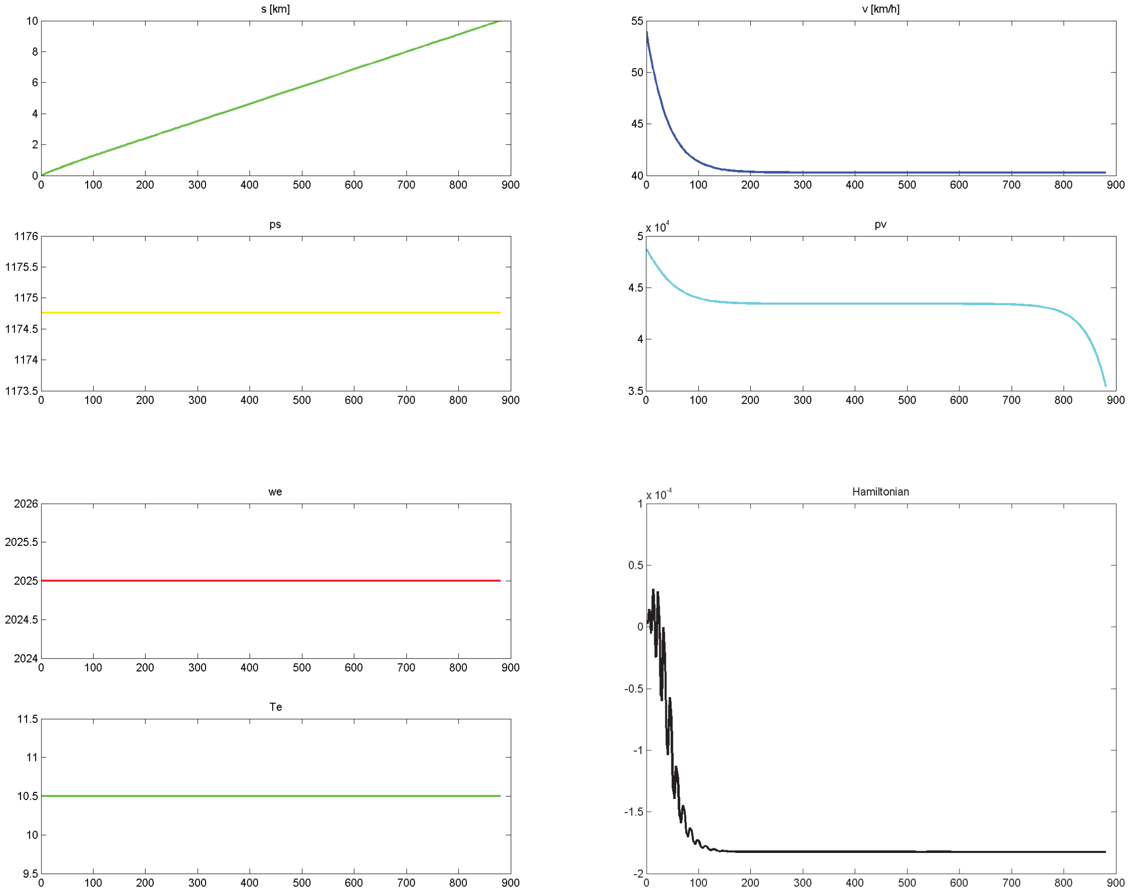


Figure 4.10: Optimization in two dimensions: CVT, T_f non-fixed

Note that the initial value $\lambda_v(0)$ is extremely difficult to determine exactly, and thus the problem is numerically very sensitive.

We can explain this sensitivity by numerically calculating $\lambda_v(0)$. For this, we will calculate the speed that corresponds to the minimum power and show its optimality.

The minimal power $\widetilde{P}$ corresponds to the lowest possible ω_e, T_e (the lower-left point of the control space graph). We note that

$$\widetilde{P} = \frac{\widetilde{\omega}_e \widetilde{T}_e}{\eta(\widetilde{\omega}_e, \widetilde{T}_e)} \tag{4.5}$$

By realizing that the force on the car is found by $P = Fv$, we can solve for the corresponding steady-state speed v_i:

$$\frac{1}{2}\rho C_d A v_i^3 + C_{rr} M g v_i = \widetilde{\omega}_e \widetilde{T}_e \tag{4.6}$$

Let

$$\begin{aligned} p &= \frac{C_{rr} M g}{\frac{1}{2} C_d A} \\ q &= -\frac{\widetilde{\omega}_e \widetilde{T}_e}{\frac{1}{2} C_d A} \end{aligned} \tag{4.7}$$

Then by finding the cubic root of eq. 4.6, we obtain

$$\widetilde{v}_i = \sqrt[3]{\frac{-q + \sqrt{q^2 + \frac{4}{27} p^3}}{2}} + \sqrt[3]{\frac{-q - \sqrt{q^2 + \frac{4}{27} p^3}}{2}} \tag{4.8}$$

With $\widetilde{v}_i$ in hand, we can look for the initial costate vector values. Since $H = 0$ and $\dot{v} = 0$, we get $\widetilde{P} = \lambda_x v_i$ which gives

$$\lambda_x(0) = \widetilde{\lambda}_x = \frac{\widetilde{\omega}_e \widetilde{T}_e}{v_i \eta(\widetilde{\omega}_e, \widetilde{T}_e)} \tag{4.9}$$

To find $\lambda_v(0)$, recall that by using the costate equation and the transversality condition

$$\begin{aligned} \dot{\lambda}_v &= \lambda_v \frac{1}{M} \left(\frac{\widetilde{\omega}_e \widetilde{T}_e}{\widetilde{v}_i^2} + \rho C_d A \widetilde{v}_i \right) - \lambda_x \\ \lambda_v(\widetilde{T}) &= 0 \end{aligned} \tag{4.10}$$

$\widetilde{T}$ is the final time, which can be calculated from the target x_i^* since the speed is constant, and thus $\widetilde{T} = \frac{x_i^*}{\widetilde{v}_i}$

By integrating in reverse time, that is to say *time-to-go*, from s=0 to $s = \widetilde{T}$ we have

$$\lambda_v(0) = \frac{\lambda_x}{\alpha} \left(1 - e^{-\alpha \widetilde{T}} \right) \tag{4.11}$$

where $\alpha = \frac{1}{M} \left(\frac{\widetilde{\omega}_e \widetilde{T}_e}{\widetilde{v}_i^2} + \rho C_d A \widetilde{v}_i \right)$

Eq. (4.11) shows the extreme sensitivity of the shooting problem when $\widetilde{T}$ is large (which will generally happen when x_i^* is large). Nonetheless, the control problem itself is numerically very stable, since the control is constant and on the limit of the control space. This calculation shows the optimality of the trajectory, at the condition of verifying afterwards that $(\widetilde{\omega}_e, \widetilde{T}_e)$ do indeed satisfy the conditions for maximizing the Hamiltonian. This can be done numerically, or even formally, since the considered polynomials are particular and of a known degree. It is also necessary to show that this extremal is optimal, which is clearly the case here, by uniqueness.

Chapter 5

Conclusion and Discussion

As cars become more and more advanced, we will have unheard of access to sensors. Thanks to these improvements, the principles of optimality as described in this paper are increasingly possible in a real-world scenario

This dissertation has shown that it is possible to create a car efficiency model from an observer based on cheap and readily available sensors. This efficiency model can then be used to create an optimal control profile.

5.1 Impact and future work

5.1.1 Sensor suite

The sensor suite proved to be very reliable and accurate. Since the beginning of the dissertation work, the total cost manufacturing cost has dropped significantly, by almost 50%. It is currently being extended to measure data in all cars, especially hybrid and diesel cars. These future models will be able to directly read the CAN bus, reducing the amount of additional sensors necessary.

The third version is undergoing testing, and the sensor suite is already seeing usage in other laboratories. Aside from measuring data directly related to a car's functioning, there are many future applications for sensor suite/observer combination, such as more accurate road mapping, traffic monitoring, etc...

5.1.2 Observer

The observer as presented in this PhD provides a basis for estimating a car's efficiency with a high degree of confidence. While there are always additional model parameters that can be taken into account for an even higher precision, the observer already gives consistent, repeatable results. The observer results are in line with results taken from a test bench dynamometer.

The hope is that this observer can be used to give researchers options for studying car efficiency and related field, other than depending on car manufacturers. Furthermore, this observer gives results that are certainly better than the original manufacturer's data, as the results are a function of a specific car, with all the variation that can occur during a car's long life.

Likewise, individual engineers and *makers* will be able to pursue their projects without having to have a large organization backing them in order to convince automotive manufacturers to divulge information.

Lastly, the results are repeatable enough that once the efficiency map is established, the engine model can be used to estimate other car parameters. For instance, it would be possible to estimate rolling resistance and thus give a sort of low-pressure tire warning. It is also possible to estimate mass, or to see the change that aerodynamic modifications make in the drag coefficient. It could furthermore be used to track slow deteriorations in the motor's general health, serving as an early warning system for engine maintenance.

Future work is to continue developing the AEKF for estimating the system state, as it promises to solve some of the observation problems, e.g. that the noise frequency can actually be slower than the perturbation frequency and that the model is in constant perturbations since there is no control input. See App. A.4 and [45].

5.1.3 Controller

As it is difficult to verify directly the fuel efficiency as predicted from the model in a non-lab environment, the optimal control provides a further validation of the efficiency model. Furthermore, while more cases must be studied for a complete application, the results presented here show that the optimal control approach is sound.

Nonetheless, while the PMP controller is easy to understand, it is difficult to implement onboard a car. Significant advances in ergonomics must be made before such a controller can be used in everyday scenarios.

Possible implementation strategies are:

- A gas pedal that when pushed to a certain position, the floor for instance, triggers the optimal control mode.
- An in-car meter that displays the costate vector
- A display that uses an up arrow or down error to indicate speeds which are too fast or too slow. The intensity of the arrow can indicate whether the setpoint error is small or large.
- A meter that displays engine efficiency, not fuel efficiency. Such a meter would take into account the kinetic and potential energy stored in a car.

It is hoped that any future studies explore first the implementation side, as it is becoming increasingly clear that the optimal control and the "average" control are radically different. The "average" control has developed naturally after a century of driving, so it is likely that the optimal control will in some way upset the traditional way of driving.

However, there are still possibilities to improve control while allowing drivers to continue driving as they always have. For instance, the car alternator can be switched on and off at appropriate times, as could the air conditioning. In hybrid cars, the battery SOC (state-of-charge) can be managed intelligently as a function of road profile.

Appendices

Part I

Theory

"In theory, there is no difference between theory and practice. In practice there is."

–Yogi Berra

Appendix A

Observe

Contents

A.1 Observers and observability

Alone, measured data is of little use. It is often too noisy to be directly added to the model, and so some amount of filtering and processing is first necessary. The filter of choice is a model-based parametric filter, one that combines the system model predictions and the real-world measurements in order to estimate the internal state. In a deterministic case, these filters are known as observers.

An observer's purpose is to:

1. Reconstruct a system's state based on measurements.

2. Suppress system and measurement noise.

3. Reconciliate data

In many, if not most, cases, it is unfeasible to measure all system states. This can be for reasons of cost, space, impracticality, or simply impossibility. Thus we employ the observer to estimate these "hidden" states.

Generally, system models can be expressed in the following terms[1] (although we will see in Section A.2 that they can be written slightly differently, depending on the time domain):

$$\begin{cases} d\mathbf{x}(t) &= f(\mathbf{x},\mathbf{u})dt + d\mathbf{w}(t) \\ d\mathbf{y}(t) &= h(\mathbf{x},\mathbf{u})dt + d\mathbf{v}(t) \end{cases}$$

where $\mathbf{x}$ is the state, $\mathbf{y}$ the measured output, $\mathbf{u}$ the control input, and $d\mathbf{w}, d\mathbf{v}$ random variables representing process and measurement noise, respectively.

Note 14 *The above model expresses the system in the most general sense. Traditionally, the noises $d\mathbf{w}(t), d\mathbf{v}(t)$ are struck from the equation as they are by definition non-derivable. Furthermore, controls texts and publications oftentimes present the affine case, $\frac{d\mathbf{x}}{dt} = \mathbf{A}(\mathbf{x}) + g(\mathbf{x})\mathbf{u}$ or even the linear case, $\frac{d\mathbf{x}}{dt} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$. In the dissertation, except when noted the most general case $f(\mathbf{x},\mathbf{u})$ is always presented.*

The challenge is to filter the noises $d\mathbf{w}$ and $d\mathbf{v}$ without filtering real-world dynamics, all while still responding rapidly to perturbations. Except in simple linear cases, we will always be forced to make some compromises, and this is before considering other issues, such as numerical stability, etc... However, some very good methods already exist and we will examine them in Section A.3.

This chapter presents an overview of observers and describes the three major time-domains used in sytem modeling, along with a discussion of techniques for handling data fusion, such as redundant and asynchronous data. Algorithms for two current filters, the Extended Kalman Filter and the Unscented Kalman Filter, are given, along with a discussion of the new AEKF.

[1] This representation assumes a Gaussian noise distribution. This is not strictly true, and in fact is likely not true for most models. However, the Gaussian assumption leads to mathematically proven observers that have proven to be robust even when the Gaussian assumption is patently false. The most general model would be $f(x, u, w) = \Phi$, but this is so abstract as to lose applicability to many real-world problems.

A.1.1 Observer

In rigorous terms, an observer is a mathematical algorithm that reconstructs the entire state of a deterministic system based only on the system model and its outputs. One common family of observer is the linear asymptotic observer, a model of the form:

$$\hat{\mathbf{x}}_{k+1} \quad = \quad \mathbf{A}\hat{\mathbf{x}}_k + \mathbf{B}\mathbf{u}_k + \mathbf{L}(\mathbf{y}_k - h(\hat{\mathbf{x}}_k)) \tag{A.1}$$

where the matrix $\mathbf{L}$ is such that $\hat{\mathbf{x}}$ approaches $\mathbf{x}$ as $k \to \infty$ for all initial conditions $\mathbf{x}(0)$ and $\hat{\mathbf{x}}(0)$. This type of matrix is called a Luenberger observer.

The Kalman Filter can be seen as a special variant of a Luenberger observer for an observable time-invariant system at T_∞. However, the Kalman method is not the only way to create a Luenberger observer. Indeed, any matrix $\mathbf{L}$ that places the eigenvalues of $(\mathbf{A}-\mathbf{LC})$ in such a way that their real components are all negative is of this family.

A.1.2 Observability

Observability describes whether or not there is sufficient information to observe all the system states. For instance, imagine that we wanted to know a car's speed between point A and B. As speed is just distance over time, if we have a distance measurement (odometer) and a time measurement (clock), we can calculate speed. This simple example system is *observable.*

Now, imagine we define a new system, one with states *speed*, *distance*, and *path*. This time we want to know the path the car took. If we still only have two instruments, the odometer and the clock, clearly we do not have enough information to observe the car's path. We would need additional information, such as a GPS or overhead photos of the car along its route. Thus, with only an odometer and clock this extended system is **not** observable.

Definition: Roughly speaking, a system is *observable* if, by knowing the controls leading up to a state and knowing that outputs of that state, it is possible to backtrack in time in order to find an earlier state, or apply the dynamic equations in order to predict where the system will be in the future.

In more rigorous terms, a system is said to be *observable* if for every vector $\mathbf{x}_d$ in $\mathbb{R}^n$ there is a finite time T such that if $\mathbf{x}(0) = \mathbf{x}_d$ and one applies an arbitrary control $\mathbf{u}$, the knowledge of the trajectory on $\mathbf{y}$ during the time interval $[0, T]$ allows one to determine $\mathbf{x}_d$.

This definition of observability is the basic definition that corresponds to our intuition. There are, however, many more nuanced definitions possible, such as for local, infinitesimal, etc..., that are sometimes for useful. For supplemental definitions, see [13].

Linear observability: In the linear case, the observability test is straightforward. For a linear system of form:

$$\begin{cases} \dfrac{d\mathbf{x}}{dt} & = & \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \\ \mathbf{y}(t) & = & \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u} \end{cases}$$

the system is observable if the *Kalman Criterion Matrix*

$$\mathbf{Q}_o = \begin{pmatrix} \mathbf{C} \\ \mathbf{CA} \\ \vdots \\ \mathbf{CA}^{n-1} \end{pmatrix} \tag{A.2}$$

is of full rank n_x.[19]

Non-linear observability: In the nonlinear case, it can happen that a system is observable at certain states and non-observable at others. (For instance, in the system described by eqns (1.16), (1.17), (1.18), when $v_i = 0$ then the system becomes non-observable.)

It is difficult to talk about global observability. Instead, we are interested in locally observable systems. First, however, we must make a quick detour to define *distinguishability.*

Definition: distinguishability– Two states $\mathbf{x}^0(0)$ and $\mathbf{x}^1(0)$ are distinguishable if there exists a bounded input $\mathbf{u}$ such that $\mathbf{y}^0(t) \neq \mathbf{y}^1(t)$ for some $t \in [0, T]$

Definition: local observability– A non-linear system is locally observable at $\mathbf{x}_0$ if there exists a neighborhood of $\mathbf{x}^0$ such that every $\mathbf{x}$ in that neighborhood other than $\mathbf{x}^0$ is distinguishable from $\mathbf{x}^0$.

A.1.2.1 Observability test by Lie derivative

One test for local observability depends on the *Lie derivative.* (The derivation and purpose of Lie derivatives is far outside the scope of this PhD. If the reader would like further information, s/he is encouraged to read Boothby's excellent description.[11])

The Lie derivative of h with respect to f is

$$L_f h = \nabla h f = \frac{\partial h}{\partial \mathbf{x}} f$$

By convention, $L_f^0(h) = h$. Higher-order Lie derivatives are give by:

$$L_f^n(h) = \frac{\partial}{\partial \mathbf{x}} \left[L_f^{n-1}(h) \right] f$$

Thus, let

$$l(\mathbf{x}, \mathbf{u}) \equiv \begin{bmatrix} L_f^0(h) \\ \vdots \\ L_f^{n_x - 1}(h) \end{bmatrix} \tag{A.3}$$

where f, h are the dynamic and measurement models, as defined in eq. (A.1).

Note 15 *If $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u})$, then $\dot{h} = \dfrac{\partial h}{\partial x}\dfrac{\partial x}{\partial t} = \dfrac{\partial h}{\partial x} f = L_f^1(h)$. Similarly, $\ddot{h} = L_f^2(h)$, $\dddot{h} = L_f^3(h)$, and so on and so forth. Thus the vector $l(\mathbf{x}, \mathbf{u})$ is nothing more than a series of derivatives of the output model.*

Finally the test for local observability is that

$$\mathbf{Q}(\mathbf{x_0}, \mathbf{u}) \equiv \frac{\partial l(\mathbf{x_0}, \mathbf{u})}{\partial \mathbf{x}} \tag{A.4}$$

is rank n_x

A.1.2.2 Observability test by normal form

A second test for observability is whether the model can be put into the normal form of observability. Indeed, all methods by which the model is brought into the normal form are by definition methods that show observability (no matter the change of coordinates, observability is an intrinsic property).[21] Therefore, we can verify observability and compute the change of coordinates in one single action, as below.

Note 16 *Both the Lie Derivative and the Observability Normal Form approaches are valid, but in Sec. 1.2.3 we present the normal form of eq. (1.15), as it has further uses beyond demonstrating observability.*

The normal structure is unique in the single output case, and thus in this special case we can call it the *canonical form*, but not in the more general multiple output case the normal structure is not unique. We use the most common generalization, but there has been no study as of yet to create a sort of "zoology" of normal forms.[7]

A.2 Time domains

Presented here is a short discussion of the three time domains into which systems and their observations can fall, following whether or not the dynamics and/or the outputs are discretized.

A.2.1 Continuous-continuous

These models are of the form

$$\left\{ \begin{array}{lcl} \dot{\mathbf{x}} & = & f(\mathbf{x}(t), \mathbf{u}(t), t) \\ \mathbf{y} & = & h(\mathbf{x}(t), \mathbf{u}(t), t) \end{array} \right. \tag{A.5}$$

These models assume that the system is continuous in time, and that the measurements are also continuous. In consequence, the prediction of the state and state covariance matrices are continuous.

Note 17 *These cases are frequent in mathematical analyses, but are unlikely to be analyzed by an observer in real-world situations. Observers are almost uniformly implemented on digital computers, and so the following two cases that have a discrete component to the outputs are of more interest in real-world controls problems.*

A.2.2 Discrete-discrete

These models are of the form

$$\left\{\begin{array}{lcl}\mathbf{x}_k & = & f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, k)\\ \mathbf{y}_k & = & h(\mathbf{x}_k, \mathbf{u}_k, k)\end{array}\right. \tag{A.6}$$

Observers based on this model rely on a discrete prediction of the state and the state covariance matrix, and a discrete correction of these two. This is the most commonly used, as it is the easiest and cheapest to program.

For linear systems, there is little reason to stay in continuous time when designing an observer. Time-invariant linear systems can be converted to discrete-time systems by [19]

$$\dot{\mathbf{x}} \quad = \quad \mathbf{Ax} + \mathbf{Bu} \Rightarrow \mathbf{x}(t) = e^{(t-t_0)\mathbf{A}}\mathbf{x_0} + \int_{t_0}^{t} e^{(t-\tau)\mathbf{A}}\mathbf{Bu}(\tau)d\tau \tag{A.7}$$

which, with the condition that $\mathbf{u}_k$ is constant along the interval $[t_k, t_{k+1}]$, yields

$$\left\{\begin{array}{rcl}\mathbf{x}_{k+1} & = & \mathbf{A}_d\mathbf{x}_k + \mathbf{B}_d\mathbf{u}_k\\ \mathbf{y}_k & = & h(\mathbf{x}_k, \mathbf{u}_k, k)\end{array}\right. \tag{A.8}$$

Furthermore, if $\mathbf{A}$ is invertible:

$$\mathbf{A}_d = e^{\mathbf{A}T}, \quad \mathbf{B}_d = \mathbf{A}^{-1}(e^{\mathbf{A}T-I})\mathbf{B} \tag{A.9}$$

In the more general case, the continuous time-domain system (A.5) can be discretized by

$$\left\{\begin{array}{rcl}\mathbf{x}_{k+1} & = & \Phi(\mathbf{x}_k)\\ \mathbf{y}_k & = & h(\mathbf{x}_k, \mathbf{u}_k, k)\end{array}\right.$$

where Φ is the resolvent[2] of the ODE in (A.5) between t_k and t_{k+1} and where

$$\mathbf{A}_d \quad = \quad \frac{\partial\Phi(\mathbf{x}_k)}{\partial\mathbf{x}_k}$$

which can be calculated by using the Jacobian of $A(\mathbf{x})$, denoted $\mathbf{A}_c = \dfrac{\partial A(\mathbf{x})}{\partial\mathbf{x}}$

$$\left\{\begin{array}{rcl}\dot{\mathbf{A}}_d & = & \mathbf{A}_c\mathbf{A}_d\\ \mathbf{A}_d(t_k) & = & \mathbf{I}\ (=\text{Identity})\end{array}\right.$$

However, the most common and expedient approach to discretizing continuous models is to first linearize the model about its current state and then discretize as in eq. (A.7). This implies taking the Jacobian, either numerically or analytically. For finite steps in time, this is inherently less accurate than using a continuous model for prediction.

