Pacific Institute for the Mathematical Sciences

IMA Institute for Mathematics and its Applications

Proceedings of the Sixth

PIMS-IMA Graduate Industrial Math Modelling Camp

May 17–22, 2003

Banff International Research Station

Co-sponsored by:

The Natural Science and Engineering Research Council of Canada

The Alberta Science and Research Authority

and

The US National Science Foundation

Editor: Rachel Kuske, University of British Columbia


NSERC
CRSNG


Alberta
INNOVATION AND SCIENCE


NSF

FOREWORD BY THE PIMS DIRECTOR

The annual PIMS **Graduate Industrial Math Modelling Camp (GIMMC)** is held in one of the PIMS universities as part of the PIMS Industrial Forum. It is part of PIMS' commitment to providing training for young mathematical scientists who are either pursing careers in academia or in industry.

The goal of the GIMMC is to provide experience in the use of mathematical modelling as a problem solving tool for graduate students in mathematics, applied mathematics, statistics and computer science. In addition to this it helps prepare them for the **Industrial Problem Solving Workshop** which is the other component of the PIMS Industrial Forum.

At the workshop students work together in teams, under the supervision of invited mentors. Each mentor poses a problem arising from an industrial or engineering application and guides his or her team of graduate students through a modelling phase to a resolution.

The Sixth GIMMC was held May 17–22, 2003, at the Banff International Research Station (BIRS) in Alberta. This year it was co-sponsored by the Institute for Mathematics and its Applications (IMA).

There were six problems posed, with a total of 33 students in attendance. The students came from all across North America with 17 from the United States.

My sincere appreciation and gratitude goes to all the people involved in this workshop, in particular I wish to thank the organisers (Rachel Kuske, Fadil Santosa, Jack Macki, Chris Bose, Huaxiong Huang, Ian Frigaard) and mentors (Emily Stone, Richard Braun, Sonja Glavaski, David Misemer, Fadil Santosa, Robert Piché).

I look forward to the 2004 GIMMC which will be held at the University of Victoria.

Dr. Nassif Ghoussoub, Director
Pacific Institute for the Mathematical Sciences

EDITOR'S PREFACE

From May 17 through May 22, 2003, thirty-three graduate students and six mentors gathered at the Banff International Research Station (BIRS) for the PIMS-IMA Graduate Industrial Mathematical Modelling Camp. Through collaboration with the Institute for Mathematics and its Applications (IMA) in Minneapolis, this year brought a higher proportion of students from US institutions together with their Canadian counterparts. It was exciting to be in the new facility in Banff and to be part of the first GIMMC workshop co-sponsored with the IMA. Since the space is limited at BIRS, the size of the camp was smaller than previous years, and competition for participation was tough. Through GIMMC and the Industrial Problem Solving Workshop, held the following week at University of Calgary and also co-sponsored by the IMA, the group had excellent opportunities to learn about math modelling in industry on both sides of the border.

Following the typical BIRS workshop schedule, the participants arrived on Saturday and settled in, with the camp officially starting on Sunday morning. Due to a power outage during the first of the presentations by the mentors, the format of the camp was slightly altered this year. During the first morning of the camp, the students broke into their teams and started getting their feet wet with the problems. This did not have an impact on the selection of the teams, since the teams were decided before the camp began. When the mentors presented the problems after the power had returned, the groups had already started to get into their work.

Again this year we had a stellar group of mentors, demonstrating their excellence well beyond the activities of the camp. They calmly improvised when necessary and got their students rolling right away. It takes a considerable amount of judgement, skill and effort to come up with good problems for the camp and lead the groups through them under a compressed schedule. From Sunday through Wednesday evening (late night?), the individual workshop groups met under the guidance of these mentors. Typical activities during this period included short lectures on background material from the mentors, digging into background literature, pencil and paper calculations, and work on the computers. By Thursday morning the groups were ready to present their findings. They gave their formal write-ups shortly after the workshop, which appear more or less as received in the rest of this document. While this may all sound fairly straightforward, in practice it is an extremely intense week for all concerned—students, mentors and organisers. The best way to appreciate this is to look at a few of the chapters which follow, keeping in mind that they represent the work of perhaps 5–6 graduate students having varied mathematical backgrounds, working in groups with some guidance from the academic mentor over a period of three and a half days.

A workshop of this size can only be successful with the effort and skill of numerous personalities both on stage and behind the scenes. Firstly, let me thank the mentors, without whom their would be no GIMMC. They were:

- Emily Stone (Utah State University, PCR Analysis)
- Richard Braun (University of Delaware, Draining of Thin Films)
- Sonja Glavaski (Honeywell, Control of Hybrid Systems)
- David Misemer (3M, Polymer Purification)
- Fadil Santosa (IMA/University of Minnesota, Solar Car Racing)
- Robert Piché (Tampere University of Technology, Finland, Machine Tool Measurements)

Some of these names will be familiar to those who have been following past PIMS industrial programmes and industrial problem solving workshops in other places, while some were first timers. All of the mentors did outstanding work both leading up to and during the week; and we can not thank them enough for their efforts. Through all their hard work, we believe they enjoyed their mathematically stimulating week with the energetic students teams.

Secondly, I'd like to thank the staff at BIRS, PIMS and the IMA, who helped in all of the organisation and behind the scenes work. At BIRS the Station Manager, Andrea Lundquist, and the Computer Systems Administrator, Brent Kearney, were invaluable. This was the first workshop of this type hosted at BIRS, and they were

always helpful, going above and beyond their usual duties to make sure the teams had everything they needed. Administrative and technical matters at PIMS were, as usual, expertly and cheerfully handled by Derek Bideshi, Andrea Hook, Heather Jenkins, Sandy Rutherford and Kelly Choo. The IMA staff provided expert assistance in getting those from US institutions across the border, so that this collaborative effort appeared seamless.

Finally, I must thank Fadil Santosa (again); not only did he co-organise the workshop, but also volunteered as a mentor. Having organised many of these types of workshops at the IMA, he was a great source of ideas and experience, from start to finish.

Rachel Kuske, Editor
Department of Mathematics
University of British Columbia

Contents

Chapter 1

Identification of the Initial Concentration of DNA in Polymerase Chain Reaction

Participants: Emily Stone (Mentor), Katharina Baamann, Yasong Jin, Serguei Lapin, Mingfei Li, Rongsong Liu, Anuj Mubayi.