[2] The resolvent is the integral of $\mathbf{A}_c$ along the path $\mathbf{x}(t)$. It has the interesting property that $\Phi_{t_{k+1}/t_k}\Phi_{t_k/t_{k-1}} = \Phi_{t_{k+1}/t_{k-1}}$ and so $\mathbf{x}_{k+1} = \Phi_{t_{k+1}/t_0}\mathbf{x}(t_0)$

A.2.3 Continuous-discrete

These models are of the form

$$\begin{cases} \dot{\mathbf{x}} &= f(\mathbf{x}(t), \mathbf{u}(t), t) \\ \mathbf{y}_k &= h(\mathbf{x}_k, \mathbf{u}_k, k) \end{cases} \tag{A.10}$$

The continuous-discrete variant is a hybrid, best-of-both-worlds solution for physical problems treated on digital computers. It models a dynamic system that changes in continuous time, but has measurements arriving at discrete intervals.

In fact, the measurements do not even need to be synchronized. The ease with which asynchronous measurements is handled is a great advantage of the continuous-discrete system over the discrete-discrete one.

For the vast majority of physical systems, continuous-discrete time is the logical choice, since they fall neatly into the two time domains. Such systems change continuously with time t, but the data measurements are at discrete times t_k. Nonetheless, discrete-discrete can be interesting in terms of numerical computation time.

Note 18 *Continuous-discrete systems pose an interesting theoretical problem for the definition of observability. The attentive reader will notice that there is no guarantee that there is ever a time when all measurements arrive simultaneously. This leads to a good question: "Is this system observable?" In this specific case, the answer is clearly "yes". However, to the best of the authors' knowledge there is no general study of observability for asynchronous measurements, as in [21] for the continuous case and [2] for the adaption to the discrete case. The problem is clearly not obvious since one can build an academic system for a continuous-time model which is observable for continuous measurements and for synchronous discrete measurements, but not for asynchronous discrete measurements.*[3]

[3] **Example:** Note $\Pi = \{(a,b) \in \mathbb{R}^2; a^2 + b^2 = 1\}$ the unit circle.
Let $X = \Pi \times \mathbb{R}_+^* \times \mathbb{R}_+^*$ and the system Σ on X.

$$\begin{cases} \dot{x}_1 &= -x_2 x_4 \\ \dot{x}_2 &= x_1 x_4 \\ \dot{x}_3 &= 0 \\ \dot{x}_4 &= 0 \end{cases}, \quad \begin{cases} y_1 &= x_1 x_3 \\ y_2 &= x_2 x_3 \\ y_3 &= x_4 \end{cases}$$

and $x_3, x_4 > 0$

1. The system Σ is observable, whether it is continuous or discrete. Specifcally, $y_1^2 + y_2^2 = x_3^2(x_1^2 + x_2^2) = x_3^2$ give x_3 and thus y_1 and y_2 suffice to give x_1 and x_2
2. The asynchronous system is **non-observable** because there are certain values for x_4 which makes x_3 indistinguishable.

   Suppose that y_1 and y_2 are measured alternatively at a step time Δt. The solution to Σ is

   $$\begin{cases} x_1 &= \cos(x_4 t + C_1) \\ x_2 &= \sin(x_4 t + C_1) \\ x_3 &= C_2 \\ x_4 &= C_3 \end{cases}$$

   If Δt is poorly chosen such that $x_4 \Delta t = \frac{\pi}{2} + 2k\pi$ then

A.3 Kalman Filters

Kalman Filters refer to a wide class of model-based stochastic filters. While originally developed as a discrete-time linear filter [32], the Kalman Filter was quickly extended (no pun intended) to continuous time and non-linear systems[26]. In the 50 years since Kalman's seminal paper, the term "Kalman Filter" has grown to encompass a much larger series of filters.

It goes without saying that different filters will have different advantages and disadvantages. For instance, the Extended Kalman Filter has good noise rejection properties, but can be slow to respond to perturbations, and global convergence has never been shown. Indeed, to the contrary, it has been shown that it does not globally converge.

The High-gain Kalman Filter[21] demonstrates global convergence, but it overly sensitive to noise. The Adaptive High-gain Extended Kalman Filter combines the EKFs noise rejection properties with the HGKF's responsiveness, but adds a number of additional parameters to be tuned.[7] The Unscented Kalman Filter presents interesting possibilities for better predictions of non-linear systems, but adds additional tuning parameters that are poorly understood, greatly increases computation time, and does not cope well with a non-linear measurement model.

We present in the following two Kalman Filter variants that are seeing wide application, the EKF and the UKF. These filters were chosen as likely candidates for modeling η (Chap. 1). We also present a new Kalman filter that presents an interesting direction for further study.[45]

A.3.1 Extended Kalman Filter

The Extended Kalman Filter (EKF) is a tried-and-true approach used extensively in science and engineering. While there are other filters available, such as the Unscented Kalman Filter [30], the DD1 Filter [36], the High Gain Kalman Filter [21], etc..., the EKF has proven to be a very suitable approach for a wide class of problems, including vehicle localization[33]. Some of EKF's advantages are simplicity, computational costs, tuneability, and years of experience in real-world applications.

The EKF requires calculating the Jacobians of various model functions.[4] We use the

$$\begin{cases} \begin{array}{rcccc} y_1(0) & = & x_3\cos(C_1) & & \\ y_2(\Delta t) & = & x_3\sin(C_1+\frac{\pi}{2}) & = & x_3\cos(C_1) \\ y_1(2\Delta t) & = & x_3\cos(C_1+\frac{2\pi}{2}) & = & -x_3\cos(C_1) \\ y_2(3\Delta t) & = & x_3\sin(C_1+\frac{3\pi}{2}) & = & -x_3\cos(C_1) \\ & \vdots & & & \end{array} \end{cases}$$

and so it becomes impossible to find both x_3 and C_1, with the obvious conclusion that the system is non-observable.

[4] Indeed, it is its principle disadvantage as the Jacobian is a first order approximation of a non-linear system at a specific point, a point which is only an estimation of the real state, and so by its very nature is sub-optimal, and can affect the filter accuracy or even lead to divergence of the filter. Many other filters have been developed that seek to address this shortcoming. See Section A.3.2 for an example of a popular one.

following shorthand to represent commonly used Jacobians:

$$\begin{aligned} \mathbf{A}_c &= \frac{\partial A}{\partial \mathbf{x}_i} \\ \mathbf{B}_c &= \frac{\partial A}{\partial \mathbf{u}_i} \\ \mathbf{H}_c &= \frac{\partial H}{\partial \mathbf{x}_i} \end{aligned} \tag{A.11}$$

A.3.1.1 Continuous-continuous EKF

In the CC-EKF, both the model and the measurements are continuous in time. This is the most classic form of the EKF, and yet the hardest to use in a strictly correct sense.[5]

$$\begin{aligned} \dot{\hat{\mathbf{x}}} &= A(\hat{\mathbf{x}}, \mathbf{u}) + \mathbf{K}(t)\left(\mathbf{y}(t) - h(\hat{\mathbf{x}}, \mathbf{u})\right) \\ \dot{\hat{\mathbf{P}}} &= \mathbf{A}_c\hat{\mathbf{P}} + \hat{\mathbf{P}}\mathbf{A}_c^T + \mathbf{Q}_c - \mathbf{K}(t)\mathbf{R}_c\mathbf{K}^T(t) \end{aligned} \tag{A.12}$$

where

$$\mathbf{K}(t) = \hat{\mathbf{P}}\mathbf{H}_c\mathbf{R}_c^{-1} \tag{A.13}$$

and where $\mathbf{Q}_c$ is a positive semi-definite matrix in $\mathbb{R}^{n_x \times n_x}$ and $\mathbf{R}_c$ is a positive-definite matrix of dimension $\mathbb{R}^{n_y \times n_y}$

A.3.1.2 Discrete-discrete EKF

The DD-EKF has a discrete model and discrete measurements. Common engineering and scientific usage is to discretize linear and non-linear systems as this form is particularly cheap to calculate.

Prediction:

$$\begin{aligned} \hat{\mathbf{x}}_k^- &= f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, t_{k-1}) \\ \hat{\mathbf{P}}_k^- &= \mathbf{A}_k\hat{\mathbf{P}}_{k-1}\mathbf{A}_k^T + \mathbf{Q}_d \end{aligned} \tag{A.14}$$

where $\mathbf{Q}_d$ is a positive semi-definite matrix in $\mathbb{R}^{n_x \times n_x}$

Correction:

$$\begin{aligned} \mathbf{K}_k &= \hat{\mathbf{P}}_k^-\mathbf{H}_k^T(\mathbf{H}_k\hat{\mathbf{P}}_k^-\mathbf{H}_k^T + \mathbf{R}_k)^{-1} \\ \hat{\mathbf{x}}_k &= \hat{\mathbf{x}}_k^- + \mathbf{K}_k\left(\mathbf{y}_k - h(\hat{\mathbf{x}}_k^-, \mathbf{u}_k)\right) \\ \hat{\mathbf{P}}_k &= (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\hat{\mathbf{P}}_k^- \end{aligned} \tag{A.15}$$

where $\mathbf{R}_k$ is a positive-definite matrix of dimension $\mathbb{R}^{n_y \times n_y}$

[5] Instead of using a true analog computer for the correction, a daunting task to say the least, digital computers running at extremely high sample rates are used instead.

A.3.1.3 Continuous-discrete EKF

The CD-EKF is used for systems with continuous dynamics, but discrete measurements.

Prediction:

$$\begin{aligned} \hat{\mathbf{x}}_k^- &= \hat{\mathbf{x}}_{k-1} + \int_{t_{k-1}}^{t_k} \frac{d\mathbf{x}}{dt} dt \\ \hat{\mathbf{P}}_k^- &= \hat{\mathbf{P}}_{k-1} + \int_{t_{k-1}}^{t_k} \frac{d\hat{\mathbf{P}}}{dt} dt \end{aligned} \tag{A.16}$$

where $\frac{d\hat{\mathbf{x}}}{dt} = f(\hat{\mathbf{x}}, \mathbf{u})$ and

$$\frac{d\hat{\mathbf{P}}}{dt} = (\mathbf{A}_c\hat{\mathbf{P}} + \hat{\mathbf{P}}\mathbf{A}_c^T + \mathbf{Q}_c) \quad \Big| \quad \hat{\mathbf{P}}|_{t_{k-1}} = \hat{\mathbf{P}}_{k-1} \tag{A.17}$$

and $\mathbf{Q}_c$ is a positive semi-definite matrix in $\mathbb{R}^{n_x \times n_x}$

Correction:

$$\begin{aligned} \mathbf{K}_k &= \hat{\mathbf{P}}_k^- \mathbf{H}_k^T (\mathbf{H}_k \hat{\mathbf{P}}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \\ \hat{\mathbf{x}}_k &= \hat{\mathbf{x}}_k^- + \mathbf{K}_k \left(\mathbf{y}_{t_k} - h(\hat{\mathbf{x}}_k^-, \mathbf{u}_k)\right) \\ \hat{\mathbf{P}}_k &= (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)\hat{\mathbf{P}}_k^- \end{aligned} \tag{A.18}$$

where $\mathbf{R}_k$ is a positive-definite matrix of dimension $\mathbb{R}^{n_y \times n_y}$

The continuous-discrete Kalman Filter has found applications in diverse industries. One of the advantages of the CD-EKF is that it allows for continuous integration until a new measurement arrives, making it, when seen under a certain light, the fastest filter, i.e. it has the shortest computational time between measurement input and calculation of the state estimate. Thus in spite of additional integration costs vs. the discrete-discrete filter, the continuous-discrete can be the best solution for a real-time observation problem.

A.3.2 Unscented Kalman Filter

The Unscented Kalman Filter is a recent innovation [30] that is based on the assumption that it is easier to approximate a gaussian distribution than it is to approximate a non-linear function.

The UKF works by creating an n-dimensional cloud of sigma points around the expected mean. These points are precisely determined in order to exactly approximate the first, second, third, and fourth moments (*kurtosis*). The exact number of sigma points required for such an approximation is $2n + 1$, where n is the number of states in the dynamic system. [28]

The properties of the UKF have been studied in detail elsewhere [29, 31], but we sum up the primary advantages here:

- As the mean and covariance of $\mathbf{x}$ are reproduced to the second order, the calculated values of the mean and covariance of $\mathbf{y}$ are correct to the second order as well. This implies that the mean is calculated to a higher order of accuracy than the EKF, whereas the covariance is calculated to the same order of accuracy.[30]

- UKF requires no calculation of derivatives or Jacobians, operations that are generally non-trivial, human error-prone, and can lead to significant implementation difficulties.

The UKF was originally designed as a discrete-discrete filter. In additional to the covariance parameters— similar to the EKF case— there are three additional parameters that must be chosen: α, β, κ

Note: if the true distribution of $\mathbf{x}$ is Gaussian, $\beta = 2$ is optimal.

Prediction:

$$\begin{aligned}
\mathbf{X}_{k-1} &= [\hat{\mathbf{x}}_{k-1} \; \ldots \; \hat{\mathbf{x}}_{k-1} \; \ldots \; \hat{\mathbf{x}}_{k-1}] + \sqrt{c}\begin{bmatrix}0 & \sqrt{\hat{\mathbf{P}}_{k-1}} & -\sqrt{\hat{\mathbf{P}}_{k-1}}\end{bmatrix} \\
\mathbf{X}_k &= f_d(\mathbf{X}_{k-1}, k-1) \\
\hat{\mathbf{x}}_k^- &= \mathbf{X}_k \mathbf{w}_m \\
\hat{\mathbf{P}}_k^- &= \mathbf{X}_k \mathbf{W} \mathbf{X}_k^T + \mathbf{Q}_{k-1}
\end{aligned} \tag{A.19}$$

where

$$\begin{aligned}
c &= \alpha^2(n+\kappa) \\
w_m &= [W_m^{(0)} \; \ldots \; W_m^{(2n)}]^T \\
\mathbf{W} &= (\mathbf{I} - [w_m \; \ldots \; w_m]) \times \text{diag}(W_c^{(0)} \; \ldots W_c^{(2n)}) \times (\mathbf{I} - [w_m \; \ldots \; w_m])^T
\end{aligned}$$

and where

$$\begin{aligned}
W_m^{(0)} &= \lambda/(n+\lambda) \\
W_c^{(0)} &= \lambda/(n+\lambda) + (1 - \alpha^2 + \beta) \\
W_m^{(i)} &= \lambda/(2(n+\lambda)), \qquad i = 1, \ldots, 2n \\
W_c^{(i)} &= \lambda/(2(n+\lambda)), \qquad i = 1, \ldots, 2n
\end{aligned}$$

Correction:

$$\begin{aligned}
\mathbf{X}_k^- &= [\hat{\mathbf{x}}_k^- \; \ldots \; \hat{\mathbf{x}}_k^- \; \ldots \; \hat{\mathbf{x}}_k^-] + \sqrt{c}\begin{bmatrix}0 & \sqrt{\hat{\mathbf{P}}_k^-} & -\sqrt{\hat{\mathbf{P}}_k^-}\end{bmatrix} \\
\mathbf{Y}_k^- &= h(\mathbf{X}_k^-, k) \\
\hat{\mu}_k &= \mathbf{Y}_k^- \mathbf{w}_m \\
\mathbf{S}_k &= \mathbf{Y}_k^- \mathbf{W} [\mathbf{Y}_k^-]^T + \mathbf{R}_k \\
\mathbf{C}_k &= \mathbf{X}_k^- \mathbf{W} [\mathbf{Y}_k^-]^T \\
\mathbf{K}_k &= \mathbf{C}_k \mathbf{S}_k^{-1} \\
\hat{\mathbf{x}}_k &= \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{y}_\mathbf{k} - \hat{\mu}_k) \\
\hat{\mathbf{P}}_k &= \hat{\mathbf{P}}_k^- - \mathbf{K}_k \mathbf{S}_k \mathbf{K}_k^T
\end{aligned} \tag{A.20}$$

However, in [41], the UKF was extended to the continuous-discrete domain, and called the Unscented Kalman-Bucy Filter. In [50], a more efficient square root form was given, which we detail below.

A.3.3 Square root filters

An algorithmic variant of classic filters is created by taking the square root of the state noise covariance matrix, **P**. In these cases, the mathematical characteristics are preserved, for instance positive-definiteness is guaranteed, and the numerical properties are improved, as computation is done more accurately because very small numbers become larger, and very large numbers smaller. [15]

A.3.3.1 Square-root continuous-discrete EKF

The following algorithm is detailed in [4, 15], but it is repeated below for ease of reference.

Assume $\mathbf{x}_0, \mathbf{P}_0$ given. (If $\mathbf{P}_0$ is not given, it can generally be initialized with the identity matrix.) Let

$$\mathbf{A}_k = e^{A(\hat{\mathbf{x}}(t_k),\mathbf{u}(t_k))\Delta t}$$

Define the following Cholesky square roots:

$$\begin{aligned} \mathbf{S}_k\mathbf{S}_k^T &= \hat{\mathbf{P}} \\ \mathbf{L}\mathbf{L}^T &= \mathbf{Q} \\ \mathbf{D}\mathbf{D}^T &= \mathbf{R} \end{aligned}$$

Prediction: Form matrix $\mathbf{L}_1$ as:

$$\mathbf{L}_1 = \begin{bmatrix} \mathbf{A}_{k-1}\mathbf{S}_{k-1} & \mathbf{L} \end{bmatrix} \tag{A.21}$$

and calculate $\mathbf{M}_1$ by taking the **Q** factor of

$$[l, \mathbf{M}_1] = \textbf{lq factorization}\{\mathbf{L}_1\} \tag{A.22}$$

where l is not further used.

Determine $\mathbf{S}_k^-$ by taking the first n_x columns of $\mathbf{M}_1$

$$\begin{bmatrix} \mathbf{S}_k^- & \mathbf{0} \end{bmatrix} = \mathbf{M}_1 \tag{A.23}$$

Integrate $\dot{\mathbf{x}}$ to find $\hat{\mathbf{x}}_k^-$

$$\hat{\mathbf{x}}_k^- = \hat{\mathbf{x}}_{k-1} + \int_{t_{k-1}}^{t_k} \frac{d\mathbf{x}}{dt} dt$$

Correction: Compute the measurement model Jacobian $\mathbf{H}_k = \mathbf{H}_c(\mathbf{x}(t_k))$ as in eq. (A.11).

Form matrix $\mathbf{L}_2$ as:

$$\mathbf{L}_2 = \begin{bmatrix} \mathbf{D} & \mathbf{H}_k\mathbf{S}_k^- \\ \mathbf{0} & \mathbf{S}_k^- \end{bmatrix} \tag{A.24}$$

and calculate $\mathbf{M}_2$ by taking the **Q** factor of

$$[l, \mathbf{M}_2] = \textbf{lq factorization}\{\mathbf{L}_2\} \tag{A.25}$$

where l is not further used.

Extract the matrices $\mathbf{W}_k, \overline{\mathbf{K}}_k, \mathbf{S}_k$

$$\begin{array}{rc} & \begin{array}{cc} \overbrace{\quad\quad}^{nx} & \overbrace{\quad\quad}^{ny} \end{array} \\ \begin{array}{r} ny\{ \\ nx\{ \end{array} & \begin{bmatrix} \mathbf{W}_k & \mathbf{0} \\ \overline{\mathbf{K}}_k & \mathbf{S}_k \end{bmatrix} \end{array} = \mathbf{M}_2 \tag{A.26}$$

Compute the Kalman gain $\mathbf{K}_k$

$$\mathbf{K}_k = \overline{\mathbf{K}}_k \mathbf{W}_k^{-1} \tag{A.27}$$

Compute final state estimate $\hat{\mathbf{x}}(t_k)$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k \left(\mathbf{y}_k - h(\hat{\mathbf{x}}_k^-, \mathbf{u})\right) \tag{A.28}$$

A.3.3.2 Square-root continuous-discrete UKF

Called the Square Root Unscented Kalman-Bucy Filter, it provides results mathematically identical to the Unscented Kalman-Bucy Filter, only for reasons of numerical stability and speed it avoids recalculating the cholesky decomposition at each time step.

The following algorithm is detailed in [41, 50], but it is repeated below for ease of reference.