PROBLEM STATEMENT: Idaho Technology develops, manufactures, and sells the fastest, highest-quality machines for DNA analysis including DNA amplification, real-time thermocycling and SNP detection. DNA amplification is performed by PCR, a biochemical reaction that is accelerated by rapid temperature cycling through efficient heat transfer by hot air to samples contained in microcapillary tubes or thin walled microcentrifuge tubes. Modelling of the reactions of PCR would help IT optimize it's designs and potentially lead to innovation in their rapid PCR technology. The goal of this project is to understand the reactions and the engineering involved in IT's family of devices, and develop models of the biochemical reactions in the devices that can be parameterized from IT data.

1.1 Introduction

Polymerase Chain Reaction (PCR) is a powerful technique for amplification of a target sequence of DNA. The central scientific fact which makes PCR so useful is that the genetic material of each living organism, whether plant, animal, bacterium or virus, possesses sequences of its nucleotide building blocks (usually DNA, sometimes RNA) that are unique and specific to itself. Indeed, complex organisms such as human beings possess unique DNA sequences that define each particular individual. These unique variations make it possible to trace genetic material back to its origin and identify with precision what kind of organism it is. It often can be determined if a DNA sample is a member of a particular species. However, such an investigation requires that enough DNA be available for the analysis, which is where PCR comes into play. PCR also has applications in pathogen detection and disease diagnosis, both of which may require that the initial amount of DNA present in the sample be known, hence the term "quantitative" PCR. Idaho Technology, a company based in Salt Lake City, Utah, manufactures devices which automate quantitative PCR to such an extent that non-medical personnel can operate them in non-laboratory environments.

PCR requires a template molecule (the DNA sample to be copied) and two primer molecules to begin the copying process. The primers are short chains of four different base pairs that can select a given sequence of genetic material. The four component base pairs are like bricks or building blocks that are used to construct genetic molecules.

The three steps of the PCR reaction are (Fig. 1.1):

1. **Denaturation** - the DNA is heated to 95^oC to render it single-stranded.
2. **Annealing** - the two primers bind to the two ends of the appropriate complementary strand; the temperature for this step depends on the of size of the primer and its homology to the target DNA.
3. **Primer Extension** - DNA polymerase extends the primer by its polymerase activity. This is done at a temperature optimal for the particular polymerase that is used; currently the most popular enzyme for this step is Taq Polymerase, the DNA polymerase from thermophilic ("heat-loving") bacteria *Thermus aquaticus*. The extension is performed at the high temperature 72^oC, which permits selective extension of the target sequence, since other sequences will not form on their own at such a high temperature.

These steps are repeated 30 to 45 times. The reaction produces copies of the DNA template essentially exponentially, and since each cycle can require up to 5 minutes, a large quantity of DNA can be produced for analysis in several hours. Idaho Technology offers devices that can complete the amplification in as little as 30 minutes.

In theory, the quantity of DNA is increasing exponentially and one can use the so-called "famous" equation to describe the process:

$$C_n = 2^n C_0, \tag{1.1}$$

where C_0 and C_n are the initial and final concentration of DNA, and n is the number of cycles. Unfortunately, the above equation works only when the PCR is performed under ideal conditions. In practice, the final concentration of DNA varies typically after performing 30 to 45 cycles, even when the initial concentrations of several DNAs are identical. Hence, the identification of the initial concentration of DNA is not straightforward from the final quantity.

The goal of our work is to introduce and test models based on differential equations that allow us to identify the initial concentration of DNA.

The paper is organized as follows: in section 1.2 we introduce differential equations that describe the PCR process. We will then derive the logistic equation from these. Sections 1.4 and 1.5 describe some other models which we studied for the identification of the initial concentration along with numerical results. We then present in section 1.6 some ideas for future work.

1.2 The ODE Model for PCR

As we described in the previous section, a cycle of PCR consists of three steps. In the denaturing step, the two strands of the DNA molecule in the solution are separated into single-stranded templates by raising the

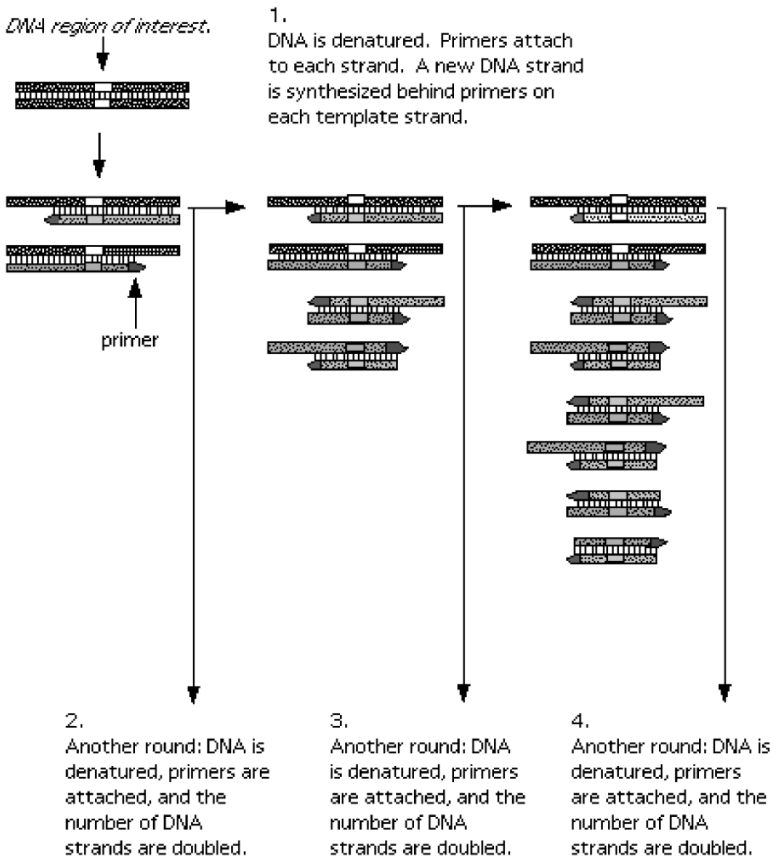

Figure 1.1: PCR process steps

temperature to 94^0C and disrupting the hydrogen bonds. In the annealing step, the solution is cooled down to approximately 60^0C to allow the primers, which are present in high concentration, to hybridize with the single stranded DNA. The primers are 20-30 nucleotides chain molecules. Once the primer-template heteroduplex is formed, it acts as the initiation complex for the DNA polymerase to recognize and bind the wanted sequence of the DNA strand. This step is crucial for the specificity of the amplification: only those molecules that have sequences complementary to the primers will be amplified. The last step is a polymerization reaction in which the solution is heated again to 70^0C, which is the optimal temperature for DNA Taq polymerase. This enzyme catalyses the binding of complementary nucleotides to the template in the direction that goes from the primer to the other end. Notice that if this polymerization proceeded to its end, at the end of the third step we would have twice as many DNA molecules as we had at the beginning of step 1. These three steps constitute one cycle of the PCR. The cycles are repeated a number of times (30 to 45) by varying the temperature in the solution in such a way that the DNA molecules that were synthesized in a given cycle are used as templates in the following cycle. Hence we get an extremely efficient amplification mechanism for DNA.

We will represent the three steps of a typical cycle of PCR by reaction equations:

Step1: 94^0C

$$D \overset{E_1}{\rightarrow} 2S$$

Step2: 60^0C

$$S + P \overset{E_2}{\rightarrow} S'$$

Step3: 70^0C

$$S' + R \overset{E_3}{\rightarrow} D$$

where
D: double stranded DNA
S: single stranded DNA
$S^{'}$: primed single stranded DNA
R: resources
P: primer
E_1: efficiency of the first reaction
E_2: efficiency of the second reaction
E_3: efficiency of the third reaction.

Now we introduce the law of mass action, which is the fundamental "law" of a chemical reaction and which describes the rate at which chemicals, whether large macromolecules or simple ions, collide and interact to form different chemical combinations. Suppose that two chemicals, say A and B, react upon collision with each other to form a product C,

$$A + B \xrightarrow{E} C.$$

The rate of this reaction is the rate of accumulation of the product $\frac{d[C]}{dt}$. This rate is the product of the number of collisions per unit time between the two reactants, and the probability that a collision is sufficiently energetic to overcome the free energy of activation of the reaction. The number of collisions per unit time is taken to be proportional to the product of the concentrations of A and B with a factor of proportionality that depends on the geometrical shapes of the reactant molecules and on the temperature of the mixture. Combining these factors we have

$$\frac{d[C]}{dt} = k[A][B]. \tag{1.2}$$

The identification of the above equation and the reaction is called the *law of mass action*, and the constant k is called the rate constant for the reaction.

According to the law of mass action, we will now derive five ordinary differential equations based on the above PCR reactions. Here we do not consider the reverse reactions.

$$\frac{dp}{dt} = -k_2 sp \tag{1.3}$$

$$\frac{dr}{dt} = -k_3 r s^{'} \tag{1.4}$$

$$\frac{ds}{dt} = -k_2 sp + 2k_1 d \tag{1.5}$$

$$\frac{ds^{'}}{dt} = k_2 sp - k_3 r s^{'} \tag{1.6}$$

$$\frac{dd}{dt} = -k_1 s^2 + k_3 r s^{'} \tag{1.7}$$

where
p=[P]: concentration of the primer in the solution
r=[R]: concentration of the resource in the solution
s=[S]: concentration of single stranded DNA in the solution
s'=[S']: concentration of primed single stranded DNA in the solution
d=[D]: concentration of the double stranded DNA in the solution
k_1: rate constant for the first reaction
k_2: rate constant for the second reaction
k_3: rate constant for the third reaction

These differential equations for p,r,s,s' and d can be approximated by maps, which simplify the calculations. For equation (1.3) we approximate dp with $\Delta p = p_{n+1} - p_n$. After multiplying the equation with Δt, we get the map:

$$p_{n+1} = p_n - k_2 \Delta t s_n p_n \tag{1.8}$$

By using the same method, we can derive all the maps for the problem:

$$r_{n+1} = r_n - k_3 \Delta t s'_n r_n \tag{1.9}$$

$$s_{n+1} = s_n - k_2 \Delta t s_n p_n + 2k_1 \Delta t d_n \tag{1.10}$$

$$s'_{n+1} = s'_n + k_2 \Delta t s_n p_n - k_3 \Delta t s'_n r_n \tag{1.11}$$

$$d_{n+1} = d_n - k_1 \Delta t s_n^2 + k_3 \Delta t s'_n r_n \tag{1.12}$$

The experimental data that we use is produced by using fluorescent enzymes that are connected to the amplified double stranded DNA. Therefore, the solution fluoresces in proportion to quantity of the dDNA. Using that information, we can generate an amplification curve for the DNA sample (Fig. 1.2), where the intensity of fluorescence is plotted verses the cycle number. In a single cycle of PCR, we use standard samples that have known initial concentrations of the DNA along with samples that have unknown concentrations.

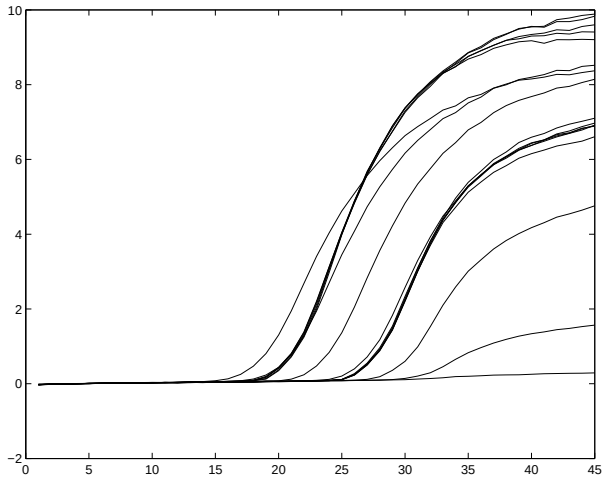

Figure 1.2: Amplification curve (x-axis: cycle number, y-axis fluorescence level).

In order to get the wanted initial value C_0, we can use the "famous" model. For that, we need to determine a reasonable Threshold T of the fluorescence level. The best Threshold is the inflection point of the sample amplification curve, since we are interested most in the exponential growing part of the amplification curve. In the amplification graph, we get then the relevant cycle values of the different standards, for which we have the initial concentration C_0. By plotting these cycle values versus the logarithm of the initial concentration values, we get a line (approximately), which is called the *standard graph*. C_0 of the unknown sample data can be computed by getting the cycle value at the inflection point of the sample data in the amplification curve and then reading C_0 from this value on the standard graph. *Mentor note: Using the inflection point of each curve for a threshold is not correct. The threshold used in this calculation should be the same for all samples in a given run*.