Prediction:

$$\begin{aligned} \mathbf{M}(t) &= \mathbf{A}^{-1}(t)\left[\mathbf{X}(t)\mathbf{W}f^T(\mathbf{X}(t),t) + f(\mathbf{X}(t),t)\mathbf{W}\mathbf{X}^T(t) + \mathbf{Q}\right]\mathbf{A}^{-T}(t) \\ \frac{d\mathbf{X}_i(t)}{dt} &= f(\mathbf{X}(t),t)\mathbf{w}_m + \sqrt{c}\,\left[0 \quad \mathbf{A}(t)\Phi(\mathbf{M}(t)) \quad -\mathbf{A}(t)\Phi(\mathbf{M}(t))\right]_i \end{aligned} \tag{A.29}$$

where

$$\begin{aligned} c &= \alpha^2(n+\kappa) \\ \mathbf{X}(t) &= \left[\hat{\mathbf{x}}_k(t) \;\ldots\; \hat{\mathbf{x}}_k(t) \;\ldots\; \hat{\mathbf{x}}_k(t)\right] + \sqrt{c}\left[0 \quad \sqrt{\mathbf{S}(t)} \quad -\sqrt{\mathbf{S}(t)}\right] \end{aligned} \tag{A.30}$$

and

$$\Phi_{ij}(\mathbf{M}(t)) = \begin{cases} \mathbf{M}_{ij}(t) & \text{, if } i > j \\ \frac{1}{2}\mathbf{M}_{ij}(t) & \text{, if } i = j \\ 0 & \text{, if } i < j \end{cases} \tag{A.31}$$

Correction:

$$
\begin{aligned}
\mathbf{Y}_k^- &= h(\mathbf{X}_k^-, k) \\
\hat{\mu}_k &= \mathbf{Y}_m^- \mathbf{w}_m \\
\left[q, \mathbf{S}_k^-\right] &= \textbf{qr decomposition} \left\{ \left[\sqrt{\mathbf{W}_1^{(c)}} (\mathbf{Y}_{i=1:2L} - \hat{\mu}_k) \quad \sqrt{\mathbf{R}_k} \right] \right\} \\
\mathbf{S}_k &= \textbf{cholesky update}\{\mathbf{S}_k^-, \mathbf{Y}_{i=0} - \hat{\mu}_k, \mathbf{W}_0^{(c)}\} \\
\mathbf{C}_k &= \mathbf{X}_k^- \mathbf{W} [\mathbf{Y}_k^-]^T \\
\mathbf{K}_k &= (\mathbf{C}_k (\mathbf{S}_k^T)^{-1}) \mathbf{S}_k^{-1} \\
\hat{\mathbf{x}}_k &= \mathbf{x}_k^- + \mathbf{K}_k (\mathbf{y}_k - \hat{\mu}_k) \\
\mathbf{U} &= \mathbf{K}_k \mathbf{S}_k \\
\mathbf{A}_k &= \textbf{cholesky update}\{\mathbf{A}_k^-, \mathbf{U}, -1\} \qquad \text{(A.32)}
\end{aligned}
$$

The algorithm, which is not particularly obvious in the literature, is as follows:

Algorithm 2: SR-UKF-CD

```
while $t < T$ do
    form $\mathbf{X}_{t_{k-1}}$ from $\mathbf{m}_{t_{k-1}}$ and $A_{t_{k-1}}$
    while no new measurement do
        integrate $\frac{d\hat{\mathbf{X}}(t)}{dt}$
    end
    $\hat{\mathbf{X}}_k^- \leftarrow \hat{\mathbf{X}}(t_k)$

    Correction
    construct new measurement covariance, $\mathbf{R}_k$
    extract $\mathbf{A}_k^-$ from $\hat{\mathbf{X}}_k^-$
    $\mathbf{S}_k^- \leftarrow$ the square, lower-diagonal matrix $\mathbf{S}_k^-$ from QR decomposition
    $\mathbf{S}_k \leftarrow$ cholesky factor of
        $\mathbf{S}_k^- + sign\left(\mathbf{W}_0^{(c)}\right) \sqrt{\mathbf{W}_0^{(c)}} (\mathbf{Y}_{i=0} - \hat{\mu}_k)(\mathbf{Y}_{i=0} - \hat{\mu}_k)^T$
    $\mathbf{U} \leftarrow \mathbf{K}_k \mathbf{S}_k$
    for $i = 1$ to $columns(\mathbf{U})$ do
        $\mathbf{A}_k^- \leftarrow$ cholesky factor of $(\mathbf{A}_k^- - \mathbf{U}_i \mathbf{U}_i^T)$
    end

    Update loop
    $\hat{\mathbf{m}}(t_k) \leftarrow \hat{\mathbf{m}}_k$
    $\hat{\mathbf{A}}(t_k) \leftarrow \hat{\mathbf{A}}_k^-$
    $k \leftarrow k + 1$
end
```

Note 19 *Taking the cholesky factor at each iteration instead of the cholesky update is not efficient. It is written in the above way only for algorithmic clarity. An efficiency SR-UKBF should use a* cholseky update.

A.4 Adaptive High-gain EKF

The AEKF is a recent innovation in the field of non-linear observers. Proposed in [8] as a solution that combines the best features of EKF noise filtering and high-gain EKF perturbation sensitivity, the AEKF is based on the concept of *innovation*, a metric that indicates the deviation between the historical predicted state and the historical measured state. When the innovation becomes arbitrarily high, it triggers an increase in gain that switches the AEKF from unity-gain mode to high-gain mode.

Furthermore, the AEKF is a mathematically proved observer. Its stability is grounded by theory, as opposed to the simple EKF. This can have important consequences for safety, if an observer is used for closed-loop control purposes.

A.4.1 Observability normal form

(In the case of a multiple-input/multiple-output system, a change of variables is not necessarily unique and thus we talk of *normal form* instead of *canonical form*.)

High-gain observers, developed within the framework of [21], have been proven to achieve global exponential convergence. (Recall that the EKF has only been proven to converge locally, i.e. when the estimated state is in a neighborhood ε of the true state.) This exponential convergence property requires that the model be in a very specific form, called *observability normal form*, which characterizes the observability of the system. The general multiple-output normal form is described in Sec. A.4.3.

Note 20 *At risk of stating the obvious, before building any kind of observer it is necessary to first verify that the system is observable. Obviously, it is impossible to design an observer if the system itself is not observable (no matter the change of coordinates, observability is an intrinsic property). However, all methods by which the model is brought into the normal form are by definition methods that show observability. Therefore, we can verify observability and compute the change of coordinates in one single action.*

A.4.2 Continuous-discrete AEKF

The multiple output CD-AEKF [7] is repeated here for completeness.

We write the continuous-discrete time normal form as

$$\left\{ \begin{array}{rcl} \dot{\mathbf{z}} & = & \mathbf{A}\mathbf{z} + b(\mathbf{z}) \\ \mathbf{y}_{t_k} & = & \mathbf{C}_k \mathbf{z} \end{array} \right. \tag{A.33}$$

For a system with n_x states and n_y measurements:

Prediction:

$$\begin{array}{rcl} \hat{\mathbf{z}}_k^- & = & \hat{\mathbf{z}}_{k-1} + \displaystyle\int_{t_{k-1}}^{t_k} \left[\mathbf{A}\hat{\mathbf{z}} + b(\hat{\mathbf{z}})\right] dt \\ \hat{\mathbf{P}}_k^- & = & \hat{\mathbf{P}}_{k-1} + \displaystyle\int_{t_{k-1}}^{t_k} \left[(\mathbf{A} + b^*(\hat{\mathbf{z}}))\,\hat{\mathbf{P}} + \hat{\mathbf{P}}\,(\mathbf{A} + b^*(\hat{\mathbf{z}}))^T + \mathbf{Q}_G \right] dt \end{array} \tag{A.34}$$

where $\hat{\mathbf{P}}|_{t_{k-1}} = \hat{\mathbf{P}}_{k-1}$ and $\hat{\mathbf{z}}|_{t_{k-1}} = \hat{\mathbf{z}}_{k-1}$

Correction:

$$\begin{array}{rcl} \mathbf{K}_k & = & \hat{\mathbf{P}}_k^- \mathbf{C}_k^T (\mathbf{C}_k \hat{\mathbf{P}}_k^- \mathbf{C}_k^T + \mathbf{R}_G(k))^{-1} \\ \hat{\mathbf{z}}_k & = & \hat{\mathbf{z}}_k^- + \mathbf{K}_k(\mathbf{y}_{t_k} - C(\hat{\mathbf{z}}_k^-)) \\ \hat{\mathbf{P}}_k & = & (\mathbf{I} - \mathbf{K}_k \mathbf{C}_k)\hat{\mathbf{P}}_k^- \\ \mathcal{I}_{d,k} & = & \sum_{i=m+1}^{k} \Delta t_i \|\gamma(\mathbf{y}_i - \widetilde{\mathbf{y}}_i)\|^2_{\mathbb{R}^{n_y}} \\ G_k & = & \mathcal{F}(G_{k-1}, \mathcal{I}_{d,k}) \end{array} \tag{A.35}$$

where

1. $\mathbf{Q}_G, \mathbf{R}_G(k)$ are the high-gain modified state noise covariance matrix and measurement noise covariance matrix, both as described in Sec. A.4.4.

2. $\mathcal{I}_{d,k}$ is the *innovation* and $\mathcal{F}$ is the *adaptation*, both explained in Sec. A.4.4.

3. $b^*(\hat{\mathbf{z}})$ is the Jacobian matrix $(\partial b(\mathbf{z})/\partial \mathbf{z})\,|_{\hat{\mathbf{z}}}$

4. γ is a diagonal matrix of normalizing constants, as explained in Sec. A.4.4

5. $G_k \geq 1$ is the high-gain parameter, explained in Sec. A.4.4. In addition, $G_0 = 1$

6. $\mathbf{z}_0$ and $\mathbf{P}_0$ are initial guesses for state and state covariance, and furthermore $\mathbf{P}_0$ is symmetric, definite, positive matrix.

A.4.3 AEKF Theory

Since, unlike the single-output case, there is no unique normal form for multiple output models[21] the observer must be designed accordingly. This representation is described immediately below.

Definition 1 *As earlier,* $\mathbf{z} \in \mathbb{R}^{n_x}$ *denotes the state vector.* $\mathbf{z}$ *is divided into sub-vectors denoted* $\mathbf{z}_i$ for $i \in \{1, \dots, p\}$*, where each one exists in a compact subset of* $\mathbb{R}^{n_{x,i}}$*. Hence the dimension of* $\mathbf{z}_i$ *is* $n_{x,i}$ *with* $\sum_{i=1}^{p} n_{x,i} = n_x$*. The dynamics of a given sub-vector* $\mathbf{z}_i = (z_{i,1} \quad \dots \quad z_{i,n_{x,i}})^T$ *are of the form:*

$$\dot{\mathbf{z}}_i = \begin{pmatrix} 0 & \alpha_{i,2} & 0 & \dots & 0 \\ & & \alpha_{i,3} & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ & & & & \alpha_{i,n_{x,i}} \\ 0 & & \dots & & 0 \end{pmatrix} \mathbf{z}_i + \begin{pmatrix} b_{i,1}(z_{i,1}) \\ b_{i,2}(z_{i,1} \ , \quad z_{i,2}) \\ \vdots \\ b_{i,n_{x,i}-1}(z_{i,1} \ , \ \dots \ , \ z_{i,n_{x,i}-1}) \\ b_{i,n_{x,i}}(\mathbf{z}) \end{pmatrix}$$

While the other $b_{i,j}(\cdot)$ *do not, the sub-vector field components of the form* $b_{i,n_{x,i}}(\cdot)$ **do** *have the right to depend on the full state.*

The full system is of the form eq. (A.33). In matrix $\mathbf{C}_k$*, the row correponding to the* i^{th} *output is composed of zeros; except for index* $(n_{x,1} + n_{x,2} + \dots + n_{x,i-1} + 1)$*, which is equal to* $\alpha_{i,1}$

The hypotheses are:

1. *Although omitted,* $\alpha_{i,j}$ *can be time dependent. All the* $\alpha_{i,j}(t)$ *functions are bounded and cannot vanish (which would lead to a loss of observability).*

2. *The vector field $b(\mathbf{z})$ is supposed compactly supported and Lipschitz.*

For such systems we can design an AEKF as in Sec. A.4.2 which has the following convergence property:

Theorem 1 *For any time $T^* > 0$ and any $\epsilon^* > 0$, there exist*

1. $0 < d < T^*$ *that define an innovation*
2. $G_0 > 1$ *and* $G_{max} > G_0$
3. *a function* $G_k = \mathcal{F}(G_{k-1}, \mathcal{I}_d)$ *such that* $G_k \in [1, G_{max}]$

such that for any time $t \geq T^*$:

$$\|\mathbf{z}(t) - \hat{\mathbf{z}}(t)\|^2_{\mathbb{R}^{n_x}} \leq \epsilon^* e^{-a(t-T^*)}$$

where $a > 0$ is a constant independent from ϵ^*

A.4.4 AEKF considerations

Understanding the AEKF requires introducing certain concepts. We start with the high-gain matrices $\mathbf{Q}_G$ and $\mathbf{R}_G$. For reasons of Lipschitz constants (explained in [7], Chapter 5):

$$\mathbf{Q}_G = G\Delta_G \mathbf{Q}_c \Delta_G \tag{A.36}$$

where $\mathbf{Q}_c$ is a symmetric positive semi-definite matrix, and Δ_G is a diagonal matrix with elements determined by

1. The normal form leads to a block-wise diagonal matrix $\mathbf{A}$, as in eq. (1.22). The matrix $\mathbf{A}$ is split into p square submatrices. Define n_x^* as the width of the largest submatrix.
2. In Δ_G, for each square block $\Delta_{G,i}$ (of width $n_{x,i}$) corresponding to one of the p blocks in $\mathbf{A}$ (also of width $n_{x,i}$), the elements are determined by

$$\text{for k=1...}n_{x,i} \quad \Delta_{G,i}(k,k) = G^{n_x^* - n_{x,i} + k - 1}$$

For similar reasons as for $\mathbf{Q}_G$

$$\mathbf{R}_G(k) = G^{-1}\delta_G \mathbf{R}_k \delta_G \tag{A.37}$$

where $\mathbf{R}_k$ is a symmetric positive semi-definite matrix, and δ_G is a diagonal matrix with elements determined by taking the first element of each subdiagonal of Δ_G

$$\text{for i=1...}p \quad \delta_G(i,i) = \Delta_{G,i}(1,1)$$

The application of $\mathbf{R}_G(k)$ to an asynchronous measurement model is detailed in **Data fusion**, Sec. A.5.

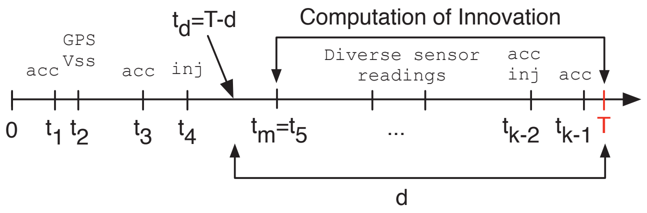


Figure A.1: INNOVATION TIMESCALE

A.4.4.1 Understanding "innovation"

The innovation at time t is calculated in a moving window from $(t-d)$ to t, and as such d is the characterizing parameter. Innovation for a variable-step asynchronous continuous-discrete system is written as

$$\mathcal{I}_{d,k}(t) \equiv \sum_{i=m+1}^{k} \Delta t_i ||\gamma(\mathbf{y}_i - \widetilde{\mathbf{y}}_i)||^2_{\mathbb{R}^{n_y}} \tag{A.38}$$

where

$$\Delta t_i = t_i - t_{i-1} \tag{A.39}$$

and

$$m = \min_{t_m \in [t_0, t_k]} (t_m \geq t - d) \tag{A.40}$$

and γ is a normalizing matrix.

$$\gamma = \mathrm{diag}(\gamma_1, \gamma_2, \ldots, \gamma_{n_y}) \tag{A.41}$$

and $\widetilde{\mathbf{y}}_i$ is found by integrating the IVP presented in eq. (A.33)

$$\left\{ \begin{array}{rcl} \dot{\mathbf{z}} & = & \mathbf{A}\mathbf{z} + b(\mathbf{z}) \\ \widetilde{\mathbf{y}}_i & = & \mathbf{C}_i \mathbf{z}(t_i) \\ & \text{and} & \\ \mathbf{z}(t_m) & = & \hat{\mathbf{z}}(t_m) \end{array} \right. \tag{A.42}$$

In other words, the innovation is the sum of the length of the time interval, Δt_i, times the difference between the measured state and the innovative predicted state, $\gamma||\mathbf{y} - \widetilde{\mathbf{y}}||^2_{\mathbb{R}^{n_y}}$, of **each** time interval that happened after $t-d$ seconds. See Fig. A.1. (This point is justified by a Lemma in [7])

Innovation can be thought of as an experiment to see what happens if the observer correction is turned off at time $t-d$ and the system is allowed to continue naturally. The amount of innovation is, roughly speaking, proportional to the square of the norm of the error between the observed trajectory and the non-observed trajectory.

A.4.4.2 Understanding "adaptation"

In a classical High-Gain Kalman Filter, the high-gain parameter, G is fixed. In an AEKF, a function called *adaptation* varies the value of G on the interval $[1, G_{max}]$. $G = 1$ and $G = G_{max}$ correspond to the unity-gain mode, otherwise known as the classical EKF, and the high-gain EKF mode, respectively. The *adaptation*, $\mathcal{F}$, is driven by the estimatation quality measurement, the *innovation*.

In the CD-AEKF, the adaptation can be used as a threshold-based switch which incrementally steps G up and down between 1 and its maximum value. As explained in [7], an incremental function is suitable in the continuous-discrete case.

$\mathcal{F}$ is used in order to limit the overshoot. It is known that for high-gain filters, there is a *peak phenomenon*[21] when the adaptation switches. This phenomenon creates a perturbation to the covariance matrix P, which resembles a poorly initialized P_0.

Note 21 *The model described by (1.16) is driven ONLY by perturbations. Thus the high-gain triggering frequency will be high, and as such the goal of the adaptation is to reduce to a maximum the peak phenomena.*

An example adaptation function $\mathcal{F}(G_{k-1}, \mathcal{I}_{d,k})$ would be

$$\begin{array}{llll} \text{If } (\mathcal{I}_{d,k} > \text{upper threshold}) & \text{AND} \quad (G < G_{max}) & \Rightarrow & (G = G + 1) \\ \text{If } (\mathcal{I}_{d,k} < \text{lower threshold}) & \text{AND} \quad (G > 1) & \Rightarrow & (G = G - 1) \end{array}$$

Preliminary results with the AEKF are promising as the high-gain component allows the observer to converge much more quickly when there is an unmodeled perturbation, such as a sudden braking or acceleration.

A.5 Data fusion

One of the particular problems that can occur in filtering is treating asynchronous and redundant data. In a multiple-output system, diverse sensor measurements are combined together in order to form the state estimation. Some sensor units might give multiple sensor readings, such as a GPS that in addition to the desired latitude and longitude outputs gives altitude, speed, bearing, etc...

Furthermore, in a discrete-time sampled system, it is often the case that various sensor data arrives asynchronously, and this asynchronous data need not necessarily have a constant sampling rate.

These situations are both handled well by parametric model-based filters (such as the Kalman Filters presented above).

A.5.1 Redundancy

In general, sensors are chosen such that each sensor gives different information from the others, however this is not always the case. When more than one sensor measures the same data, this data becomes *redundant*. One way to handle the redundant data measurements is to simply ignore them, but this is less than ideal, especially as the redundant case is easy to treat.

Each redundant data source acts like a new output in the measurement model $H(\mathbf{x}, \mathbf{u})$. Each source has an associated covariance value (in the $\mathbf{R}$ matrix) that reflects the confidence in each particular measurement, exactly as would be done if the data were not redundant.

For instance, take the case of a system receiving altitude information from both a GPS and a DEM (Digital Elevation Map). A GPS is not very accurate in altitude terms, and tends to get a "stuck" on a particular bias, so the measurement covariance parameter for the GPS would be tuned to a relatively high value. On the other hand, a DEM is often accurate to sub-cm resolution, so it would have a relatively low covariance value.

The advantage of data redundancy is that if one sensor or another is lost, the model still retains its good qualities, and, more importantly, cannot lose observability. Again, drawing on the above example, there are times when a GPS cuts out— urban canyons, tunnels, etc..., and times when a DEM does not have any relevant data. The combination of the two "sensors" gives us the best possible results.

Redundancy can occur in any time domain. Furthermore, with certain sensors the information is complementary, e.g. digital map data is more precise, but GPS data has a finer grid resolution.

A.5.2 Asynchronous data

Asynchronous measurements, by their very nature, can only occur in discrete update time domains, e.g. continuous-discrete and discrete-discrete. There have been many papers written about data fusion of asynchronous signals, some which recommend interpolating lower frequency signals in order to form a sort of "oversampling", or resampling higher frequency signals in order to form a sort of "undersampling". The author recommends against this approach, as not only can it introduce unwanted effects, it completely breaks down if some of the measurements are not only asynchronous but also do not have a consistent sampling time.

The asynchronous case is not as straightforward as the redundancy case. Some measurements can arrive as a function of time, some as a function of a state variable, and some as externally triggered events. (See Note 18 for a discussion of asynchronous system observability.)

Such a set of measurement data necessitates an approach with a variable time step, where a new update/correction is triggered at each moment there is a new data value. This implies a constantly changing measurement matrix $\mathbf{H}_k$.

As shown in Algorithm 1, we integrate the differential system equations until we encounter an event at time t_k. At this moment, we start by determining what measurements are present and then stack them into the output vector, $\mathbf{y}_k$. Then we assemble the measurement Jacobian matrix $\mathbf{H}_k$ in the same fashion, again by stacking the individual row vectors associated with the measurements.

Once we have assembled $\mathbf{H}_k$, we can proceed as normal with the discrete update/correction of the observer.

Additionally, the measurement covariance matrix $\mathbf{R}_k$ must change as a function of the time step and of the available data. It is always of dimension $\mathbb{R}^{n_y \times n_y}$

Note 22 *The above fusion techniques are equally valid for the discrete-discrete approach. However, if the DD-EKF had been used instead of the CD-EKF,* $\mathbf{Q}$ *would have varied at each iteration as a function of time since the covariance of a gaussian noise of covariance* q *after a time* Δt *is* $q\Delta t$*. I.e., the longer the observer waits without any measurement data, the higher the uncertainty, which is represented by a growing* q*. Thus in the discrete-discrete case:*

$$\mathbf{Q}_d = \mathbf{Q}_c \Delta t_{t_k} \tag{A.43}$$

Appendix B

Control

Contents

Dynamic optimization problems can be solved with different approaches, depending on the particular nature of the problem. Globally, optimization problems can be tackled along three major lines, presented in descending order of computational costs:

- Dynamic programming: uses the principle of optimality of subtrajectories in order to recursively compute a control for any $\mathbf{x_0}$ and any t. The solution for a continuous problem stems from resolving the Hamilton-Jacobi-Bellman (HJB) equation, a partial differential equation in state space.
- Direct methods: referred to as "first discretize, then optimize". These methods require discretizing the commands u along multiple subsections of time, and then seeking the optimal trajectory across each subspace. The *direct shooting* and *direct multiple shooting* methods are examples of this approach.
- Indirect methods: referred to as "first optimize, then discretize". They have the advantage of being able to present a complete optimal synthesis of the problem, dramatically reducing computational costs. The *shooting* and *multiple shooting* methods are examples of this approach.

B.1 Controllers and controllability

B.1.1 Controllers

Roughly speaking, the control is the vector $\mathbf{u}$ that brings the state from $\mathbf{x}(0)$ to $\mathbf{x}(t_f)$ in finite time $t_f < T$.

Thus, the controller is the mathematical algorithm that calculates the control $\mathbf{u}$.