1.3 The Logistic Model

The logistic model can be derived form the maps in the previous section, along with some additional assumptions. First, we assume that all initial double strands are split into two single strands. This means we do not need equation (1.10), and equation (1.12) can be written:

$$d_{n+1} = d_n + k_3 \Delta t s'_n r_n \tag{1.13}$$

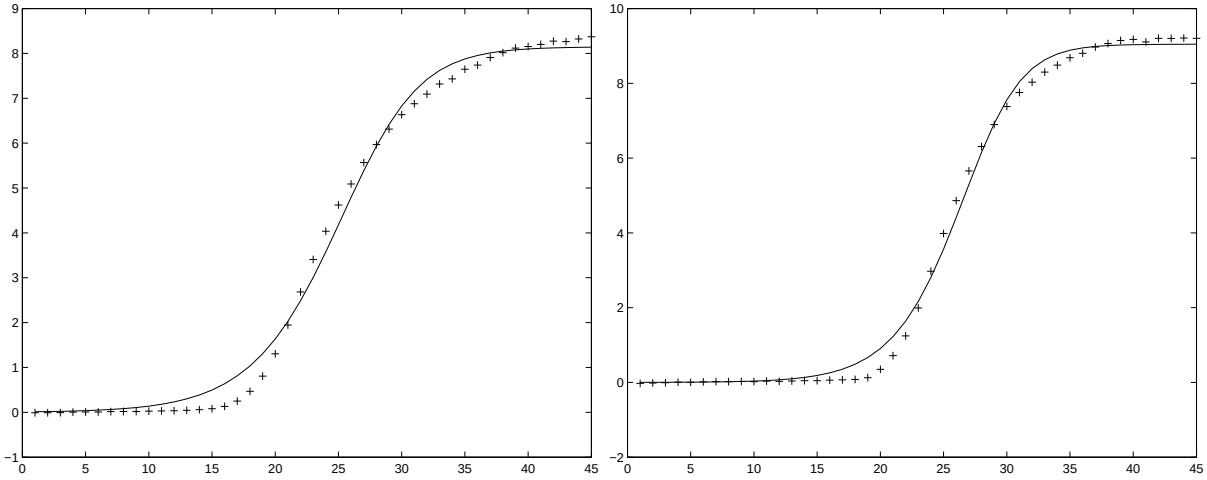

Figure 1.3: Logistic model(-) versus standard data(+) (left) and logistic model(-) versus sample data(+) (right) of data set 1 by using least square method (x-axis: cycle number, y-axis:fluorescence level).

Let's now add equation (1.9) to equation (1.12):

$$d_{n+1} + r_{n+1} = d_n + r_n. \tag{1.14}$$

This means that $d_n + r_n = K$, where K is a constant. Hence $r_n = K - d_n$, and equation (1.12) becomes:

$$d_{n+1} = d_n + k_3 \Delta t s'_n (K - d_n). \tag{1.15}$$

If we also assume that all single strands s_n are transformed into primed single strands s'_n, we see that $s'_n = 2d_n$. Then equation (1.12) becomes:

$$d_{n+1} = d_n + k_3 \Delta t 2 d_n (K - d_n). \tag{1.16}$$

which has the form of a logistic map

$$x_{n+1} = x_n + e x_n (K - x_n) \tag{1.17}$$

where e is growth rate and K is the carrying capacity. This model is not realistic since we are making strong assumptions, but it is more realistic than the "famous" model, in that it yields saturation at large copy numbers. The saturation is due to the limited amount of resource present in solution.

In order to get the initial value of the concentration, we must first find the parameters $e = k_3 \Delta t$ and K of this equation. We present two ways to do this and compare the results. The first approach uses the least squares method to fit the data from the reaction to the model equations. By using $e = 2k_3 \Delta t$ the equation (1.15) can be now written as:

$$d_n eK - e d_n^2 = d_{n+1} - d_n \tag{1.18}$$

This can also be written in matrix form:

$$AX = b, \tag{1.19}$$

where $A = \begin{pmatrix} d_1 & -d_1^2 \\ \vdots & \vdots \\ d_{n-1} & d_{n-1}^2 \end{pmatrix}$, $X = \begin{pmatrix} ek \\ e \end{pmatrix}$, $b = \begin{pmatrix} d_2 - d_1 \\ d_n - d_{n-1} \end{pmatrix}$ and n is the cycle number.

According to the least squares method, we can solve for X in the following manner:

$$X = (A^T A)^{-1} A^T b. \tag{1.20}$$

From here, we designate the sequence $\{y_i\}_{i=0}^n$ to represent the fluorescence level data from the PCR reactions. We designate the sequence $\{d_i\}_{i=0}^n$ to be the approximate data from our model. The matrices A and b are formed from the experimental data, and e and K are found from the least squares fit. In order to find d_0, the initial value, we minimize the function $J(d_0) = \sqrt{(\sum_{i=1}^n |d_i - y_i|^2)}$, with the found values of e and K. From the standard graph formed from the data with known initial concentrations we can compute C_0, which is the initial copy number of DNA in our target sample, for a given d_0. Tables 1.1 and 1.2 show the values of the constants e, k, the initial value d_0 and the initial DNA concentration C_0 of different data sets. The standards have known initial value, while the samples have unknown initial concentrations of DNA.

Data set	e	k	d_0	C_0
standard 1	0.0369	8.1473	0.0101	10^5
standard 2	0.0378	8.2833	0.0048	10^4
standard 3	0.0429	7.9008	0.0013	10^3
standard 4	0.00592	6.6789	$1.4725 * 10^{-4}$	10^2
standard 5	0.0914	4.5960	$2.5558 * 10^{-5}$	10
standard 6	0.2543	1.5449	$1.1349 * 10^{-5}$	1
sample 1	0.0432	9.0478	0.0013	$2.2016 * 10^3$
sample 2	0.0682	6.3617	$5.5504 * 10^{-5}$	16.3296

Table 1.1: Logistic model, dataset 1.

Data	e	k	x_0	C_0
standard 1	0.4393	1.2197	$2.9297 * 10^{-4}$	10^7
standard 4	0.4219	1.3436	$3.624 * 10^{-5}$	10^6
standard 7	0.4249	1.3670	$4.3869 * 10^{-6}$	10^5
standard 10	0.3706	1.4995	$1.7405 * 10^{-6}$	10^4
standard 13	0.4293	1.3553	$1.9073 * 10^{-7}$	10^3
standard 16	0.4689	1.4026	$7.1013 * 10^{-9}$	10^2

Table 1.2: Logistic model, dataset 2.

The second approach uses a non-linear optimization method. We minimize the error between the experimental data y and the model data d in order to get the best constants. With this method it is also possible to get e, k and the initial concentration C_0 simultaneously. To this end, Matlab provides a function called *fminsearch*, which minimizes a given function that depends on a finite number of variables. After initializing d_1, we get all the d_i (which depend on e and k) by iteration over n. The error function is:

$$J(e, k, d_0) = \sum_{i=1}^{N} (y_i - d_i(e, k, d_0))^2$$

where the y_i's are the normed experimental data-points and the d_i are our generated model values. If we now use J and an initial guess of e and k for the Matlab function *fminsearch*, then it will minimize the error in order to

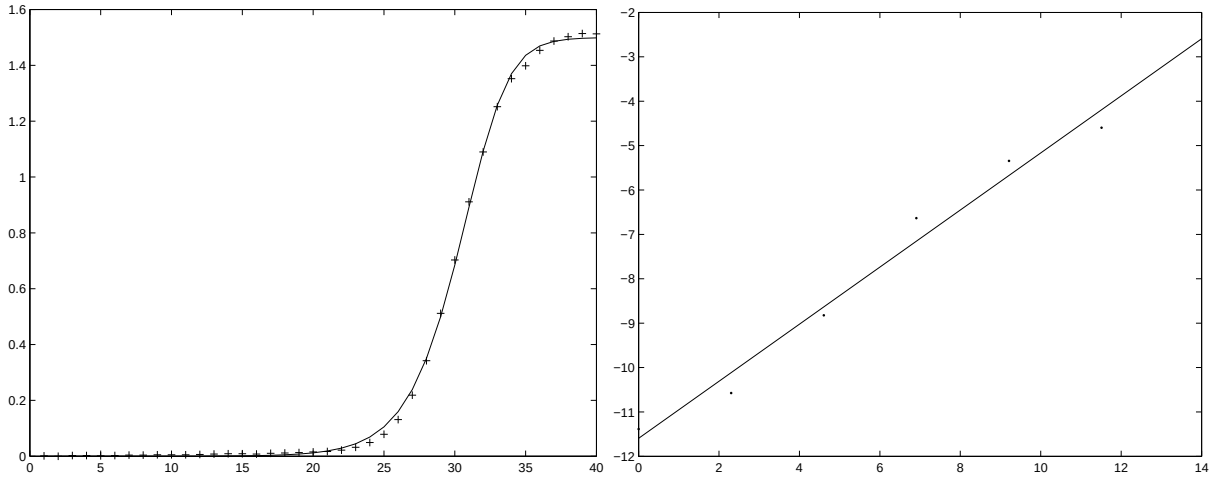

Figure 1.4: Left: logistic model(-) versus standard data(+) of sample run from data set 2 by using least squares method (x-axis: cycle number, y-axis: fluorescence level). Right: standard line mapping initial fluorescence to initial copy number (x-axis: log of C_0, y-axis: log of initial fluorescence level), using known standards.

Data	e	k	x_0
data 1 sample 1	0.0458	8.9841	0.0013
data 2 standard 1	0.5260	1.2097	0.0001
data 2 standard 4	0.4416	1.3426	0.00001
data 2 standard 10	0.3753	1.4954	10^{-6}

Table 1.3: Logistic model

get e and k. We've performed calculations for several datasets and plotted the result versus the experimental data (Fig. 1.3–1.4). Table 1.3 shows the generated values of e, k and x_0 by using the described optimization method.

It is also possible to have *fminsearch* find the initial value x_0, as well as e and k. The results for this optimization are shown in Fig. 1.5.

1.3.1 Analysis of the Logistic Model

We can estimate the range of the parameter of the logistic equation by doing some analysis on the equation:

$$d_{n+1} = d_n + 2e_1 k d_n (k - d_n/k) \tag{1.21}$$

where $k = r_n + d_n$.

The two equilibria of this discrete equation are $x = 0$ and $x = k$. We first consider case $x = 0$. Let us introduce $\alpha = 2e_1 k$, then one can show that the equilibrium $x = 0$ is asymptotically stable for $-2 < \alpha < 0$. Since $e_1 > 0$ and $k > 0$, therefore α has to be greater than 0. Hence, $x = 0$ is unstable.

Now, the equilibrium $x = k$ is asymptotically stable for $0 < \alpha < 2$ (i.e. for $0 < e_1 k < 1$). So for $0 < e_1 k < 1$ every solution with initial condition close to k, tends to the equilibrium k as n increases. If $\alpha > 2$, i.e. $e_1 k > k$, then equilibrium $x = k$ is unstable and there is no asymptotically stable equilibrium to which the solution tends. *Note: If $e_1 k > 1$, i.e. $\alpha > 2$ then period doubling occurs.* If $2 < \alpha < 2.449$, there is a solution of period 2 to which every solution $x_{n+1} = f(x_n)$ tends. If $\alpha > \sqrt{6}$ the solution of period 2 is unstable but it can be shown that a solution of period 4 appears and it is asymptotically stable $\sqrt{6} < \alpha < 2.544$. This period doubling phenomenon continues until $\alpha = 2.570$ when periodic solutions whose periods are not powers of 2 begins to appear but these solutions are unstable. The numerical results (i.e. the values of e_1 and k for different samples and standards) verify this analysis.

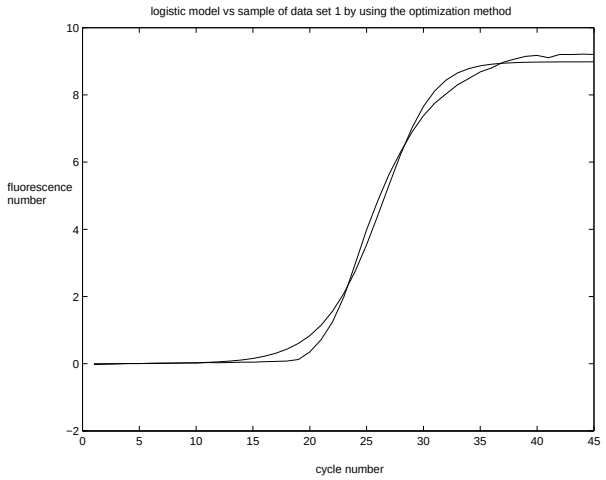

Figure 1.5: Logistic model versus sample data of data set 1 by using optimization method (x-axis: cycle number, y-axis: fluorescence level).

1.4 The Taq Model

This model modifies the equations (1.9)–(1.12) by assuming that the efficiency of the Taq enzyme decreases as the copy number of DNA to replicated grows very large. Instead of a growth rate e in the logistic equation we use a modified growth rate, $ef(d)$, where the function f decreases from 1 with increasing d. This results in the following variation on the logistic equation:

$$d_{n+1} = d_n + ef(d_n)d_n(k - d_n).$$

The exact form of f is not known, so we examined two different cases:

- When reaction starts the efficiency of reaction of Taq decrease very sharply because the concentration of Taq is decreasing. We use the function $e\frac{1}{1+d}$ to describe this phenomenon, where e is a constant.
- When reaction starts the efficiency of reaction of Taq keeps some kind of high rate or only decrease slowly, then after a while it decrease sharply. We use the function $e\frac{\frac{(\arctan(3-2*d)}{500}+\pi/2}{1.6}$, where e is a constant (Fig 1.6).
- *Mentor's note: this one appears to be more physical.*

In order to get the parameters e and k and the initial copy number C_0, we can use the same methods as described in the previous section. We found the best results using the first function, and these are presented in figure 1.7 and table 1.4. The second function, which should be more realistic, however was not designed to decrease the efficiency very significantly (the lowest value is 0.9795, very close to the starting value of 1). More tests would have to be done with a better range on the arctangent function before it can be compared to the other function.