B.1.2 Controllability

Controllability, like observability, is an important property for any system we wish to completely control. Controllability refers to the ability to act on all the variables required to control the system. The world is full of systems that we can observe, but not act upon: a bouncing ball, a paper airplane, a waterfall, etc... What is missing in all these cases is that we have no external actor, and thus they are not *controllable*.

Controllability is a mathematical way of expressing the idea that if a system starts at A then there is a sequence of commands, out of all the possible commands, that will bring the system to state B.

In more rigorous terms, a system is said to be controllable if for all vector pairs $(\mathbf{x}_a, \mathbf{x}_b)$ in $\mathbb{R}^n$ there exists a finite time T and a command $\mathbf{u}$ defined on $[0, T]$ such that, when the command $\mathbf{u}$ is applied, the solution $\mathbf{x}(t)$ starting from $\mathbf{x}(0) = \mathbf{x}_a$ gives $\mathbf{x}(T) = \mathbf{x}_b$.

Linear controllability: The Kalman rank condition states that a linear system of the form

$$\left\{ \begin{array}{rcl} \dfrac{d\mathbf{x}}{dt} & = & \mathbf{Ax} + \mathbf{Bu} \\ \mathbf{y}(t) & = & \mathbf{Cx} + \mathbf{Du} \end{array} \right.$$

is controllable if and only if the *controllability matrix*

$$\left(\begin{array}{cccc} \mathbf{B} & \mathbf{AB} & \dots & \mathbf{A}^{n-1}\mathbf{B} \end{array} \right) \tag{B.1}$$

is of rank n_x.[19]

Non-linear controllability: In the nonlinear case, it can happen that the system is controllable at certain points, but not at others.

We can speak about a larger condition of global controllability. For an affine system of the form

$$\dot{\mathbf{x}} = f(\mathbf{x}) + \sum_{i=1}^{n_u} g_i(\mathbf{x}) u_i$$

where g_i is a vector valued function.

We have a test for local controllability, but similar to the observability test in Sec. A.1.2, we must use Lie algebra to apply it.

First we define the Lie bracket operator (The derivation and purpose of Lie brackets is far outside the scope of this PhD. If the reader would like further information, s/he is encouraged to read Boothby's excellent description.[11]):

$$[f, g] \equiv \frac{\partial g}{\partial \mathbf{x}} f - \frac{\partial f}{\partial \mathbf{x}} g$$

and higher order Lie brackets:

$$\begin{array}{rcl} (ad_f^1, g) & \equiv & [f, g] \\ (ad_f^2, g) & \equiv & [f, [f, g]] \\ & \vdots & \\ (ad_f^k, g) & \equiv & [f, (ad_f^{k-1}, g)] \end{array}$$

Note: *ad* is read "adjoint".

The system is locally controllable about a point if and only if the controllability matrix

$$\left[g_1 \dots g_{n_u} \quad [ad_{g_i}^1, g_j] \dots [ad_{g_i}^{n_u}, g_j] \quad \dots \quad [ad_f^1, g_i] \dots [ad_f^{n_x-1}, g_i] \right] \tag{B.2}$$

has rank n_x.

B.2 Optimal control

We might ask ourselves the question, 'If a system is controllable, is there a such thing as a "perfect" control?'

First, we have to define what a "perfect" control is. One popular choice is to define "perfect" as "minimizing a cost". (Or "maximizing a benefit". The opposite of one is mathematically identical to the other.)

Thus, among all the possible solutions that bring $\mathbf{x}(0)$ to $\mathbf{x}(T)$, we look to find a trajectory $\mathbf{u}^*$ that minimizes a cost, $J(T, \mathbf{x}, \mathbf{u})$. Such a trajectory, if it exists, is called an "optimal control" for this cost.

The existence of an optimal trajectory is dependent on the system regularity and the cost function. It is possible that an optimal control does not exist in the class of considered controls, but that it exists in a larger space of controls.[49]

B.2.1 Concept of time in optimal control

In [5] Bellman showed that the natural approach to optimality problems is to work backwards in time from the solution. Call this *time to go.*

Time to go makes a certain amount of sense when it is considered that the objective is known, but not necessarily the initial point. While we can easily calculate the finite number of ways to optimally approach the final point, there are an infinite number of ways to leave the initial point, and we have no way to immediately distinguish those that are possibly optimal from those that are not.

Thus, we talk about integrating from the final target $\mathbf{x}(T)$ to the initial point $\mathbf{x}(0)$.

One confusing facet of inverse time is how to refer to it. If one talks of time T in an inverse time problem, does this refer to time T with respect to forward or backward time? For this reason, we prefer the phrase "time to go", which, when scrupulously used, removes all ambiguity. We represent "time to go" by s and thus

$$\begin{array}{ccccccc} t & = & 0 & \mapsto & s & = & T \\ t & = & T & \mapsto & s & = & 0 \end{array}$$

Inversing the dynamic equations is straightforward. It suffices to take the negative of the dynamic equation

$$\dot{\mathbf{x}}^* = -\dot{\mathbf{x}} \tag{B.3}$$

B.2.2 Linear quadratic control

When the system is linear, or is linearized about an operation point, a special general solution can be found in the case where the cost is of the form

$$J = \frac{1}{2}\int_0^{t_f} (\mathbf{x}^T\mathbf{Q}_{lqr}\mathbf{x} + \mathbf{u}^T\mathbf{R}_{lqr}\mathbf{u})dt \tag{B.4}$$

where $\mathbf{Q}_{lqr}$ and $\mathbf{R}_{lqr}$ are positive semi-definite and positive definite constant matrices, respectively.

Note 23 *Since the solution has already been found for all linear systems with the cost as described in eq. (B.4), the principle of* time to go *is not used here.*

B.2.2.1 Continuous LQR

The optimal control is then

$$\mathbf{u}^* = -\mathbf{K}_{lqr}\mathbf{x} \tag{B.5}$$

where

$$\mathbf{K}_{lqr} \quad = \quad \mathbf{R}_{lqr}^{-1}\mathbf{B}^T\mathbf{S}_{lqr} \tag{B.6}$$

and where $\mathbf{S}_{lqr}$ is the convergence of the differential Riccati equation

$$\dot{\mathbf{S}}_{lqr} \quad = \quad -\mathbf{S}_{lqr}\mathbf{A} - \mathbf{A}^T\mathbf{S}_{lqr} + \mathbf{S}_{lqr}\mathbf{B}\mathbf{R}_{lqr}^{-1}\mathbf{B}^T\mathbf{S}_{lqr}\mathbf{Q}_{lqr} \tag{B.7}$$

B.2.2.2 Discrete LQR

Perhaps a more practical use for the LQR is optimal control in discrete-time. We present here the slightly modified approach for the discrete LQR.

$$\begin{array}{lll} \mathbf{S}_0 & = & \mathbf{Q}_{dlqr} + \mathbf{A}_k^T(\mathbf{B}_k\mathbf{R}_{dlqr}^{-1}\mathbf{B}_k^T)^{-1}\mathbf{A}_k \\ \mathbf{S}_k & = & \mathbf{Q}_{dlqr} + \mathbf{A}_k^T(\mathbf{S}_{k-1}^{-1} + \mathbf{B}_k\mathbf{R}_{dlqr}^{-1}\mathbf{B}_k^T)^{-1}\mathbf{A}_k \\ \mathbf{K}_{dlqr,k} & = & (\mathbf{B}_k^T\mathbf{S}_k\mathbf{B}_k + \mathbf{R}_{dlqr})(\mathbf{B}_k^T)^{-1}\mathbf{S}_k\mathbf{A}_k \end{array} \tag{B.8}$$

(B.9)

where $\mathbf{Q}_{lqr}$ and $\mathbf{R}_{lqr}$ are as in the above case for the continuous control.

The control is as before:

$$\mathbf{u}^* = -\mathbf{K}_{dlqr,k}\mathbf{x}_k$$

The LQR is particularly interesting for basic control, as it provides a first point of entry to optimal control. If the model is known, then the Jacobian can be derived (or calculated) and as a consequence it is computationally cheap to create an optimal control that has sufficient degrees of freedom for minimizing the inputs or the state values.

The LQR control is easy to apply, and far outperforms rudimentary controllers such as PIDs. If, however, the cost that we seek to minimize cannot be effectively approximated by eq. (B.4), we are forced to explore higher complexity optimization solutions.

B.2.3 Dynamic Programming

Dynamic programming is a design technique similar to “divide-and-conquer”. Divide-and-conquer algorithms partition the problem into independent subproblems, solve the subproblems recursively, and then combine their solutions in order to solve the original problem.

Thus, dynamic programming is an optimality approach where a complex optimization problem is broken down into a sequence of decision steps over time. It uses the princple of optimality of subarcs in order to recursively compute feedback control for all time t and all $\mathbf{x}_0$

In the continuous-time case, this leads to the Hamilton-Jacobi-Bellman (HJB) equation.

$$\frac{\partial S}{\partial t} + H_1\left(\mathbf{x}, \frac{\partial S}{\partial \mathbf{x}}\right) = 0$$

Dynamic programming, while “optimal” in the sense that it eventually finds an optimal solution, is resource intensive. It is also not well adapted to high accuracy, and it is possible that it miss singular optimal paths. The allure of dynamic programming is its programming simplicity for complex models with arbitrary initial states.

B.2.4 Non-linear Programming

Non-linear programming (NLP) reformulates the optimal control problem into a finite dimensional model of the form

$$\min_{\mathbf{w}} a(\mathbf{w}) \text{ subject to } b(\mathbf{w}) = 0,\ c(\mathbf{w}) \geq 0 \tag{B.10}$$

where the finite dimensional vector $\mathbf{w}$ represents the optimization degrees of freedom, and with the differentiable scalar function a and vector valued functions b, c

NLP, while widely used in industry, does not give any feel to the problem dynamics. Moreover, while the computation technique is more graceful than dynamic programming, it still is a process that requires intense usage of computer memory and processor. Moreover, there is no way with NLP to be sure that the optimal solution as given by the solver is the true optimal solution.

B.2.5 Pontryagin's Maximum Principle

Note 24 *The PMP is often used as* Pontryagin's Minimum Principle*, which simply inverses Pontryagin's original "benefit function" to be maximized into a "cost function" to be minimized. The mathematics are the same, only the sign of the cost J is different.*

PMP states a necessary condition that must hold on the entire length of an optimum trajectory, starting from a *fixed* initial state, $\mathbf{x}(0)$. [49]

Consider the system

$$\dot{\mathbf{x}} \quad = \quad f(\mathbf{x}(t), \mathbf{u}(t)) \tag{B.11}$$

with associated cost function

$$J(\mathbf{x}, \mathbf{u}, t) \quad = \quad K(\mathbf{x}(T)) + \int_0^T L(\mathbf{x}(t), \mathbf{u}(t), t)dt \tag{B.12}$$

where L the instantaneous cost, and K the terminal cost.

We can define the Hamiltonian as

$$H(\mathbf{x}, \mathbf{u}, \lambda) \quad = \quad \lambda^T f(\mathbf{x}, \mathbf{u}) + \lambda^0 L(\mathbf{x}, \mathbf{u}, t) \tag{B.13}$$

where λ is the *costate vector*, also known as the *adjoint vector*, and $\lambda^0 \leq 0$ (since this is Pontryagin's Maximum Principle. In the case of the Minimum Principle, $\lambda^0 \geq 0$) such that the vector $< \lambda^0, \lambda > \neq \vec{0}$

Note 25 *λ^0 is generally normalized to -1 by dividing the costate vector λ by λ^0. However, in the case where $\lambda^0 = 0$ this obviously cannot be done. This case arises when the optimal trajectory is independent of the cost, which can occurs, for instance, when there is only one possible trajectory.*

Suppose that $\mathbf{x}^*(t), \mathbf{u}^*(t), \lambda^*(t)$ represent the optimal state, optimal control, and optimal costate trajectories, respectively. I.e. $\mathbf{u}^*(t)$ is the control that that brings the state from an initial state $\mathbf{x}(0)$ to a final state $\mathbf{x}(T)$ and minimizes the cost J, and $\mathbf{x}(t)$ is the associated state trajectory. Then there exists a costate trajectory $\lambda(t)$ such that

$$\begin{aligned} \dot{\mathbf{x}} &= \frac{\partial H}{\partial \lambda} \\ \dot{\lambda} &= -\frac{\partial H}{\partial \mathbf{x}} \end{aligned} \tag{B.14}$$

and for almost all t, $0 \leq t \leq T$

$$H(\mathbf{x}^*, \mathbf{u}, \lambda) \quad \leq \quad H(\mathbf{x}^*, \mathbf{u}^*, \lambda) \tag{B.15}$$

that is to say that the optimal control $\mathbf{u}^*$ is the control that maximizes H amongst all the possible controls $\mathbf{u}$ that can be applied to bring $\mathbf{x}(0)$ to $\mathbf{x}(T)$.

By using the system and costate vector, we can define the *augmented system* $\mathbf{z}$:

$$\dot{\mathbf{z}} = \begin{cases} \dot{\mathbf{x}} \\ \dot{\lambda} \end{cases} \tag{B.16}$$

The PMP is particularly adept at solving minimal time problems (when the cost function depends only on time), but can also be solved for fixed time, or can be solved independent of time. If the fixed time limit is set far enough in the future, this will yield identical results to the solving independently of time.

PMP can give a set of global rules that can be expressed in analytic terms, called an *optimal synthesis*. Such an optimal synthesis can easily be programmed into a microprocessor, greatly reducing the real-world resources necessary to compute the optimal control. These rules read like an instruction manual— they dictate what control should be applied and when, much like a numerical recipe. See Appendix G for a worked example.

B.2.5.1 Transversality conditions

Transversality on the costate vector If not all states at the final state $\mathbf{x}(T)$ are fixed, then the costate vector can be such that the transversality conditions at one or both endpoints satisfy

$$\lambda_0^T \mathbf{x}(0)\sigma \quad = \quad 0 \tag{B.17}$$

and

$$\left(\lambda_T - \lambda\frac{\partial K}{\partial t}\right)^T \mathbf{x}(T)\sigma \quad = \quad 0 \tag{B.18}$$

for all σ such that $x + \epsilon\sigma$ is within $o(\epsilon)$ of the termination point of an optimal trajectory for all sufficiently small $\epsilon > 0$.

In other words, these two conditions verify that the costate vector is perpendicular to the state vector at the endpoints.

Note 26 *If there is no value-added cost K, this condition has an immediate consequence when a state $\mathbf{x}_j$is not present in the optimal target. The transversality condition gives $\lambda_j = 0$.*

Note 27 *If the target $\mathbf{x} = \vec{0}$, then we can only take $\sigma = 0$ and the transversality conditions do not add any additional information to the problem.*

Transversality on the Hamiltonian If the final time for arriving at the final state $\mathbf{x}(T)$ is not fixed, there is a transversality condition at $t = T$

$$H(\mathbf{x}^*, \mathbf{u}^*, \lambda^*) \quad = \quad -\lambda^0 \frac{\partial K}{\partial t} \tag{B.19}$$

B.2.5.2 Limits on the control

Limits on the control pose no particular problem. Indeed, this sort of control limit oftentimes presents itself in the form of *bang-bang* controls, controls that are applied only at the upper or lower bound of the possible control.

B.2.5.3 Limits on the state

Limits on the state are much more complicated to deal with. While there are several versions of the Maximum Principle that allow for state limits– see for instance [16], the theory and application is sufficiently complicated that satisfactory results can be more easily obtained with other methods. The problem with state limits comes from the fact that the costate vector can become discontinuous at the state boundaries.

Oftentimes, a state limit is avoided by the simple nature of the problem. For instance, it is unlikely that a fixed-time energy optimal trajectory for a car trip will push the car to break the speed limit. If it does, the fixed-time horizon can simply be elongated, which will almost certainly find a solution at a slower speed. It is in general easier to examine the PMP optimal solution and modify the input parameters in case of a state limit, than it is to treat the state limits analytically.

Penalization methods: Another method to manipulate the optimal solution so as to impose state limits it to us *penalization methods*. By weighting certain limits, the optimal solution can be forced to obey the state limits. For instance, suppose that we want to impose a limit on the state. We can add a cost of the form

$$\alpha \int_0^T g(\mathbf{x}(t)) dt$$

to the original cost function $J(\mathbf{x}, \mathbf{u}, t)$, where α is the weighting value, and g is a function that is 0 when the state is within its limits, and strictly positive everywhere else. The hope is that by adding this cost the the optimal solution will force the optimal trajectory to stay within the prescribed limits.

The mathematical justification for this approach can be found in [49].

B.3 Numerical methods for optimal control

Choosing the appropriate method is based on questions of numerical stability, computation time, and desired accuracy. While this dissertation focuses on Pontryagin, we also present here strategies for Non-Linear Programming and Dynamic Programming problems.

B.3.1 Indirect methods

Indirect methods are used for Pontryagin problems where the costate vector λ is also sought. Their advantages are a higher level of precision, converge toward the true optimal solution without risk of getting stuck at local minima, and are memory efficient. Lastly, the multiple shooting method is parallelizable, for increased computational speed.

B.3.1.1 Shooting method

The shooting method is a way of reducing a *boundary value problem* (BVP) to an *initial value problem* (IVP). Its name is inspired from the problem of shooting at a known target. It is not known in advance how high to aim the shot, so one simply aims, fires, and sees where the shot lands. If it lands short, the next is aimed higher. Conversely, if it lands far, the next is aimed lower.

Figure B.1: Shooting method

The principle is to take the first boundary condition as an initial point and then create a second condition stating the gradient of the function at the initial point. The IVP is now completely well-posed and the system can be integrated. If the system arrives at a final value other than the target, the initial gradient condition is adjusted. This loop is iterated until the integrated value converges on the true target to within an acceptable amount of error. The solution to the IVP is now also the solution to the BVP.

In mathematical terms, we have a BVP of the form

$$\left\{\begin{array}{rcl} \dot{\mathbf{z}}(t) & = & f(t, \mathbf{z}(t)) \\ r(\mathbf{z}(0), \mathbf{z}(t_f)) & = & \vec{0} \end{array}\right. \tag{B.20}$$

where $r(x, y)$ stands for "boundary conditions at x and boundary conditions at y", and where

$$\mathbf{z} = \begin{pmatrix} \mathbf{x} \\ \lambda \end{pmatrix}$$

is the *augmented state.*

We transform this into an IVP by supposing $\mathbf{z}_0 = \mathbf{z}(0)$ and

$$\left\{\begin{array}{rcl} \dot{\mathbf{z}}(t) & = & f(t, \mathbf{z}(t)) \\ r(\mathbf{z}_0, \mathbf{z}(t_f, \mathbf{z}_0)) & = & \vec{0} \end{array}\right. \tag{B.21}$$

We now have a classic IVP where we seek $\mathbf{z}_0$ such that the second half of eq. (B.21) holds true. Effectively, we are looking for the roots, and this can be solved by a classical zero-crossing method, for example Newton or bisection.

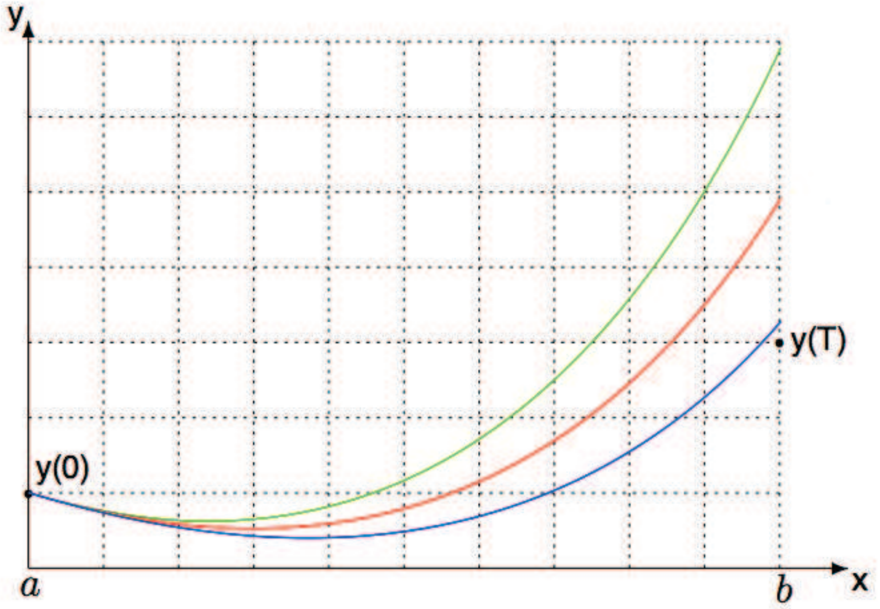


Figure B.2: Shooting method

B.3.1.2 Multiple shooting method

With respect to the simple shooting method, the multiple shooting method breaks the interval $[0, t_f]$ into N intervals $[t_i, t_{i+1}]$ and assigns an initial guess values $\mathbf{z}(t_i)$ to the beginning of each interval. The interest is to improve the stability of the shooting method.

The application of the maximum principle as applied to the optimal control problem $\dot{\mathbf{z}}$ as described in eq. B.20 reduces to

$$\dot{\mathbf{z}}(t) = f(t, \mathbf{z}(t)) = \begin{cases} f_0(t, \mathbf{z}(t)) & \text{if } t_0 \leq t_1 \\ f_1(t, \mathbf{z}(t)) & \text{if } t_1 \leq t_2 \\ \quad\vdots & \\ f_s(t, \mathbf{z}(t)) & \text{if } t_s \leq t_f \end{cases} \tag{B.22}$$

where $t_1, t_2, \ldots, t_s \in [t_0, t_f]$ can be *commutations* of the optimal control and, in the case of state constraints, *contact* with the state frontier. Continuity is be preserved for $\mathbf{z}$ at the commutations, but the costate can jump at the frontiers in state constraint cases.

We begin by defining a function $\Phi(t)$ that is valid on $t = [0, T]$ (a line connecting each state initial value with its final value is a good first guess, but in fact, it can be any Lipschitz continuous function). An initial guess of unknowns $\mathbf{z}_0$ is made. Then system (B.22) is integrated until $||\mathbf{z}(t, \mathbf{z}_0) - \Phi(t)|| > \varepsilon$ for some $\varepsilon > 0$. We designate the time variable at this point as t_1. Now the integration of the system continues from t_1 using $\Phi(t_1)$ as the initial 'guess' for the solution. This process continues until time $t = b$ is reached.