1.4.1 Analysis of the Taq Model

We analyze here the Taq model, put in a more standard form with x replacing d:

$$x_{n+1} = x_n + e\frac{1}{1 + x_n}x_n(k - x_n)$$

.

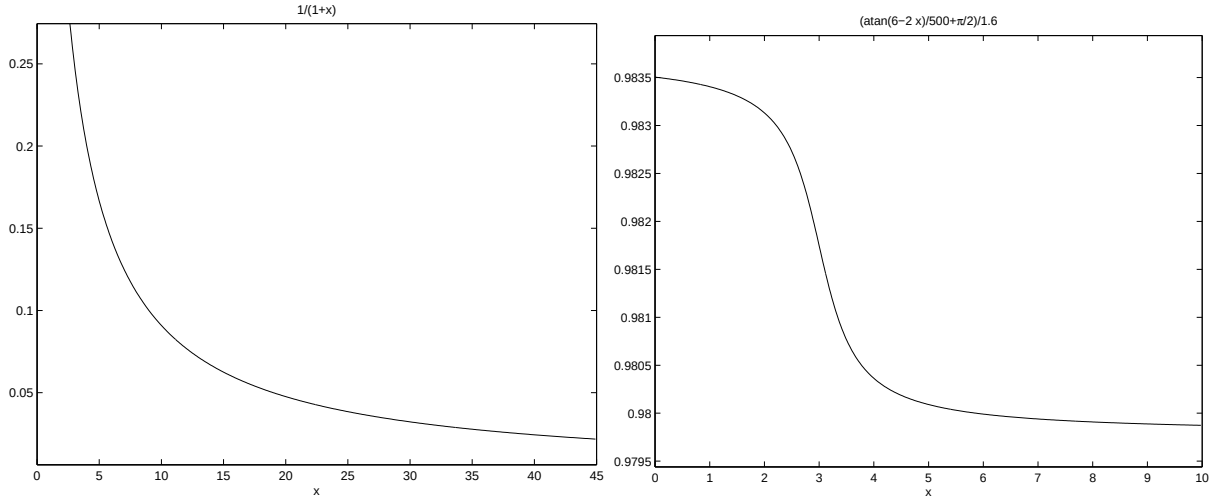

Figure 1.6: Left: $y = \frac{1}{1+x}$. Right: $y = \frac{\frac{(\arctan(3-2*x)}{500}+\pi/2}{1.6}$

Data	e	k	x_0	C_0
standard 1	0.1548	8.8321	$1.335 * 10^{-7}$	10^5
standard 2	0.15555	8.8321	$1.601875 * 10^{-8}$	10^4
standard 3	0.1641	8.5218	$1.0523947 * 10^{-9}$	10^3
standard 4	0.1914	7.2555	$7.1596 * 10^{-11}$	10^2
sample 1	0.1885	9.49	$4.4 * 10^{-10}$	$4.7 * 10^2$

Table 1.4: Taq model.

This map has two equilibria, namely $x = 0$ and $x = k$. If $-2 < ek < 0$ then equilibrium $x = 0$ is asymptotically stable, but this is not physically possible since e and k are greater than zero. Hence $x = 0$ is unstable equilibrium. If $0 < \frac{ek}{1+k} < 2$ then $x = k$ is asymptotically stable and it will be unstable for $2 < \frac{ek}{1+k}$. This means that every solution with the initial value x_0 close enough to k remains close to k and tends to k as n goes to infinity, provided $0 < e < \frac{2(1+k)}{k}$.

1.5 A More Realistic Model

In order to investigate the dynamics of the PCR process, we analyzed more complicated models. In the next model we only used the assumption that all the initial double strands split into the single strands, $s_n = 2d_n$. Substituting that into the equations (1.9)–(1.12), we get the following four maps:

$$\begin{aligned} p_n &= p_{n-1} - e_1 2 d_{n-1} p_{n-1} \\ s'_n &= s'_{n-1} + e_1 2 d_{n-1} p_{n-1} - e_2 s'_{n-1} r_{n-1} \\ r_n &= r_{n-1} - e_2 s'_{n-1} r_{n-1} \\ d_n &= d_{n-1} + e_2 s'_{n-1} r_{n-1}, \end{aligned} \tag{1.22}$$

Assuming normalized variables, we use the following initial values for p, s' and r:

$$p_1 = 1.0;\ s'_1 = 0.01;\ r_1 = 1.0.$$

Now the solution to (1.22) depends on the three parameters e_1, e_2 and d_0. As done before for the logistic model, we use an optimization approach to determine simultaneously values of e_1, e_2 and initial concentration

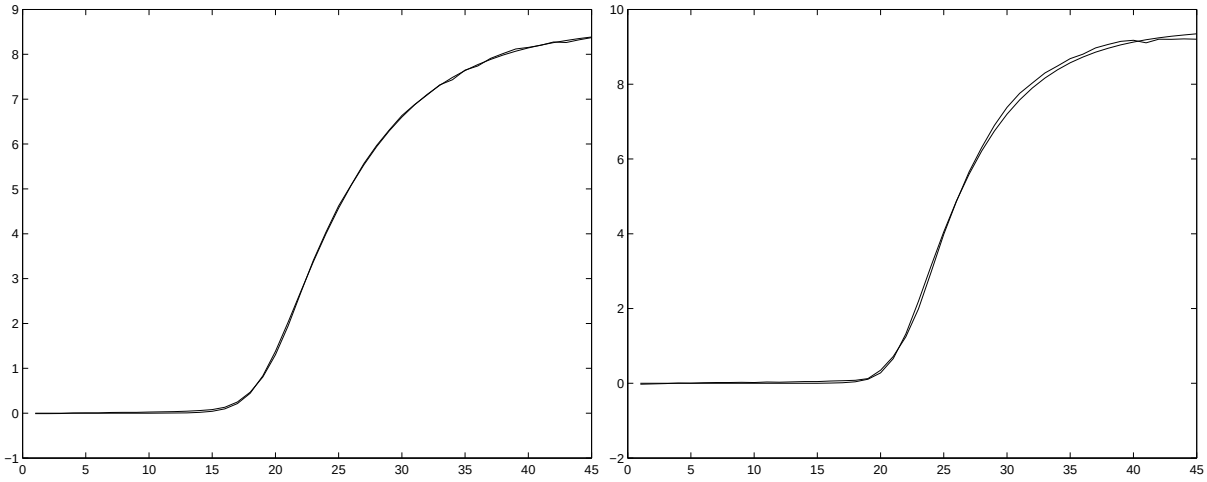

Figure 1.7: Taq model versus standard data (left) and taq model versus sample data of data set 1(right) (x-axis: cycle number, y-axis: fluorescence level).

d_0. In order to do that, we use the experimental data y and generate the following discrete objective function

$$J(e_1, e_2, d_0) = ||d(e_1, e_2, d_0) - y||_2. \tag{1.23}$$

We then find min J in order to minimize the difference between the sequences d_i and y_i, using the same Matlab function as in section 1.3. Some values of these constants are presented in table 1.5 and graphs are presented in Fig. 1.8.

Data	e_1	e_2	d_0
data 1 sample 1	0.2970	1.4195	-0.0075
data 1 sample 2	0.4533	0.8769	-0.0001
data 2 standard 1	0.8191	1.0171	-0.0054
data 2 standard 8	0.4655	1.4219	-0.0070
data 2 standard 10	0.4436	0.9273	-0.0063

Table 1.5: Realistic model

Mentor's note: obviously there is a problem since d_0 should not be negative!

1.6 Future Work

Our map model in the previous chapters consists of the system

$$\begin{cases} d' = -k_2 s^2 + k_3 ir \\ s' = -k_2 sp + 2k_1 d \\ i' = k_2 sp - k_3 ir \\ p' = -k_2 sp \\ r' = -k_3 ri \end{cases} \tag{1.24}$$

We can do some local qualitative analysis of equations (1.24) and estimate the parameters. In particular, we show below that the solutions of the above system will be bounded for a finite number of cycles. From the "famous" model, the number of double stranded DNA at the n-th cycle will be given as $d_n = 2^n d_0$ (d_0 is the initial concentration of dsDNA). Hence, for the above system at every cycle the maximum concentration of dsDNA we

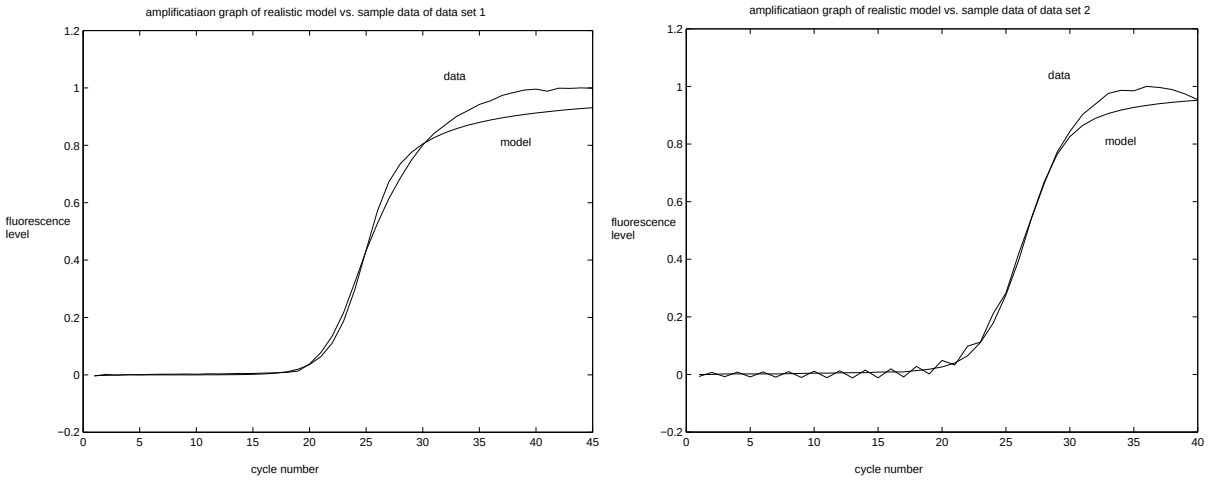

Figure 1.8: Left: realistic model versus sample data of data set 1. Right: realistic model versus sample data of data set 2.

can have is $d_n \leq 2^n d_0$. For a particular sample, from the beginning of the DNA amplification process we know, for how many cycles (say n_0) we have, so $d_n \leq 2^n d_0 \leq 2^{n_0} d_0$ for all $n \leq n_0$. Now in an ideal case at every step one dsDNA is splitting into 2 ssDNA, therefore $s_n \leq 2^{n_0+1} d_0$ for all $n \leq n_0$ and similarly $i_n \leq 2^{n_0+1} d_0$ for all $n \leq n_0$, $p_n \leq 2^{n_0+1} d_0$ for all $n \leq n_0$, and $r_n \leq 2^{n_0+1} d_0$ for all $n \leq n_0$. Hence all the solutions of the above system will be bounded.

The above system can be made more realistic by taking following factors into consideration:

- Since in these reactions temperature is playing an important role we can introduce temperature dependent rates.
- These reactions are not ideal, therefore we can also consider the reverse processes in the first two reactions.

There are several possible factors which can affect reactions and thus modify our model:

- Denaturation efficiency will decrease as the concentration of the DNA product increases, because of the increase of the melting point.
- Thermal inactivation of the DNA polymerase may affect the amplification of the sequence, especially if they differ in length.

In conclusion, in this report we analyzed a set of dynamic models for the amplification of DNA during PCR, and estimated parameters by using data sets from real PCR runs. Both linear and nonlinear least squares methods were used in the parameter estimation, with varying results. There is much room for future work in both reaction modelling and parameter estimation for this problem, which can ultimately aid in process design for the PCR reaction.

Bibliography

[1] Freeman W.M., Walker S.J., Vrana K.E. *Quantitative RT-PCR: Pitfalls and potentials*, BioTechniques 26:112–125 (January 1999).

[2] Raeymaekers Luc *General Principles of Quantitative PCR*, Methods in Molecular Medicine, Vol 26, Quantitative PCR Protocols.

[3] Hsu J.T., Das S., Mohapatra S. *Polymerase Chain Reaction Engineering*, Biotechnology and Bioengineering, Vol.55, No.2, pp.359–366, 1997.

[4] Karp R.M. *Mathematical Challenges from Genomics and Molecular Biology*, Notices of the AMS, Vol.49, pp.544–553.