Now the error function $e(s) = ||\mathbf{z}(t_i) - \mathbf{z}_i||$ is formulated where $s = \begin{bmatrix} \mathbf{z}_0 & \mathbf{z}_1 & \ldots & \mathbf{z}_{N-1} \end{bmatrix}^T$, where $\mathbf{z}_i$ is the initial state for the trajectory on the interval $[\mathbf{z}_i, \mathbf{z}_{i+1}$. After this is accomplished, a correction is made to $\mathbf{z}_0$ using a classical zero-crossing method, for example Newton or bisection.

This process is repeated, only now the number of iterations N is fixed, and the function $\Phi(t_i)$ is replaced by $\mathbf{z}_i$. The iterations stop once the global convergence is satisfactory.

Figure B.3 shows the multiple shooting method for the simple case of a one-dimensional system.

Note 28 *The multiple shooting method improves stability by increasing the number of nodes over the simple shooting method. While there are many more unknowns in the multiple shooting method, the method is parallelizable.[12]*

B.3.2 Direct methods

Direct methods are used for Non-Linear Programming models. These solutions are less accurate than the indirect methods, but are oftentimes easier to program, trading computer time for programming time. They also easily handle state constraints and are not very sensitive to the initial value guess.

B.3.2.1 Direct single shooting

Generally, direct shooting methods are the the easiest when a controls problem is first approached. The strong points of direct single shooting are (1) that it can use fully adaptive, error controlled, state-of-the-art ODE or DAE solvers; (2) that it has only a few optimization degrees of freedom even for large ODE or DAE systems; and (3) that only initial guesses for the control degrees of freedom are needed.

The weak points are (1) that the ODE solution $\mathbf{x}(t; \mathbf{q})$, where the arguments after the semi-colon (;) denote the dependence on the interval's initial value and control, can depend

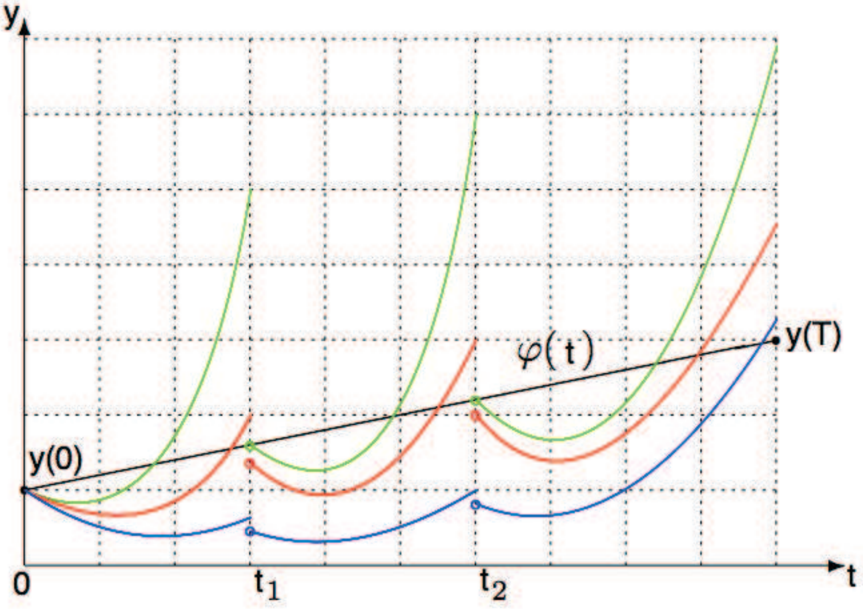


Figure B.3: Multiple shooting method

very nonlinearly on $\mathbf{q}$; (2) that unstable systems are difficult to treat; and (3) it tells us little about the nature of the costate vector.[20]

Due to its simplicity and general efficiency, the direct single shooting approach is often used in engineering applications, e.g. gOPT.[38]

The direct single shooting approach starts by discretizing the controls into N subintervals. We choose grid points on the interval $t_0 = \tau_0 < \tau_1 < \tau_2 < \cdots < \tau_{N-1} < \tau_N = t_f$ and chose the controls $\mathbf{u}$ piecewise constant, $\mathbf{u}(t) = \mathbf{q}_i$ for $t \in [\tau_i, \tau_{i+1}]$ so that $\mathbf{u}(t)$ depends only on the finitely many control parameters $\mathbf{q} = \{\mathbf{q}_0, \mathbf{q}_1, \ldots, \mathbf{q}_{N-1}, T\}$ and can be denoted by $\mathbf{u}(t;\mathbf{q})$, again where the arguments after the semi-colon (;) denote the dependence on the interval's initial value and control. If that problem is for a fixed time T, the last element of $\mathbf{q}$ disappears, as T is no longer an optimization variable.

Path constraints are discretized in order to avoid a semi-infinite problem, for example by requiring that the constraints hold only at grid points τ_i. Thus the optimal problem reduces to the following finite dimensional nonlinear programming problem (NLP):

$$\min_{\mathbf{q}} \int_0^T L(\mathbf{x}(t;\mathbf{q}), \mathbf{u}(t;\mathbf{q}))dt + K(\mathbf{x}(T;\mathbf{q})) \tag{B.23}$$

subject to path and terminal constraints and to

$$\mathbf{s}_0 - \mathbf{x}_0 = 0$$

This problem can be solved by a finite dimensional optimization solver, e.g. sequential quadratic programming (SQP).[35]

B.3.2.2 Direct multiple shooting

The direct multiple shooting, as developed in [6], combines advantages of simultaneous methods with the direct single shooting approach. In direct multiple shooting, we proceed as follows. First, as in the case for direct single shooting, we discretize the controls piecewise on a coarse grid.

$$\mathbf{u}(t) = \mathbf{q}_i \text{ for } t \in [\tau_i, \tau_{i+1}]$$

where the intervals can be as large as in the direct single shooting case. Second, we solve the ODE on each interval $[\tau_i, \tau_{i+1}]$ independently, starting with an artificial initial value $\mathbf{s}_i$

$$\begin{cases} \dot{\mathbf{x}}_i &= f(\mathbf{x}_i(t), \mathbf{q}_i) \text{ for } t \in [\tau_i, \tau_{i+1}] \\ \mathbf{x}_i(\tau_i) &= \mathbf{s}_i \end{cases}$$

By numerical solution of these IVPs, we obtain trajectory arcs $\mathbf{x}_i(t; \mathbf{s}_i, \mathbf{q}_i)$

Simultaneously with the decoupled ODE solution, we also numerically compute the cost function integrals

$$l_i(\mathbf{s}_i, \mathbf{q}_i) = \int_{\tau_i}^{\tau_{i+1}} L(\mathbf{x}_i(\tau_i; \mathbf{s}_i, \mathbf{q}_i), \mathbf{q}_i) dt$$

In order to constrain the artificial degrees of freedom $\mathbf{s}_i$ to physically meaningful values, we impose continuity conditions $\mathbf{s}_{i+1} = \mathbf{x}_i(\tau_{i+1}; \mathbf{s}_i, \mathbf{q}_i)$

Thus we come to the following NLP formulation that is completely equivalent to the direct single shooting NLP, but contains the extra variables $\mathbf{s}_i$ and has a block sparse structure.

$$\min_{\mathbf{s},\mathbf{q}} \sum_{i=0}^{N-1} l_i(\mathbf{s}_i, \mathbf{q}_i) + K(\mathbf{s}_N) \tag{B.24}$$

subject to path and terminal constraints and to

$$\begin{aligned} \mathbf{s}_0 - \mathbf{x}_0 &= 0 \\ \mathbf{s}_{i+1} - \mathbf{x}_i(\tau_{i+1}; \mathbf{s}_i, \mathbf{q}_i) &= 0 \end{aligned}$$

If we summarize all variables as $\mathbf{w} = (\mathbf{s}_0, \mathbf{q}_0, \mathbf{s}_1, \mathbf{q}_1, \ldots, \mathbf{s}_N, \mathbf{q}_N)$ we obtain an NLP which can be exploited in the tailored SQP solution procedure.[6]

B.3.3 Dynamic programming methods

The Hamilton-Jacobi-Bellman equation (HJB) is

$$\frac{\partial S}{\partial t} + H_1\left(\mathbf{x}, \frac{\partial S}{\partial \mathbf{x}}\right) = 0 \tag{B.25}$$

where

$$H_1 = \max_{\mathbf{u} \in U} \lambda^T \dot{\mathbf{x}} - L(\mathbf{x}, \mathbf{u})$$

There are a number of discretization methods. The scheme for finite difference discretization proposed below is one of the simplest, but higher order schemes can be used. We start by writing a finite difference method simple discretization of eq. (B.25). Remark that in order to ensure stability, the discretization scheme must be decentered. Thus, in order to discretize

$$\left\langle \frac{\partial S}{\partial \mathbf{x}}, \dot{\mathbf{x}}(\mathbf{x}, \mathbf{u}) \right\rangle = \sum_{p=1}^{n} \frac{\partial S}{\partial x_p} \dot{x}_p$$

we must discretize $\frac{\partial S}{\partial x_p}$ by a difference divided to the right or left, depending on the sign of $\dot{x}_p$

Let us consider a mesh in space $(x_{\bar{i}})$, where $\bar{i} = (i_1, \ldots, i_n) \in \mathbb{Z}$, which we will suppose to be regular in order to simplify matters, and as consequence a regular discretization of time, (t_j). Let $h = (h_1, \ldots, h_n)$ be the spatial step size, and $k = t_{j+1} - t_j$ be the temporal step size. Let $S_{\bar{i}}$ be the approached value of $S(t_j, \mathbf{x}_{\bar{i}})$. It suffices to approach $\frac{\partial S}{\partial x_p}(\mathbf{x}_{\bar{i}})$ by a difference divided to the left if $\dot{x}_p(x_{\bar{i}}, \mathbf{u}) > 0$, and to the right if $\dot{x}_p(x_{\bar{i}}, \mathbf{u}) < 0$

For $a \in \mathbb{R}$

$$\begin{array}{rclcl} a^+ & = & max(a, 0) & = & \frac{a+|a|}{2} \\ a^- & = & min(a, 0) & = & \frac{a-|a|}{2} \end{array} \tag{B.26}$$

For all $p \in \{1, \ldots, n\}$ we note $\hat{\mathbf{e}}_p = (0, \ldots, 1, \ldots, 0)$ where the 1 is at the p$^{\text{th}}$ place. We thus obtain the explicit scheme

$$0 = \frac{S_{\bar{i},k+1} - S_{\bar{i},k}}{k} + \max_{\mathbf{u} \in U} \left(\sum_{p=1}^{n} \left(\frac{S_{\bar{i},k} - S_{\bar{i}-\hat{\mathbf{e}}_p,k}}{h_p} \dot{x}_p(\mathbf{x}, \mathbf{u})^+ + \frac{S_{\bar{i}+\hat{\mathbf{e}}_p,k} - S_{\bar{i},k}}{h_p} \dot{x}_p(\mathbf{x}, \mathbf{u})^- \right) - L(x_{\bar{i}}, \mathbf{u}) \right) \tag{B.27}$$

Part II

Practice

"Practice is everything."

–Periander, Tyrant of Corinth

"Everything is practice."

–Pelé

Appendix C

Data formats

Contents

C.1 Overview

This appendix details the various data formats used in this dissertation. Specifically

- Quad-coordinates: database specification for quad-tile coordinate system (Sec. C.1.1)
- Configuration: the XML formats used to characterize cars and trips (Sec. C.1.3)
- Observer files manifest: the files generated by the observer program (Sec. C.1.2)
- Logging: the protocol in which the microcontroller code saves the data (Sec. C.1.4)

C.1.1 Quad-tiles and quad-coordinates

While latitude-longitude is the traditional way to geolocate objects, it is not a format that leads itself to easy and rapid search algorithms with fixed-point microcontrollers. A better system is quad-tile coordinates.

Quad-tiles split the world into 4 tiles at each zoom level. Zoom level 1, the entire Earth, has four tiles made by two splits: one along the equator and another along the prime meridian (assuming a Mercator projection where the world is already split in half along the 180° meridian). See Fig. C.1.

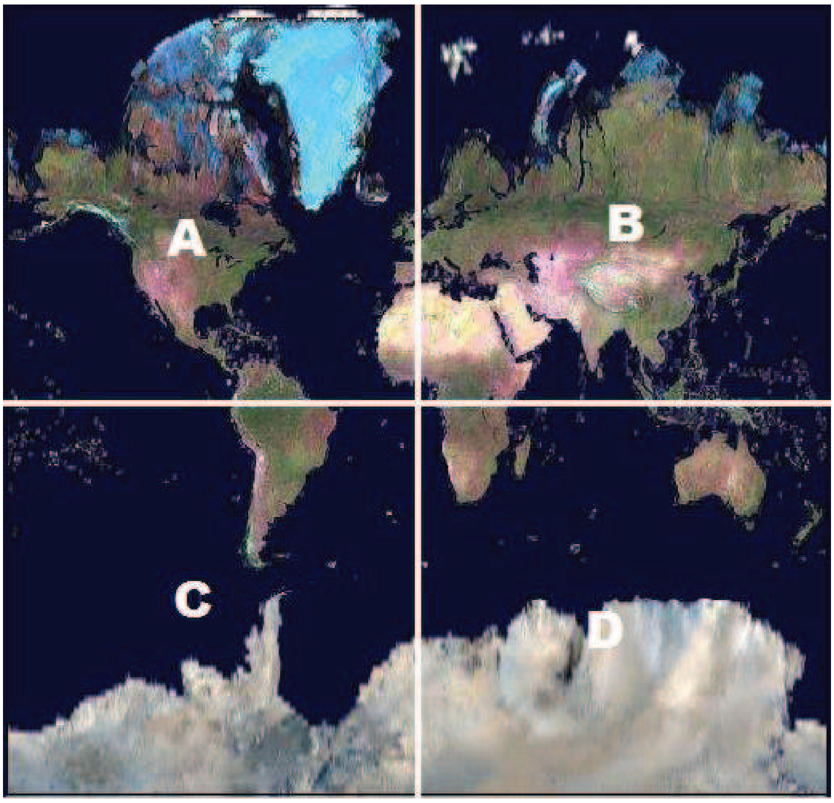


Figure C.1: Quad-tile, zoom level 1

Subsequent tiles can be cut into four smaller tiles, *ad infinium*. The tile information can be represented as a series of pairs of 1's and 0's. In each pair, the first bit, the MSB, represents the left/right side(0=left, 1=right) and the second bit, the LSB, represents the top/bottom (0=top, 1=bottom). Thus each zoom level requires 2 bits.

As such, the entire Australian continent is contained in the quad-tile `11`, all of Greenland is contained in the quad-tile `00`, and the South American continent is split between quad-tiles `01` and `00`. See Fig. C.1.

With a 32-bit integer (zoom level 16), the tiles are $<$ 600m on a vertical side (where obviously the horizontal width will be a function of latitude), which is already a quite small bounding box.

With a 64-bit integer (zoom level 32), the tiles are $<$ 1cm on a side, which gives a very high degree of precision for geolocalization.

Thus quad-tiles lead themselves to geolocalization because these base integer sizes are (luckily) already well-suited to the problem.

Note 29 *The attentive reader will remark that quadcoordinates are incapable of representing an exact point. Furthermore, any point falling on the edge of one of the two bisecting axes at zoom level 1 cannot even be contained in a quad-tile. Thus, both the Equator and North Pole pose representational problems, even if from an engineering perspective this is not relevant.*

C.1.1.1 Quad-tiles database entry format

A system of roads can be seen as an ordered list of interconnected nodes. Nodes can be shared by multiple roads. The database storage and search algorithm used in this dissertation are detailed below:

Node entry: Each node ID is the same as the quad-tile coordinate for that same node. Trading storage space (cheap) for processor speed (expensive), each node also has its associated latitude-longitude coordinates. The number of intersecting ways is provided, and lastly the 32-bit ID for the n ways that intersect the node are given.

	Node entry			
Node ID	Latitude	Longitude	# of Intersecting ways	n Intersecting ways ID
32-bit unsigned	32-bit signed	32-bit signed	8-bit unsigned	n * 32-bit unsigned

Table C.1: Node entry data format

Way entry: Each way ID is represented by a 32-bit unsigned integer. The number of nodes that a way includes is given (in the case of a way that has more than 256 nodes, it can be split into two ways with no loss of information), followed by the node IDs themselves. Lastly, a way is split into a given number of subpoints, which do not necessarily correspond to the nodes making up the way, and the elevation for each node is given.

Way entry				
Way ID	# of nodes	n Nodes	# of Elevation points	n Elevation points
32-bit unsigned	8-bit unsigned	n * 32-bit unsigned	8-bit unsigned	n * 32-bit unsigned

Table C.2: Way entry data format

Note 30 *If it is not immediately obvious why simply using nodes for elevation is unacceptable, consider that most mapping agencies only use nodes when necessary to represent the curvature of a road. The road is considered to be straight between two nodes. If the distance between two nodes is several kilometers (imagine a road in the Australian Outback, or the desert in western America), there will certainly be a number of hills that are hidden by this oversimplification.*

Quad-tile bounding box entry: The quad-tile bounding box entry indicates all the ways that cross the bounding box. This is a useful representation for finding all the ways close to a given point. As in the case for the node entries, the quad-tile bounding box's ID is also its coordinate. The zoom level is also given (by necessity, as the 64-bit field `1100000000000000000000000000000000000000000000000000000000000000` at zoom level 1 would give one fourth the planet, whereas at zoom level 32 it would represent a square roughly one cm on a side almost exactly at the intersection of the equator and the prime meridian.), followed by the number of ways and finally by the IDs of all the contained ways.

Note 31 *There is nor preferred sense to a quad-tile's binary representation. That is to say, a level 1 tile could as well put level one completely at the right, and the left-shift as the zoom increases, or a level 1 zoom could be associated only with the first two left-most bits (otherwise known as MSB) followed by level 2 at the third and fourth bits, etc... The choice depends on the needs of a particular system.*

Quad-tile bounding box entry			
Quad tile coordinate ID	Zoom level	# of contained ways	n Contained ways
64-bit unsigned	8-bit unsigned	8-bit unsigned	n * 32-bit unsigned

Table C.3: Quad-tile bounding box data format

C.1.1.2 Quad-tiles database tree format

Node search algorithm: A simple binary search tree. (See Fig. C.2) The node value leaf data entry is the pointer to the appropriate information in the data table. The center, i.e. starting node, is the very first file entry in the index file. All searches are based on this starting point.

Node binary search tree			
Node ID	Pointer to left leaf	Pointer to right leaf	Pointer to entry in data file
32-bit unsigned	32-bit unsigned	32-bit unsigned	64-bit unsigned

Table C.4: Nodes binary search tree data format

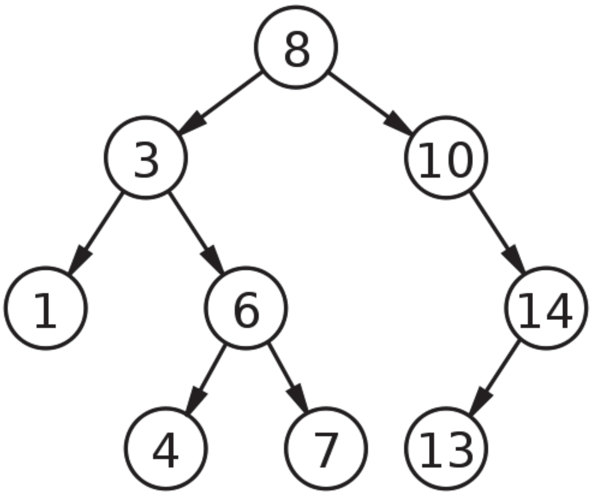


Figure C.2: Binary search tree

Ways search algorithm: A simple binary search tree. The way value data points to the appropriate information in the data table. The center, i.e. starting node, is the very first file entry. All searches are based on this starting point.

Ways binary search tree			
Way ID	Pointer to left leaf	Pointer to right leaf	Pointer to entry in data file
32-bit unsigned	32-bit unsigned	32-bit unsigned	64-bit unsigned

Table C.5: Ways binary search tree data format

Quad-tile bounding box search algorithm: A somewhat more complex quad search tree. The way value data points to the appropriate information in the data table. The center, i.e. starting node, is the very first file entry in the index file. All searches are based on this starting point. In a quadtree, subgroups are divided into four leaves (See Fig. C.3), not two as in a binary search tree. As in the case for quad-tiles, `00` represents the NW quadrant, `01` the SW, `10` the NE, and `11` the SE one.

Quad tile quad search tree					
Quad-tile ID	Pointer to `00` tile	Pointer to `01` tile	Pointer to `10` tile	Pointer to `11` tile	Pointer to entry in data file
64-bit unsigned	32-bit unsigned	32-bit unsigned	32-bit unsigned	32-bit unsigned	64-bit unsigned

Table C.6: Quad-tile bounding box quad search tree data format

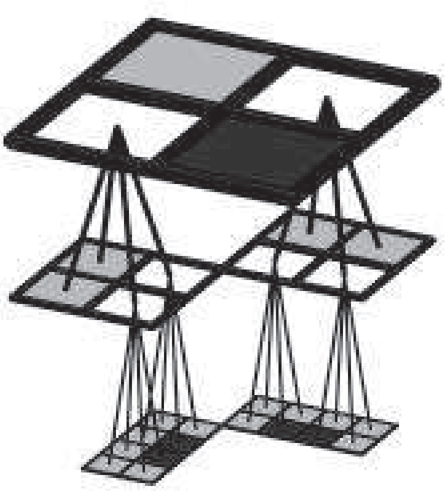


Figure C.3: Quad search tree

C.1.2 XML configuration file overview

The XML file

A sample XML file is provided:

```
<?xml version="1.0" encoding="UTF-8"?>
<ponthy_config>
    <car>
    <owner owner=”Kenn Sebesta”/>
    <make make=”Smart”/>
    <model model=”Roadster”/>
    <year year=”2004”/>
    <color color=”grey”/>
</car>
<engine>
    <type type=”698cc Inline 3 Turbo Suprex”/>
    <displacement displacement=”698cc”/>
    <power power=”60kW @5250rpm”/>
    <torque torque=”110Nm @4500rpm”/>
    <compression comp=”9:1”/>
    <idle rpm=”850”/>
    <inertia lumped=”0.061”/>
</engine>
<wheels>
    <tire summer=”205/45 R16”/>
    <inertia lumped=”0.62”/>
    <!--Inertia PER wheel/tire assembly-->
</wheels>
<accelerometer type=”LIS3LV02DL”>
    <calibration>
        <roll phi=”2.58287237237842”/>
        <pitch theta=”0.0277744014756975”/>
        <yaw psi=”1.66349305108413”/>
        <scale scale=”0.989656407165839”/>
    </calibration>
</accelerometer>
<transmission type=”6-spd sequential”>
    <!--These are the final drive ratios between the engine and the wheel axel-->
    <reverse gear=””/>
    <first gear=”13.25”/>
    <second gear=”9.604”/>
    <third gear=”7.37”/>
    <fourth gear=”5.796”/>
    <fifth gear=”4.208”/>
    <sixth gear=”3.23”/>
</transmission>
<drag>
    <aero Cd=”0.39”/>
    <rolling Crr=”0.0125”/>
</drag>
<area frontal=”1.93”/>
<mass wet=”844”/>
<inj>
    <calibration fuelPerSecond=”0.008”/>
    <calibration lag=”-0.0004”/>
    <!--The difference between the actual injector
    opening time and the injector voltage pulse
    length, in [s].-->
    <timing pattern=”sequential”/>
</inj>
<vss>
    <calibration metersPerTick=”0.270183975491887”/>
</vss>
<file Name=”2009-10-27.dat”/>
```

```
    <file Name="2009-10-28.dat"/>
    <file Name="2009-10-30.dat"/>
    <file Name="2009-11-01.dat"/>
</ponthy_config>
```

This file can be adapted to any new car. For the most part, it is self explanatory. However, included are explanations for the few that are not:

- **engine–> inertia lumped:** the total lumped rotational inertia for all of the drivetrain *except* the tires and wheels.
- **accelerometer–> calibration:** the rotation matrix and scale that must be applied to the accelerations measured by the sensor in order to bring them into the body frame.
- **drag–> Cd:** Coefficient of air resistance
- **drag–> Crr:** Coefficient of rolling resistance
- **area–> frontal:** maximum cross-sectional area as seen from the front
- **mass–> wet:** car mass including all fluids (coolant, oil, windshield wiper fluid, etc...)
- **inj–> calibration –> lag:** the difference between the actual injector opening time and the injector voltage pulse length, in [s]
- **inj–> calibration –> fuelPerSecond:** the quantity of fuel delivered by *all* injectors in one second
- **vss–> calibration:** how far the car rolls forward per vss pulse
- **file Name:** the individual trip data files that have been imported for a given car.

C.1.3 Observer file generation overview

When data is imported, a certain number of files are generated:

- ***<file_name>*, data.mat** Data extracted from all sensors, *except* GPS
- ***<file_name>*, extract.mat** Data extracted from the GPS NMEA strings
- ***<file_name>*, results.mat** The Kalman Filter results
- ***<file_name>*, DEM.mat** Optional. An ordered list of the ways taken during a trip.
- ***<file_name>*, gps.txt** All the GPS NMEA strings in text format
- ***<file_name>*, spd.kmz** A Google Earth file in KML format that shows the trip path and speed. See App. F for details.

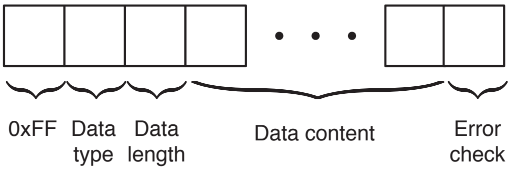


Figure C.4: Overall data format

C.1.4 Data packet overview

Here we discuss the data format used by the sensor suite for saving data to the SD card.

Logged data is saved sequentially onto the SD card, with all data sources interwoven together.

Each type of data is prefaced by a preamble and ended with an error checking byte (modified CRC). (See Fig. C.4)

1. The first byte is always 0xFF.

2. The second byte indicates the message type:
   0x00 Error message
   0x01 Accelerometer data
   0x02 GPS data
   0x03 Vss data
   0x04 Injector data
   0x05 Time skew data

3. The third byte is the message length, including the error check.

4. The message itself depends on the data type. See the following section, C.1.5.

5. Finally, the message is terminated by an error-checking byte. This byte is calculated as the sum modulo 256 of a subset of the message bytes, starting with the data length byte and continuing to the last message byte (therefore not including the error-checking byte itself).

Note 32 *Classically, CRC algorithms use XOR operations, instead of addition modulo 256. Future versions of the logging program will move to the true CRC algorithm.*

C.1.5 Data types

C.1.5.1 Error messages

Error messages are saved in ASCII human-readable format, followed by a newline.

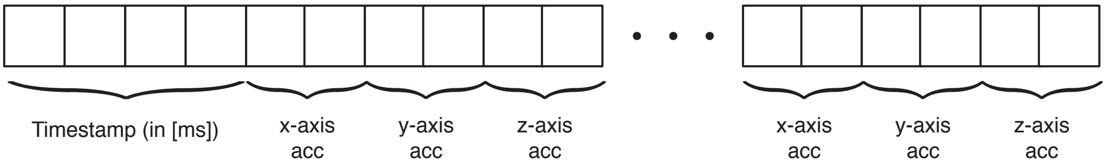


Figure C.5: Accelerometer data format

C.1.5.2 Accelerometer data

The accelerometer data (Fig. C.5) is composed of:

- 4-byte timestamp in [ms]
- 2-byte x-axis acceleration data in [mg]
- 2-byte y-axis acceleration data in [mg]
- 2-byte z-axis acceleration data in [mg]

C.1.5.3 GPS data

GPS data is saved as NMEA raw strings, without any processing. Since data is only written to memory once the 512-byte buffer is full, there is no guarantee that any individual GPS data packet contains a complete NMEA string.

C.1.5.4 Vss data

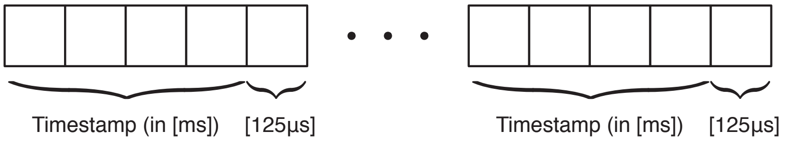


Figure C.6: Vss data format

Vss data (Fig. C.6) is composed of:

- 4-byte timestamp in [ms]
- 1-byte timestamp representing [125μs]

C.1.5.5 Injection data

Injector data (Fig. C.7) is composed of:

- 4-byte timestamp in [ms]
- 2-byte injector duration in [250ns] increments

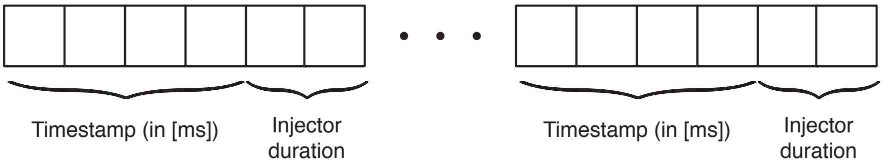


Figure C.7: Injection data format

C.1.5.6 Skew data

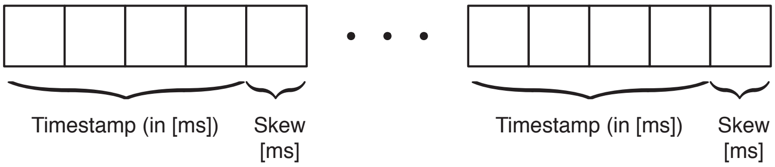


Figure C.8: Skew data format

Skew data (Fig. C.8) is composed of

- 4-byte timestamp in [ms]
- 1-byte skew in [ms]

Appendix D

CAN bus

Contents

D.1 Overview

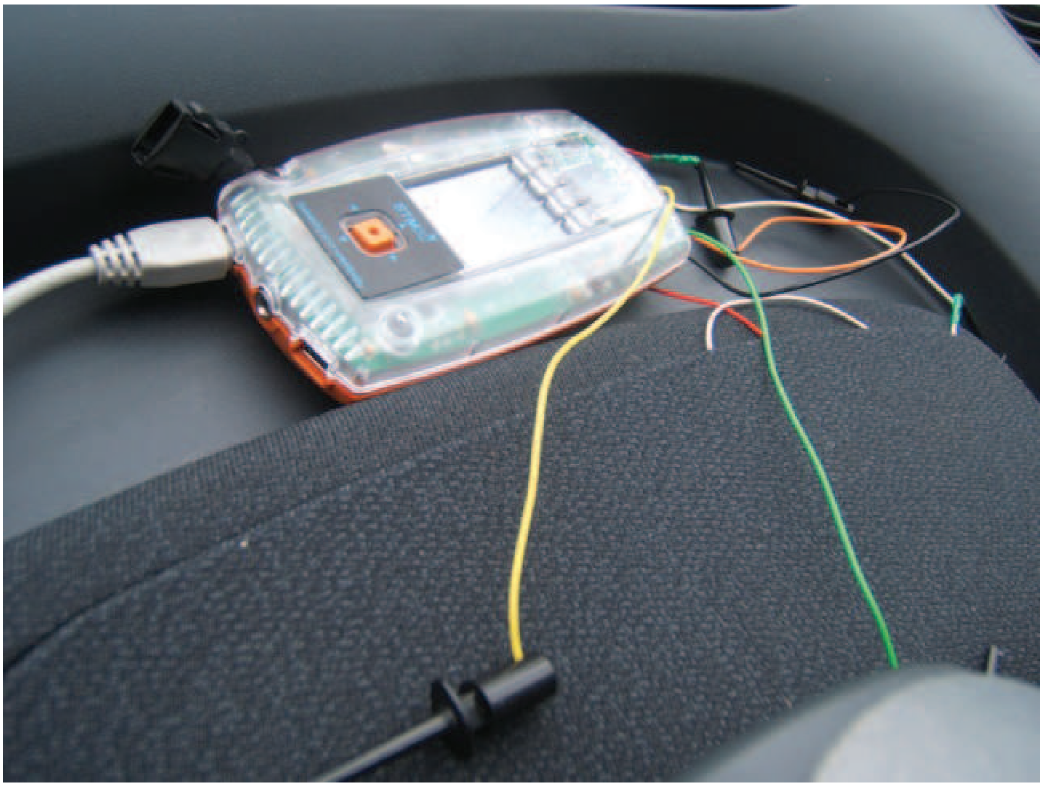


Figure D.1: CAN bus installation

The CAN bus (a.k.a. CANbus), is a fault-tolerant, bi-directional bus that was originally designed for automotive use, but has since become a quasi-standard in industry and noisy environments. CAN is mandatory in all European vehicles since 2004.

While CAN is technically only a bit transmission protocol, in the automotive world the term is used interchangeably to mean ISO 11898-2, which completely defines the physical layers, including signal levels, wire count, etc...

In the case of ISO 11898-2, there are two wires, both transmitting the same bit. The wires pass between dominant, 0, and recessive, 1, states. The dominant is so-named since if both a dominant and recessive bit are sent simultaneously, only the dominant bit is transmitted on the bus.

Technically, this is achieved by having an open-collector style bus, where the line is pulled HI (recessive) by a pull-up resistor, and then is pulled LO (dominant) by a transistor. The transistor obviously being far stronger than the pull-up resistor, when both a dominant and recessive bit are sent at the same time by two different devices, the transistor signal squashes the resistor signal.

A CAN message is composed of an arbitration field, a control field, a data field, and a CRC field. The arbitration field is designed such that if two CAN devices start speaking simultaneously, then the one with lower arbitration has priority.

(This occurs by the physics of the communication: if two arbitration fields are different, at some moment one device will transmit a 0 (dominant) when the other is transmitting a 1 (recessive). Since the 1 is recessive, all other devices on the bus will see a 0. As each CAN device is monitoring the bus as well as transmitting on it, as soon as a device recognizes that its recessive bit has been overwritten by a dominant, it stops transmitting. As such, for an 11-bit arbitration field, `0b00000000000` is obviously always highest priority and `0b11111111111` is always lowest.)

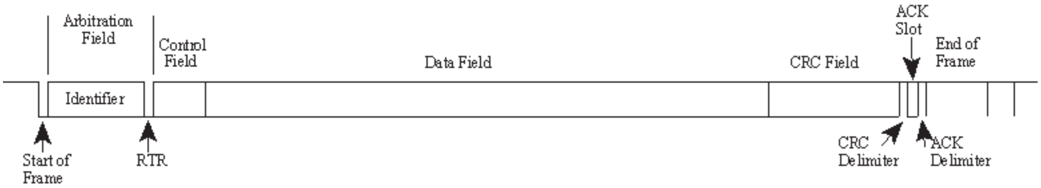


Figure D.2: CAN 2.0A ("standard CAN") Data Frame

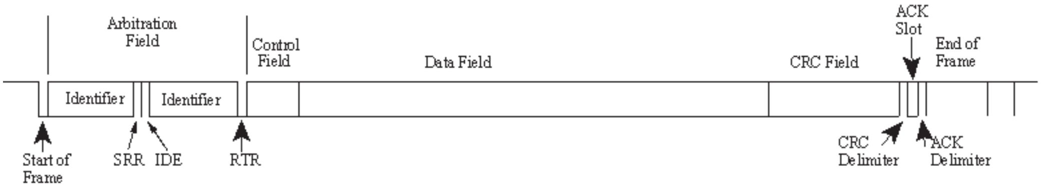


Figure D.3: CAN 2.0B ("extended CAN") Data Frame

Convention states that the arbitration field is also the message ID. While this is by no means a requirement of the protocol, unless there is strong reason to suspect otherwise it can be assumed that an arbitration field is the same as the message ID.

The data field is always between 0 and 8 bytes long, as indicated by the control field.

D.2 Case study

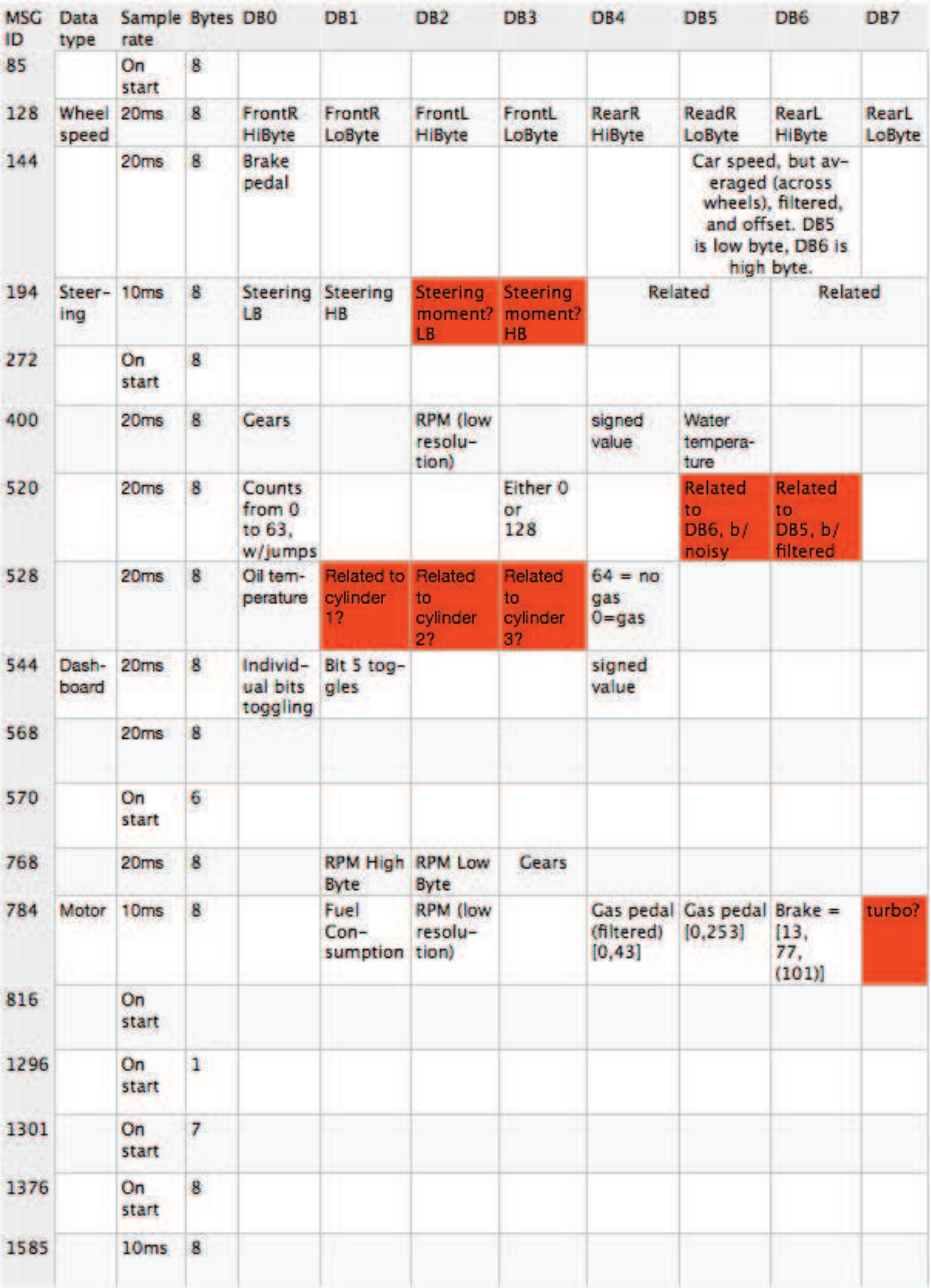
Inspired by [42], and armed with the above information, a Smart Roadster's CAN bus was reverse engineering by logging all CAN messages, and then comparing with logged data from the sensor suite. There are many messages for which the meaning was never discovered, but they were unnecessary for our purposes. Some certainly indicate if headlights are on, if the convertible top is open, etc... (These could be later interesting for trying to gauge electrical load as a part of the overall engine output.)

A simple circuit was designed to interface to the automotive CAN bus. The circuit needs

- A CAN transceiver to turn dominant/recessive bits into 0's and 1's and vice-versa
- A CAN controller in order to read the bit stream, perform error checking, bit destuffing, etc...

In this case, we chose the STM32F103VET6, an ARM Cortex M3 microcontroller with onboard CAN controller, and coupled it to a MAX3051 CAN transceiver.

The CAN messages as decoded are below:



MSG ID	Data type	Sample rate	Bytes	DB0	DB1	DB2	DB3	DB4	DB5	DB6	DB7
85		On start	8								
128	Wheel speed	20ms	8	FrontR HiByte	FrontR LoByte	FrontL HiByte	FrontL LoByte	RearR HiByte	ReadR LoByte	RearL HiByte	RearL LoByte
144		20ms	8	Brake pedal					Car speed, but averaged (across wheels), filtered, and offset. DB5 is low byte, DB6 is high byte.		
194	Steering	10ms	8	Steering LB	Steering HB	Steering moment? LB	Steering moment? HB	Related		Related	
272		On start	8								
400		20ms	8	Gears		RPM (low resolution)		signed value	Water temperature		
520		20ms	8	Counts from 0 to 63, w/jumps			Either 0 or 128		Related to DB6, b/ noisy	Related to DB5, b/ filtered	
528		20ms	8	Oil temperature	Related to cylinder 1?	Related to cylinder 2?	Related to cylinder 3?	64 = no gas 0=gas			
544	Dashboard	20ms	8	Individual bits toggling	Bit 5 toggles			signed value			
568		20ms	8								
570		On start	6								
768		20ms	8		RPM High Byte	RPM Low Byte	Gears				
784	Motor	10ms	8		Fuel Consumption	RPM (low resolution)		Gas pedal (filtered) [0,43]	Gas pedal [0,253]	Brake = [13, 77, (101)]	turbo?
816		On start									
1296		On start	1								
1301		On start	7								
1376		On start	8								
1585		10ms	8								

Figure D.4: CAN Messages table

Appendix E

Data logger circuit

Contents

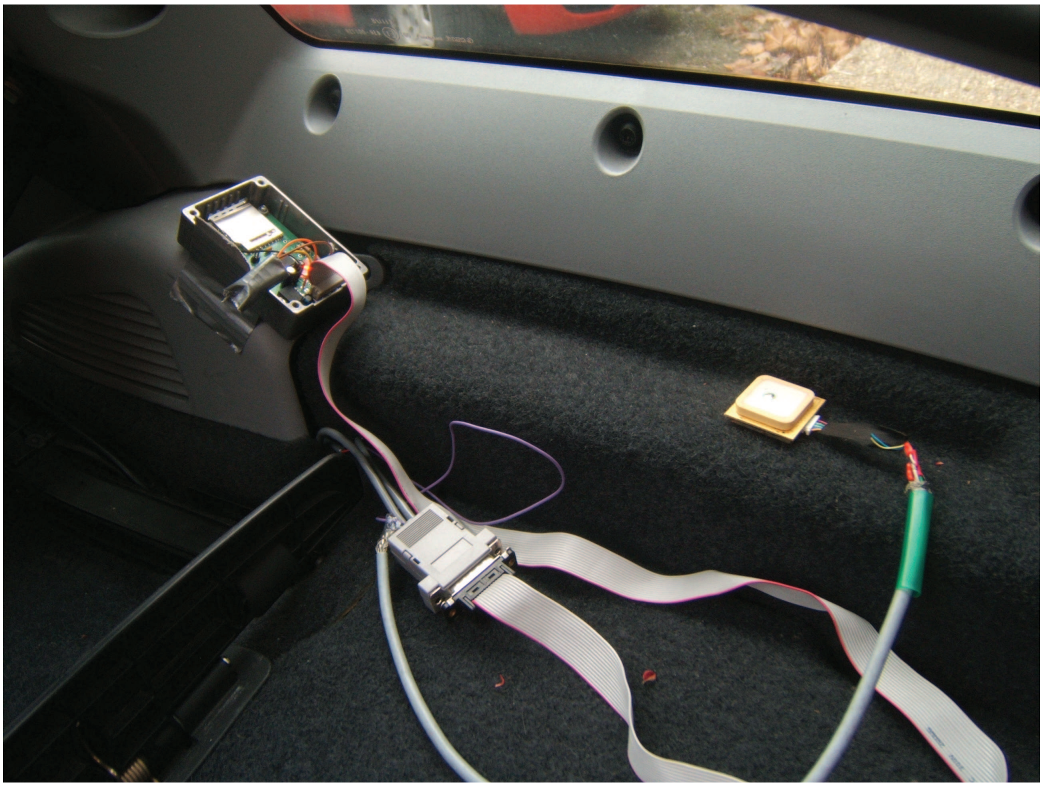


Figure E.1: Data logger installation

E.1 Circuit-board schematic

The circuit board schematic was designed in Eagle CAD.

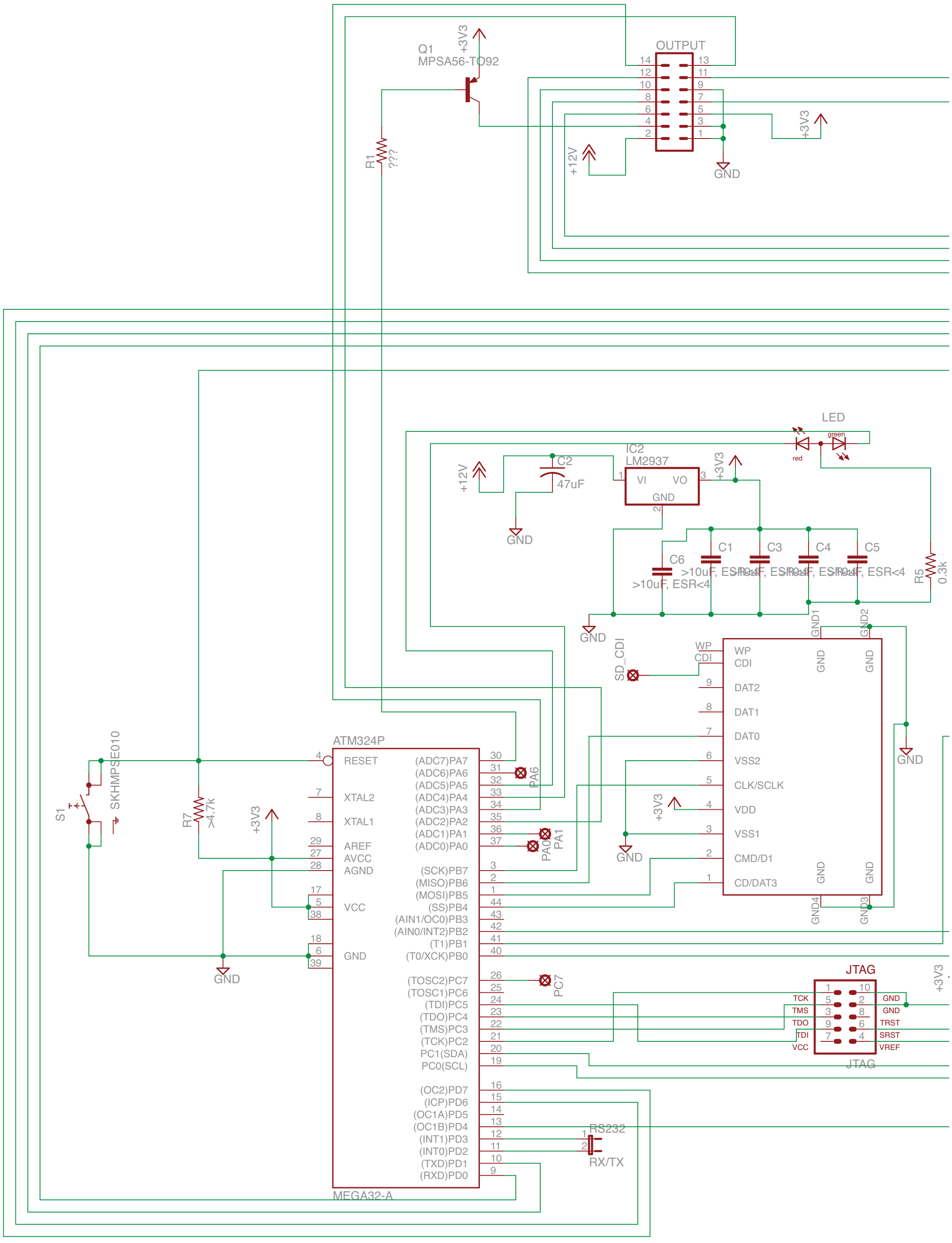


Figure E.2: Data logger schematic, pg. 1

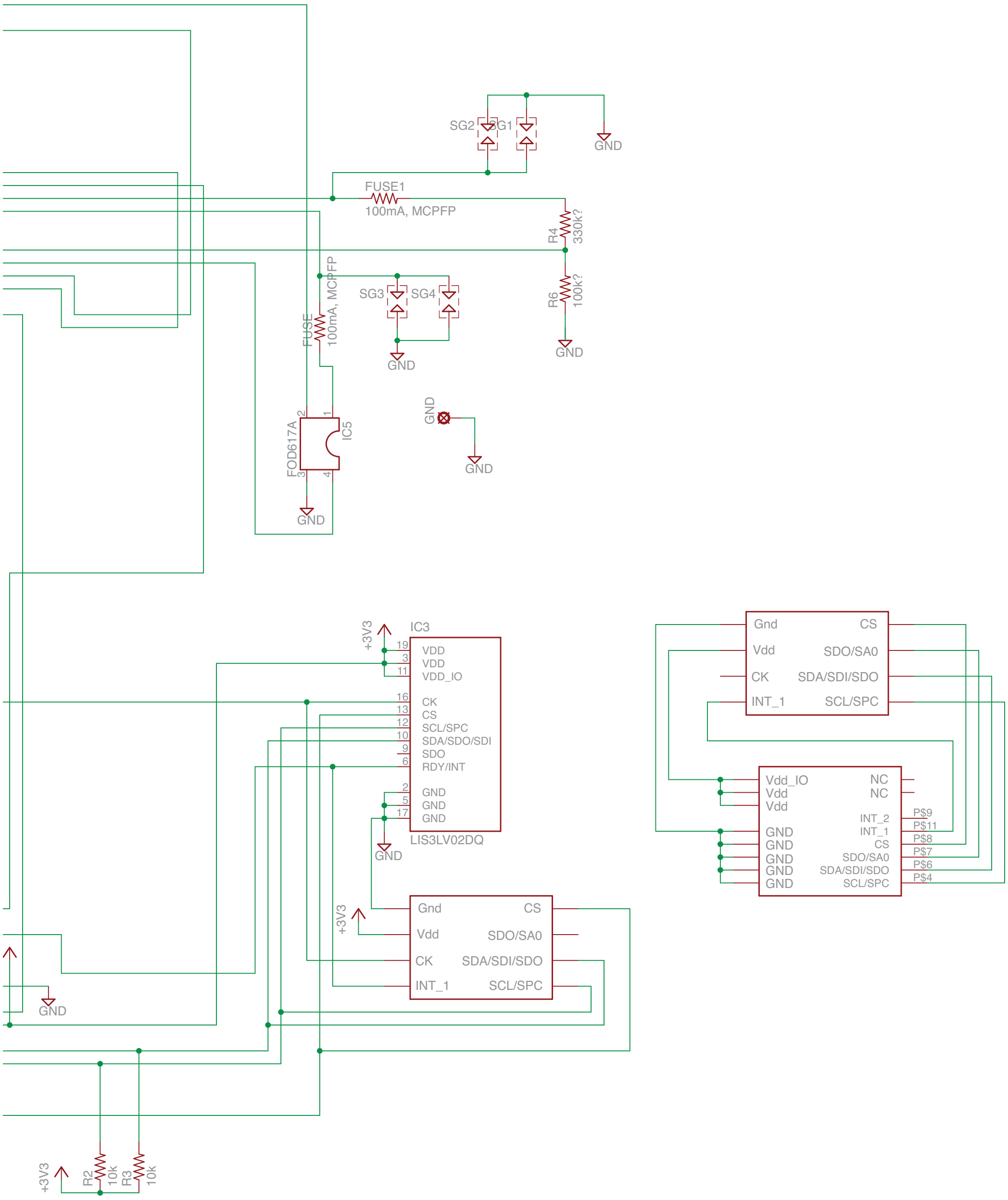


Figure E.3: Data logger schematic, pg. 2

E.2 Circuit-board layout

The circuit board was designed in Eagle.
Top layer:

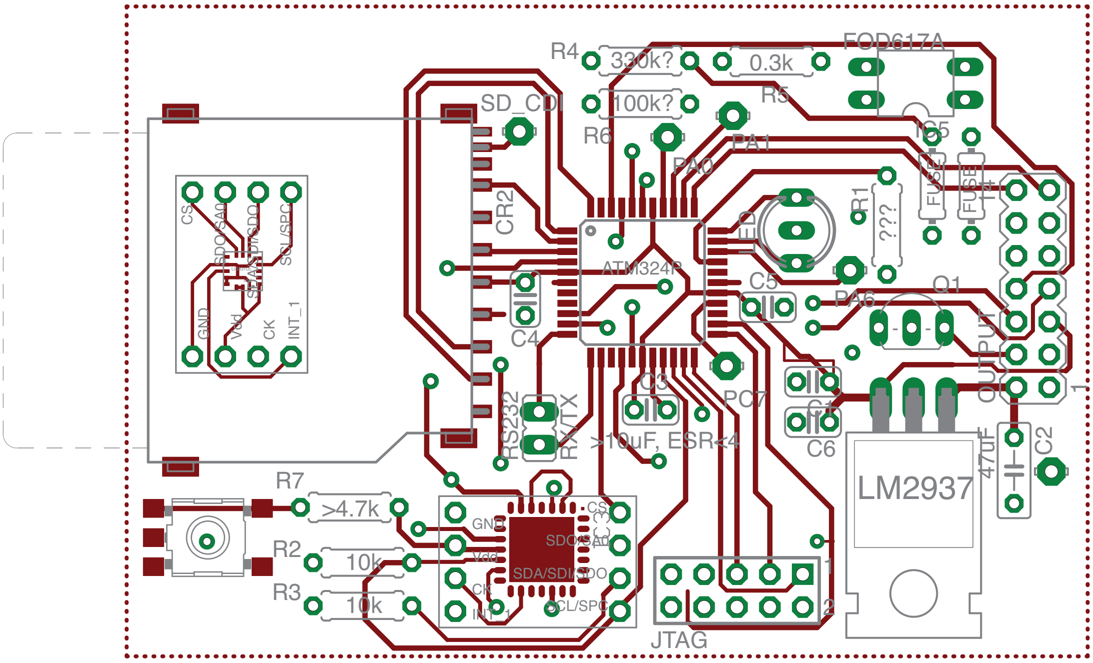


Figure E.4: Data logger layout, top

Bottom layer:

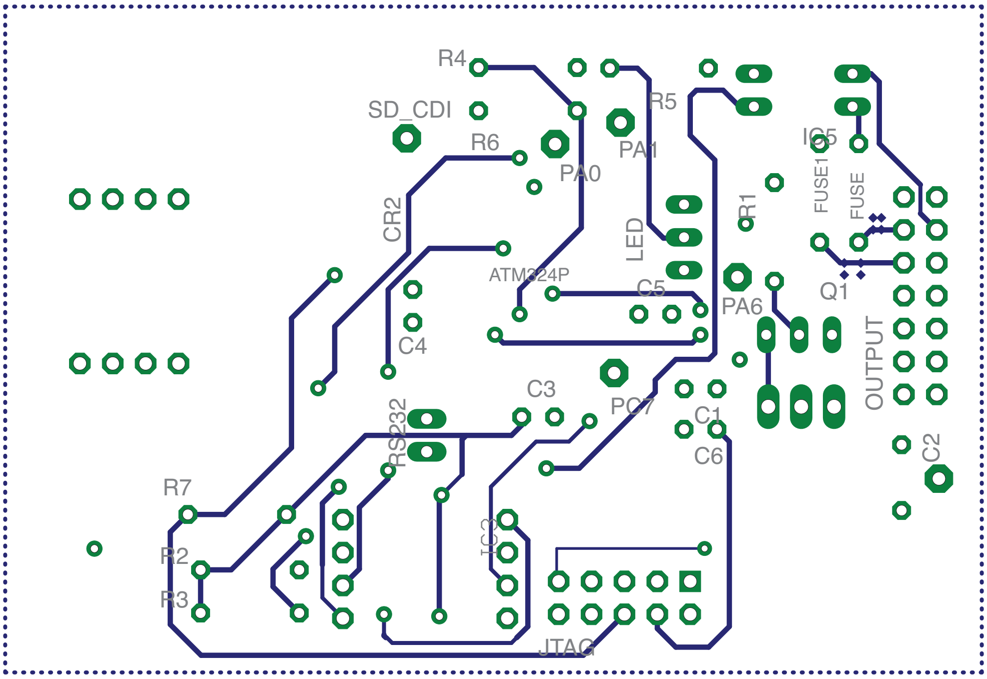


Figure E.5: Data logger layout, bottom

E.3 Improvements

If the reader would like to use these schematics directly, s/he would be well advised to take into account the following bugs and worthwhile upgrades:

- The ATMega has been superseded by the XMEGA line. These chips are in all ways superior; it is cheaper, faster, more accurate, and has more peripherals.
- In the functioning board, the transistor Q1 had to be replaced by an optorelay because the FV-M8 GPS had trouble initializing correctly. The optorelay has a faster rise time. It's primary disadvantage is cost, so with a different GPS the transistor might not need replacing.
- The CK trace on the accelerometer can be removed. It is not important that the accelerometer have an external syncing signal.
- The reset button should be reconfigured to use an RC filter so that a reset is delayed by several seconds, in order that the button can also be used for other purposes.
- The PPS input should have an RC filter

A future version could/should include:

- A CAN bus and LIN bus transceiver, in order to be compatible with all foreseeable cars.
- An analog input that can be used for fuel pressure sensing (diesel cars)
- A DC-DC converter in order to draw less current from the car's battery (important when the car is off)

As a small bonus, a separate circuit trace was made under the SD card. This is specifically designed to be able to test the STMicro LIS331DL. The relevant area must first be cut out from the circuit board. The pad layout is designed to be soldered on top of the existing accelerometer area. In fact, **any** digital accelerometer can be installed into this area, as long as the output pads correspond.

In addition, there are a number of PA* pads on the board. These are unconnected ADC inputs, and can be used for future expansion, depending on needs.

E.4 Sensor suite parts manifest

Part	Value	Count
ATM324P	ATMega324p	1
C1	$\geq$ 10uF, ESR $\leq$ 4	1
C2	47uF	1
C3	$\geq$ 10uF, ESR $\leq$ 4	1
C4	$\geq$ 10uF, ESR $\leq$ 4	1
C5	$\geq$ 10uF, ESR $\leq$ 4	11
C6	$\geq$ 10uF, ESR $\leq$ 4	1
CR2		
FUSE	100mA fast blow	2
IC2	LM2937	1
IC3	LIS3LV02DQ	1
IC5	FOD617A	1
LED	Bi-color	1
Q1	PNP transistor	1
R1	47Ω	1
R2	10kΩ	1
R3	10kΩ	1
R4	330kΩ	1
R5	300Ω	1
R6	100kΩ	1
R7	$\leq$ 4.7kΩ	1
S1	SKHMPSE010	1
JTAG	10-pin IDC header	1
OUPUT	14-pin IDC header	1
RS232	2-pin IDC header	1

Appendix F

Google Earth

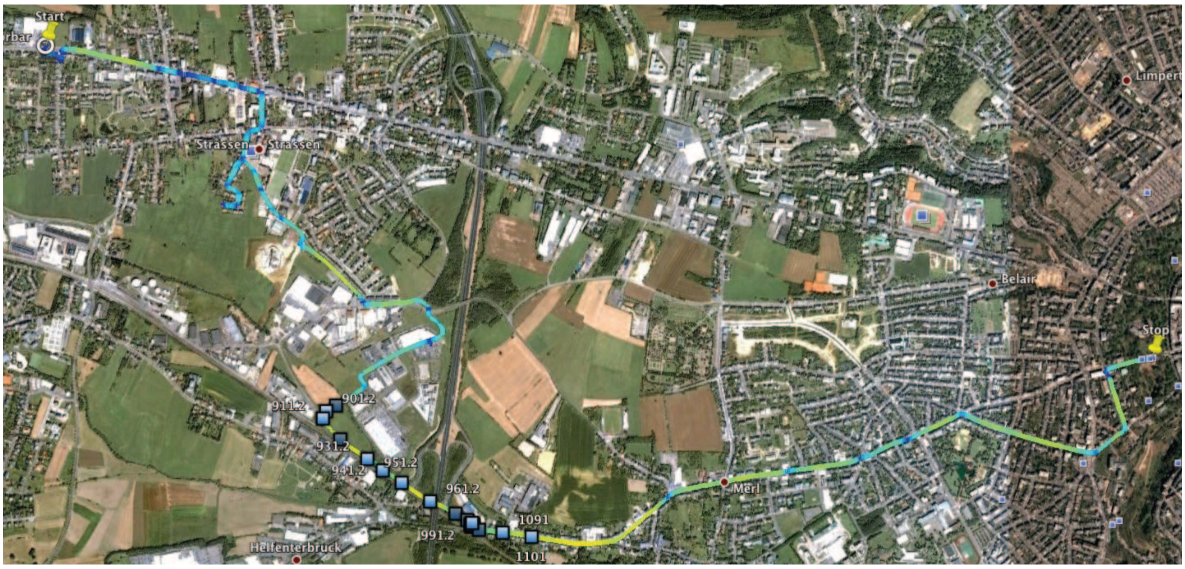


Figure F.1: Google Earth

The Google Earth Toolbox [48], is a useful collection of utilities that aid in the presentation, understanding, and analysis of geodetic data. Certainly, this project could not have gone as far as it did if it were not for the excellent capabilities of Google Earth and Google Earth Toolbox.

For this project, Google Earth Toolbox code was modified: cleaned, streamlined, and upgraded. Specific needs that were not present in Google Earth Toolbox were addressed, such as the (in)ability to display properly formatted XML code, or to output arbitrary XML code to the Google Earth KML format.

The Google Earth files generated by the Matlab code allow one to see, at a glance, car path, car speed, starting and ending points, and trip time. Take for instance Fig. F.1. The KML output file identifies:

- Start and Stop locations, helpfully marked with thumbtacks.
- Speed, as represented by color. The colormap was intentionally chosen to imitate Matlab's "jet". Dark blue represents a stopped car, whereas deep red represents high speed. All outputs are normalized to 180km/hr, a value that seems quite reasonable for the upper end, and yet still leaves room for definition at slower speeds
- Waypoints, as represented by the blue boxes with floating numbers. This indicates the trip time that the car was at the particular spot.

Trip time is perhaps the most important feature of the Google Toolbox Extensions, as it allows easy cross correlation between logged data time (absolute ms since the trip started) and GPS data time (UTC).

Code modifications have been submitted to Google Earth Toolbox for integration into future versions of the Toolbox.

Appendix G

The Ball and Wheel: a worked example for Pontryagin's Maximum Principle

Contents

G.1 Introduction

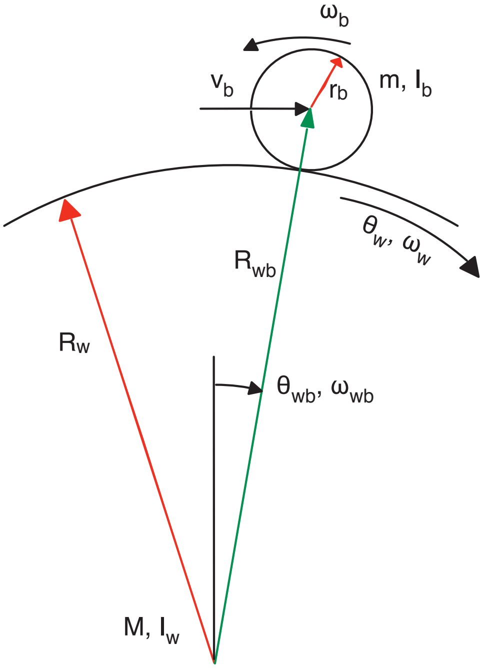


Figure G.1: Diagram of Ball and Wheel

The Ball and Wheel system is made of a spherical ball with angular momentum I_b rolling on the rim of a wheel with angular momentum I_w. The wheel is driven by a torque source, τ_w that is limited by $\tau_{w,min} \leq \tau_w \leq \tau_{w,max}$. The objective is to balance the ball at the unstable equilibrium point at the very top of the wheel. In order for the system to be *controllable* (see Sec. B.1.2), $\tau_{w,min} < 0$ and $\tau_{w,max} > 0$ must hold. The ball is acted upon by gravity and by friction with the wheel rim. We assume a no-slip condition between the wheel and the ball.

This appendix presents a two-variable system, where only the ball's position and translational velocity with respect to the wheel are modeled. In consequence, the wheel and the ball can have arbitrary angular velocities, as long as the ball does not physically move from the set-point position. In this example, we develop a full synthesis solution.

Optimal target: In this example, we are looking for the time optimal control. In other words, we want the sequence of controls that minimize the time required to go from an initial point to a final set-point. Note that this is a different optimization problem from the one we wish to solve for the car, which minimizes fuel; and is also different from the LQR (See Sec. B.2.2), which seeks to minimize the state and control deviations.

The time optimal cost is expressed as

$$J(\mathbf{x}, \mathbf{u}, t) = \int_0^{t_f} 1dt \tag{G.1}$$

and thus from eq. (B.12)

$$L(\mathbf{x}, \mathbf{u}, t) = 1 \tag{G.2}$$

G.2 Two-state example

(We rapidly present here the Ball and Wheel dynamic equations, without modeling them. Interested readers are encouraged to read [23] for a detailed derivation using Euler-Lagrange.)

The state vector is

$$\mathbf{x} \;=\; \begin{pmatrix} \theta_{wb} \\ \dot{\theta}_{wb} \end{pmatrix} \tag{G.3}$$

with dynamic equations

$$\dot{\mathbf{x}} = \begin{pmatrix} \dot{\theta}_{wb} \\ \left(\frac{1}{R+r}\right)\left[\frac{I_b R \tau_w + mg\sin(\theta_{wb})(I_w r^2 + I_b R^2)}{I_b I_w + I_w m r^2 + I_b m R^2}\right] \end{pmatrix} \tag{G.4}$$

G.2.1 Linearization by feedback

Linearization by feedback is a technique which greatly facilitates designing the control. The idea behind this approach is to find a diffeomorphism and a state-feedback control law that transform the nonlinear system into a linear time-invariant (LTI) system, as these LTI systems have certain desirable properties of convergence and stability. A complicated nonlinear control can be easily designed on the linear system. We give here the equations for a *full-state-feedback* linearization.

Let

$$u_f(\tau_w, \theta_{wb}) = I_b R \tau_w + mg\sin(\theta_{wb})(I_w r^2 + I_b R^2) \tag{G.5}$$

and

$$\gamma = \left(\frac{1}{R+r}\right)\left(\frac{1}{I_b I_w + I_w m r^2 + I_b m R^2}\right) \tag{G.6}$$

This transforms (G.4) into

$$\dot{\mathbf{x}} = \begin{pmatrix} \dot{\theta}_{wb} \\ \gamma u_f \end{pmatrix} = \begin{pmatrix} x_2 \\ \gamma u_f \end{pmatrix} \tag{G.7}$$

and

$$\tau_w = \frac{u_f - mg\sin(\theta_{wb})(I_w r^2 + I_b R^2)}{I_b R} \tag{G.8}$$

G.2.2 Pontryagin's Maximum Principle

The primordial goal is to find the equations for the *costate vector* (a.k.a. *adjoint vector*), $\lambda^T = (\lambda_1 \dots \lambda_{nx})$, as these describe the control.

First, recall the Hamiltonian, eq. (B.13)

$$H(\mathbf{x}, \mathbf{u}, \lambda) = \lambda^T f(\mathbf{x}, \mathbf{u}) - L(\mathbf{x}, \mathbf{u}, t) \tag{G.9}$$

then combining (G.2), (G.7) (where $f(\mathbf{x}, \mathbf{u}) = \dot{\mathbf{x}} = (\dot{\theta}_{wb} \quad \gamma u_f)^T$), and (G.9)

$$H = \lambda_1 \dot{\theta}_{wb} + \lambda_2 \gamma u_f - 1 \tag{G.10}$$

En suite, we want the control that maximizes the Hamiltonian across the trajectory. We find this classically, by 1) testing for absolute extrema at the control boundaries and 2) looking for local extrema by taking the partial derivative of H with respect to $\mathbf{u}$ and setting equal to 0

$$\frac{\partial H}{\partial u_f} = 0 = \lambda_2 \gamma \tag{G.11}$$

This reveals that there are no local extrema as a function of u_f. Therefore, the absolute extrema are the only extrema.[1]

Recalling that the torque is limited by $\tau_{w,min} \leq \tau_w \leq \tau_{w,max}$ (as described in the initial problem statement), we deduce that the u_f that maximizes H is

$$u_f^* = \begin{cases} u_{f,min} &= u_f(\tau_{w,min}, 0) \quad \text{when } \lambda_2 < 0 \\ u_{f,max} &= u_f(\tau_{w,max}, 0) \quad \text{when } \lambda_2 > 0 \end{cases} \tag{G.12}$$

Note 33 *We choose $\theta = 0$ as it has a negligible effect on u_f^*. In fact, this step is not necessary if we were finding a numerical solution, but in order to have a simple synthesis we make this very minor simplification in calculating the control saturations.*

Note 34 *This type of control is called* bang-bang*. The control is either maximum or minimum, depending only on the sign of λ_2. As such, it becomes paramount to know* when *λ_2 crosses zero. This bears repeating: here the most important information we seek from the costate vector is the zero-crossing time. Once we can express this analytically for any (controllable) state, then we are done and we have only to check that the trajectory is in fact optimal.*

Next, recall the relationship between the Hamiltonian and the costate vector (eq. (B.14))

$$\dot{\lambda} = -H_{\mathbf{x}} \tag{G.13}$$

which gives

$$\begin{aligned} \dot{\lambda}_1 &= -\frac{\partial H}{\partial x_1} = -\frac{\partial H}{\partial \theta_{wb}} = 0 \\ \dot{\lambda}_2 &= -\frac{\partial H}{\partial x_2} = -\frac{\partial H}{\partial \dot{\theta}_{wb}} = -\lambda_1 \end{aligned} \tag{G.14}$$

Thus we can deduce that $\lambda_1 = C_1$, and $\lambda_2 = -C_1 t + C_2$

Let's pause for a moment and take stock of what we know:

1. $H = 0$ (From eq. (B.19))

2. u_f is bang-bang, i.e. it is either $u_{f,min}$ or $u_{f,max}$

3. λ_1 is constant, and λ_2 is a line dependent on time, with slope $-\lambda_1$

[1] Of course, this goes without saying as it is a necessary result for a linear system. The approach is presented for completeness in the case of a non-linear PMP optimization.

Point (1): We know that the Hamiltonian is always 0, and we know that at $t = T$ that $u_f(T) = u_f^*$. We also know that $\dot{\theta}_{wb}(T) = 0$ and thus from (G.10) we can calculate that

$$\lambda_2(T) = \frac{1}{u_f^*}$$

Unfortunately we cannot further use the transversality conditions, since— from Note 27 (pg. 112)— due to the fact that our final point is $\mathbf{x} = \vec{0}$ we cannot conclude anything more from the condition.

Point (2): While we know what controls we will give, it still remains to be seen *when* we will give these controls.

Point (3): Unfortunately, we have no additional equations with which to fix C_1, C_2. We must therefore perform an exhaustive analysis of λ_2 in order to understand when to switch between controls.

Note 35 *Keep in mind that Pontryagin's Maximum Principle is a* necessary *but not* sufficient *condition for optimality. The fact that we have found trajectories that maximize H does not, therefore, mean that they are necessarily optimal.*

Inverse time and "time to go": Optimization problems are naturally considered in inverse time. This makes a certain amount of sense when it is considered that the objective is known, but not necessarily the initial point. While we can easily calculate the finite number of ways to optimally approach the final point, there are an infinite number of ways to leave the initial point, and we have no way to distinguish those that are possibly optimal from those that are not.

Thus, we talk about integrating from the final target $\mathbf{x}(T)$ to the initial point $\mathbf{x}(0)$. The author suggests that if the above is not clear, that it be reread, and kept in mind throughout the following analysis. It bears repeating: *in optimal control, we integrate starting at the final target and only stop once we reach the initial target.*

One confusing facet of inverse time is how to refer to it. If one talks of time T in an inverse time problem, does this refer to time T with respect to forward or backward time? For this reason, we prefer the phrase "time to go", which, when scrupulously used, removes all ambiguity. We represent "time to go" by s and thus

$$\begin{array}{ccccccc} t & = & 0 & \mapsto & s & = & T \\ t & = & T & \mapsto & s & = & 0 \end{array}$$

Inversing the dynamic equations is straightforward. It suffices to take the negative of the dynamic equation

$$\dot{\mathbf{x}}^* = -\dot{\mathbf{x}} \tag{G.15}$$

This gives us the new extended system

$$\begin{pmatrix} \dot{\mathbf{x}}^* \\ \dot{\lambda}^* \end{pmatrix} = \begin{pmatrix} -x_2^* \\ -\gamma u_f^* \\ 0 \\ \lambda_1 \end{pmatrix} \tag{G.16}$$

As we know $\mathbf{x}^*(0)$ and $\mathbf{x}^*(s = T)$ this can be seen as a BVP (Boundary Value Problem) which can be solved, for example with shooting or multiple-shooting methods. (See Sec. B.3.) However, in this particular case we can analytically turn the BVP into an IVP (Initial Value Problem) in order to have a *synthesis* of what to do across the entire space of attainable points (the control convex).

In other words, we create the synthesis by finding the solution for the exhaustive list of all possible IVPs.

Consider that $\boldsymbol{\lambda_2}^*$, like $\boldsymbol{\lambda_2}$, is a linear function of time-to-go of the form

$$\lambda_2^*(s) = \lambda_1^* s + \beta = \alpha s + \beta \tag{G.17}$$

where $\lambda_1^* = \text{Constant} = \alpha$. We know that at time-to-go $s = 0$ that

$$\lambda_2^*(0) = \lambda_2(T) = \frac{1}{u_f^*} \tag{G.18}$$

and so we clearly see that $(\lambda_1^*)0 + \beta = \beta = \dfrac{1}{u_f^*}$

Knowing that $u_f(T)$ can be either positive or negative, we can directly infer the nature of λ_2^*— it is a line that has at most 1 zero-crossing, depending only on whether α and β have the same sign or not. As the two cases are symmetrical in the linear case, we will only treat the case where $\beta > 0$.[2]

If $\beta > 0$ and $\alpha \geq 0$, then from (G.17) $\lambda_2^*(s) > 0$ and has no zero-crossing for all time-to-go s. Therefor $u_f^*(t) = u_{f,max}$ and thus by integrating eq. (G.16), we can solve the IVP which yields

$$\begin{aligned} x_1^*(s) &= \left(\frac{1}{2}\gamma u_{f,max}\right) s^2 \\ x_2^*(s) &= -(\gamma u_{f,max}) s \end{aligned} \tag{G.19}$$

And thus by rearranging (G.19) we see that in this case the optimal trajectory lies on the parabola $x_1^* = \dfrac{1}{2\gamma u_{f,max}}(x_2^*)^2,\ x_1^* \geq 0,\ x_2^* \leq 0$. This is half of the **switching locus**, the other half occurring when $\beta < 0$.

[2] Note that $B \to 0$ is not treated as this would imply $u_f^* \to \infty$. Clearly if the control can be infinite, then in the case of a linear system $T \to 0$.

If $\beta > 0$ and $\alpha < 0$, then from (G.17) λ_2^* has one zero-crossing at $0 = \alpha s_0 + \dfrac{1}{u_{f,max}}$

$$s_0 = -\frac{1}{\alpha u_{f,max}} \tag{G.20}$$

and thus by integrating the IVP (and using the final values of the segment from 0 to s_0 as the initial values for the segment from $s \geq s_0$)

$$\left.\begin{array}{rcl} u_f & = & u_{f,max} \\ x_1^*(s) & = & \frac{1}{2}\gamma u_{f,max} s^2 \\ x_2^*(s) & = & -\gamma u_{f,max} s \end{array}\right\} \quad \text{for} \quad 0 \leq s \leq s_0$$
$$\left.\begin{array}{rcl} u_f & = & u_{f,min} \\ x_1^*(s) & = & \gamma(\frac{1}{2}u_{f,min}s^2 + (u_{f,min} - u_{f,max})s_0 s - \frac{1}{2}s_0^2(u_{f,min} - u_{f,max})) \\ x_2^*(s) & = & \gamma(-u_{f,min}s + (u_{f,min} - u_{f,max})s_0) \end{array}\right\} \quad \text{for} \quad s_0 \leq s \tag{G.21}$$

Now we must perform the same analysis for $\beta < 0$, but thanks to the system's symmetry we find equivalent results, only negative.

It might not seem it, but we're almost done.

G.3 Interpreting the synthesis

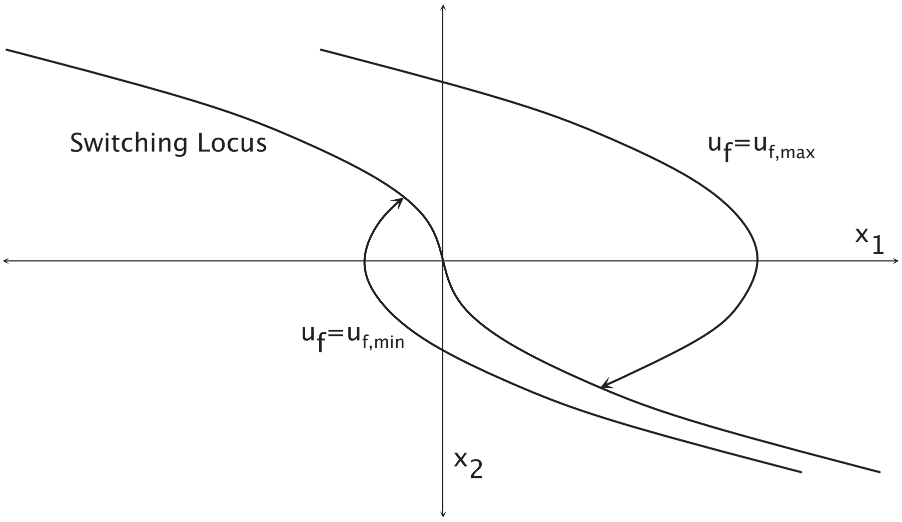


Figure G.2: Optimal control synthesis for Ball and Wheel

Note that the first part of eq. (G.21) is identical to eq. (G.19). From this we can gather that the optimal control from any space intersects the switching locus and then commutes to follow the parabola to the final destination. (see Fig G.2)

In other words, there are four possibilities:

- The initial point is on the switching locus to the right of the origin. In this case, there are no commutations to be made and the optimal control is to apply u_f^* until the system reaches $\vec{0}$.

- The initial point is above the switching locus. In this case, the optimal strategy is to apply $u_{f,min}^*$ until the switching locus is reached, at which point the bang-bang control commutes to $u_{f,max}^*$, which is applied until the system reaches $\vec{0}$.
- The initial point is on the switching locus to the left of the origin. The exact same as when to the right, only the opposite control is applied.
- The initial point is above the switching locus. The exact same as when the point is above the switching locus, only the opposite sequence of controls is applied.

Thus we have the time optimal control. Incidentally, the optimal time T can be found by simply integrating from the final point $\mathbf{x}_f = \vec{0}$ to the initial point $\mathbf{x}_0$

Note 36 *What happens in the case where the optimal control is not perfectly applied? In this case, one can see that if the optimal control is misapplied, or the system has a perturbation, that this is equivalent to an IVP with a new initial value at the current system state. The system still converges to the set point, only an additional commutation will be required.*

References

[1] Arduino [online]. Available from: http://arduino.cc. 20

[2] S. Ammar and J. C. Vivalda. On the preservation of observability under sampling. *Systems & Control Letters*, 52(1):7 – 15, 2004. 89

[3] ASTM. Standard test method for measuring the longitudinal profile of traveled surfaces with an accelerometer established inertial profiling reference. Technical report, ASTM International, 2004. 24

[4] V.M. Becerra, P.D. Roberts, and G.W. Griffiths. Applying the extended kalman filter to systems described by nonlinear di!erential-algebraic equations. *Control Engineering Practice*, 2000. 94

[5] R. Bellman. *Dynamic Programming.* New Jersey: Princeton Univ. Press, 1957. 108

[6] H.G. Bock and K.J. Plitt. A multiple shooting algorithm for direct solution of optimal control problems. In *Proceedings 9th IFAC World Congress Budapest*, pages pp. 243–247. Pergamon Press, 1984. 117

[7] Nicolas Boizot. *Adaptive High-gain Extended Kalman Filter and Applications.* PhD thesis, University of Luxemburg and Université de Bourgogne, 2010. 87, 90, 97, 99, 100, 101

[8] Nicolas Boizot, Eric Busvelle, and Jean-Paul Gauthier. An adaptive high-gain observer for nonlinear systems. *Automatica, accepted for publication*, 2010. 36, 97

[9] J. Frederic Bonnans and Therese Guilbaud et al. Parametric optimization of hybrid car engines. *Optimization and Engineering*, pages 395–415, 2004. 6

[10] Bernard Bonnard, Ludovic Faubourg, and Emmanuel Trélat. *Mécanique céleste et contrôle des véhicules spatiaux.* Springer, 2006 2006. 67

[11] William M. Boothby. *An Introduction to Differentiable Manifolds and Riemannian Geometry.* Academic Press, 2002. 86, 107

[12] Kevin Burrage. *Parallel and Sequential Methods for Ordinary Differential Equations.* Oxford University Press, 1995. 115

[13] E. Busvelle and J.-P. Gauthier. Observation and identification tools for nonlinear systems. application to a fluid catalytic cracker. *International Journal of Control*, Vol. 78, 3, 2005. 51, 85

[14] E. Busvelle, R. Kharab, A. J. Maciejewski, and J.-M. Strelcyn. Numerical integration of differential equations in presence of first integrals: Observer method. *Applicationes Mathematicae*, 22(3), 1994. 65

[15] C. Chui and G. Chen. *Kalman Filtering with Real-Time Applications.* Springer, 1999. 94

[16] Clarke. The maximum principle under minimal hypotheses. *SIAM J. Control Optim.*, pages 1078–1091, 1974. 2, 112

[17] Collaborative effort. MPGuino, 2010. Available from: http://ecomodder.com/wiki/index.php/MPGuino. 20, 26

[18] Collaborative effort. Open Street Map, 2010. Available from: www.openstreetmap.org [cited 2010-02-08]. 29

[19] B. d'Andréa Novel and M. Cohen de Lara. *Commande linéaire des systèmes dynamiques.* Presses De L'Ecole Des Mines, 2000. 86, 88, 107

[20] M Diehl, H.G. Bock, H. Diedam, and P.-B. Wieber. Fast direct multiple shooting algorithms for optimal robot control. In *Lecture Notes in Control and Information Sciences*, volume Volume 340/2006. Springer Berlin / Heidelberg, 2006. 116

[21] JP. Gauthier and I. Kupka. *Deterministic Observation Theory and Applications.* Cambridge University Press, 2001. 12, 87, 89, 90, 97, 98, 101

[22] Inertial Guidance. *Charles Draper and Walter Wrigley and John Hovorka.* Pergamon Press, 1960. 27

[23] Ming-Tzu Ho, Yi-Wei Tu, and Hao-Shuan Lin. Controlling a ball and wheel system using full-state-feedback linearization. *IEEE Control Systems Magazine*, October 2009. 151

[24] http://www.harpers.org/archive/2004/12/0080313. English as a second language, 2004. Available from: http://www.harpers.org/archive/2004/12/0080313 [cited 2010-03-10]. 1

[25] M. Ivarsson, J. Aslund, and L. Nielsen. Look-ahead control for heavy trucks to minimize trip time and fuel consumption. *Control Engineering Practice*, 17:245–254, 2009. 2, 6, 33

[26] A. H. Jazwinski. *Stochastic Process and Filtering Theory.* Academic Press, New York, New York, 1970. 90

[27] Valerie H. Johnson, Keith B. Wipke, and David J. Rausen. Hev control strategy for real-time optimization of fuel economy and emissions. Technical Report 2000-01-1543, SAE, 2000. 60

[28] S. Julier. The spherical simplex unscented transformation. In *Proceedings of the American Control Conference*, Denver, 2003. 92

[29] S. Julier and J. Uhlmann. A general method for approximating nonlinear transformations of probability distribution. *http://www.robots.ox.ac.uk/ siju*, 1994. 92

[30] S. Julier and J. Uhlmann. A new extension of the kalman filter to nonlinear systems. volume 3068, pages 182–193, 1997. 90, 92

[31] S. J. Julier, J. K. Uhlmann, and H.F. Durrant-Whyte. A new approach for the nonlinear transformation of means and covariances in linear filters. *IEEE Transactions on Automatic Control*, 1996. 92

[32] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960. 90

[33] B. Mourllion, D. Gruyer, A. Lambert, and S. Glaser. Kalman filters predictive steps comparison for vehicle localization. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005. 35, 90

[34] NASA. SRTM, 2010. Available from: `http://www2.jpl.nasa.gov/srtm/` [cited 2010-02-08]. 29

[35] J. Nocedal and S.J. Wright. *Numerical Optimization*. Springer, Heidelberg, 1999. 116

[36] M. Nørgaard, N.K. Poulsen, and O. Ravn. Advances in derivative-free state estimation for nonlinear systems. Technical Report (IMM-REP-1998-15), Technical University of Denmark, April 2000. 90

[37] L.S. Pontryagin. *The Mathematical Theory of Optimal Processes*. Interscience, 1962. 67

[38] PSE. *gPROMS Advanced User's Manual*. London, 2000. 116

[39] G. Rousseau. *Véhicule Hybride et Commande Optimale*. PhD thesis, ENS des Mines de Paris, 2008. 2, 6, 33

[40] J.M. Sanz-Serna. Symplectic integrators for hamiltonian problems: an overview. *Acta Numerica*, pages 243–286, 1991. 65

[41] S. Sarkka. On unscented kalman filtering for state estimation of continuous-time nonlinear systems. *IEEE Transactions on Automatic Control*, October 2007. 93, 95

[42] J. Schroeder, U. Mueller, and R. Dillmann. Smart roadster project: Setting up drive-by-wire or how to remote-control your car. 135

[43] Kenneth Sebesta, Eric Busvelle, and Juergen Sachau. On mapping engine efficiency through normal driving. *submitted for publication*, 2010. 6

[44] Kenneth D. Sebesta, 2010. Available from: `http://www.eissq.com/ponthy`. 21

[45] Kenneth D. Sebesta, Nicolas Boizot, Eric Busvelle, and Juergen Sachau. Using an adaptive high-gain extended kalman filter with a car efficiency model. In *ASME Digital Systems Control Conference*, 2010. 36, 78, 90

[46] Hector J. Sussmann. A strong version of the maximum principle under weak hypotheses. In *Proceedings of the 33rd IEEE CDC*, Orlando, FL, December 1994. 2

[47] D. Titterton and J. Weston. *Strapdown Inertial Navigation Technology*. The Institution of Electrical Engineers, 2004. 27

[48] Google Earth Toolbox [online]. 2010. Available from: `http://code.google.com/p/googleearthtoolbox`. 148

[49] E. Trélat. *Contrôle Optimal: théorie et applications*. Vuibert, 2005. 108, 110, 112

[50] R. van der Merwe and E. Wan. The square-root unscented kalman filter for state and parameter estimation. 93, 95

[51] T. van Keulen, B. de Jager, A. Serrarens, and M. Steinbuch. Optimal energy management in hybrid electric trucks using route information. 2009. 2, 33

[52] P. Wasmeier. Geodetic transformation [online]. 2010. 8

[53] Xi Wei. *Modeling and control of a hybrid drivetrain for optimum fuel economy*. PhD thesis, The Ohio State University, 2004. 6

ABSTRACT

Title of dissertation:	Optimal Observers and Optimal Control: Improving Car Efficiency with Kalman and Pontryagin
	Kenneth D. Sebesta, Doctor of Philosophy, 2010
Dissertation directed by:	Professor Juergen Sachau University of Luxembourg Systems and Controls Engineering
	Professor Eric Busvelle Université de Bourgogne LE2I

The PhD presents a combined approach to improving individual car efficiency. An optimal observer, the Extended Kalman Filter, is used to create an efficiency model for the car. Particular attention was paid to handling the asynchronous and redundant nature of the measurement data.

A low-cost sensor suite developed to measure data is described. This sensor suite was installed on multiple vehicles to good success. It employs an accelerometer, gps, fuel injector timer, and Vss input to measure all the data necessary to reconstruct the car's state. This observer and sensor suite can be used as the base for any study which requires car efficiency maps, allowing research to proceed without manufacturer supplied data.

Once the efficiency map is found, it is then curve-fitted in order to reduce model complexity. The simplified model is then used as a basis for optimal control through Pontryagin's Maximum Principle.

Real-world test results are given, both for efficiency mapping, and for optimal control. Detailed discussion of the observer and controller is presented, in order to ease understanding and save implementation time.