[5] Edelstein-Keshet L., *Mathematical Models in Biology*, Random House, New York, 1988.

[6] Newton C.R., Graham A. *PCR*, Springer-Verlag, New York, 1997.

[7] Gustavo Stolovotzky and Guillermo Cecchi *Efficiency of DNA replication in the polymerase chain reaction.* Proc. Natl.Acad. Sci. USA, Vol. 93, pp. 12947-12952, Nov. 1996.

Chapter 2

Vertical Draining of Thin Films: Two Fluids Case

Participants: Richard Braun (Mentor), Olivier Dubois, Marcio Gameiro, Andrew Taylor, Haris Widjaya, Kerianne Yewchuk.

PROBLEM STATEMENT: The Dow Corning corporation uses interferometry to measure the decay of the boundary in a thin fluid, which determines the quality of the surfactants the company produces. A mathematical model that describes the evolution of the thin film boundary will make quality control more efficient. The Navier-Stokes equations are a commonly accepted mathematical model for fluid dynamics. Using information about the measurement method, we can simplify the Navier-Stokes equations. We can then analyze the equations using analytical and numerical methods to get a description of the thin film boundary decay rate. This method has been applied to single fluid thin film measurements. We want to analyze the two fluid case in this report.

2.1 Introduction

Thin films are a special case in fluid dynamics. There are two different generic geometries considered:

- Thin liquid film on a solid substrate and
- Free films which are bounded by a liquid or gas.

There are a variety of phenomena we can observe, such as drainage, details of rupture, or non-Newtonian surface properties [2].

These observed phenomena can help to characterize and describe physical processes that occur in the real world. Such knowledge can be used in the following areas [3]:

- Surface coatings in paint, varnish, protective wax, and silver layers on a CD, and
- Foam development.

Foam is a complex fluids system that, in general, is very difficult to model. However, regions where the bubbles meet are basically thin films draining downwards. (This is why soap bubbles do not stay as bubbles.) Understanding the drainage of thin films is important in foam fabrication, where the properties of foam products are determined by the draining rate of the thin films into the plateau borders. Knowledge of the optimal draining rate is important for producing foams economically. During foam fabrication, surfactants may be added to attain an optimal draining rate. By using an interferometry method, researchers at Dow Corning have been able to measure the rate at which a fluid drains within a film. This method can be used to perform quality inspection of the surfactants that are added into the foam products. However, this procedure is time consuming, which makes it impractical for high volume industrial application. A mathematical model based on a single fluid film that obtains an accurate model of the film boundary decay over time is developed by Braun *et al* [2]. This is the second geometry case for a single fluid.

We are interested in modelling the second general geometry introduced above, where a draining thin film is formed between two liquids. The same simplification methods found in [2] are used to obtain simpler equations that describe the boundary of the two fluids. We apply the method of characteristics to obtain exact solutions for the inner boundary and we used similarity solutions to get an approximation for the outer film boundary. Our approximation for the outer boundary equation is verified against the numerical results that we obtained.

For the more complex case when the interface between two liquids is free to move, we use similarity solutions to approximate the partial differential equation governing the two boundary conditions. We report the results of our numerical solutions in section 2.3 below.

2.2 One Fluid Case

In this section we outline the one fluid case, which was done in [2]. This case can be idealized by the schematic representation shown in Figure 2.1.

The Navier-Stokes equations are used to model the velocity of the fluids inside the thin films. Using the fact that

$$\frac{d}{D} = \delta^3 \ll 1, \tag{2.1}$$

we can make an appropriate change of variables and neglect small terms to get simplified equations for the fluid and the boundary. The final equation for the boundary is given by

$$k_t + F_z = 0, \tag{2.2}$$

where

$$F = \frac{k^3}{3}. \tag{2.3}$$

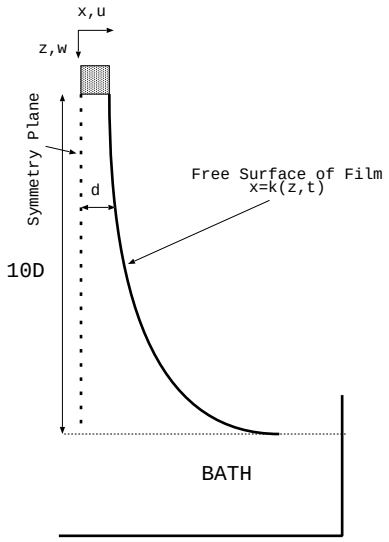

Figure 2.1: Thin film for a single fluid.

This equation can then be analytically solved using the method of characteristics, yielding the implicit solution

$$k(z,t) = k_0(z - (k(z,t))^2 t), \tag{2.4}$$

where $k_0(z) = k(z,0)$ is the initial condition. In particular, if we take $k_0(z) = sz$, we get the explicit solution

$$k(z,t) = \frac{-1 + \sqrt{1 + 4s^2tz}}{2st}, \tag{2.5}$$

which behaves asymptotically as $\sqrt{z}t^{-1/2}$. From here we can see that, in agreement with experiments, the film thickness decays as $t^{-1/2}$ for fixed z.

2.3 Two Fluid Case

After using the single film case as a starting point, we now analyze a more complicated situation with two different fluids. The model for this problem is outlined in Figure 2.2.

We have two regions in the film containing different fluids. We will refer to the inner and outer regions as fluids 1 and 2, respectively. From experimental evidence, we assume symmetry in the x direction and that the problem can be adequately represented by two dimensions. The velocity of the fluids will be written as

$$\vec{u}^{(i)}(x,z,t) = (u^{(i)}, w^{(i)}), \tag{2.6}$$

where $i = 1$ or 2 depending on the fluid in question. The functions describing the free boundaries are given by $x = k^{(i)}(z,t)$. The governing equations for incompressible fluid flow are the well-known Navier-Stokes equations (see [1])

$$\rho^{(i)}\left(\vec{u}_t^{(i)} + \vec{u}^{(i)} \cdot \nabla \vec{u}^{(i)}\right) = -\nabla p^{(i)} + \mu^{(i)} \nabla^2 \vec{u}^{(i)} + \rho^{(i)} \vec{g}, \tag{2.7}$$

$$\nabla \cdot \vec{u}^{(i)} = 0, \tag{2.8}$$

where $\rho^{(i)}$ is the density and $\mu^{(i)}$ is the viscosity of fluid i. The first set of equations comes from conservation of momentum, while the divergence-free condition comes from conservation of mass. The equations are non-linear and thus hard to solve in general. In addition, we need to find the location of the boundary as part of the solution, which only increases the difficulty of the problem.

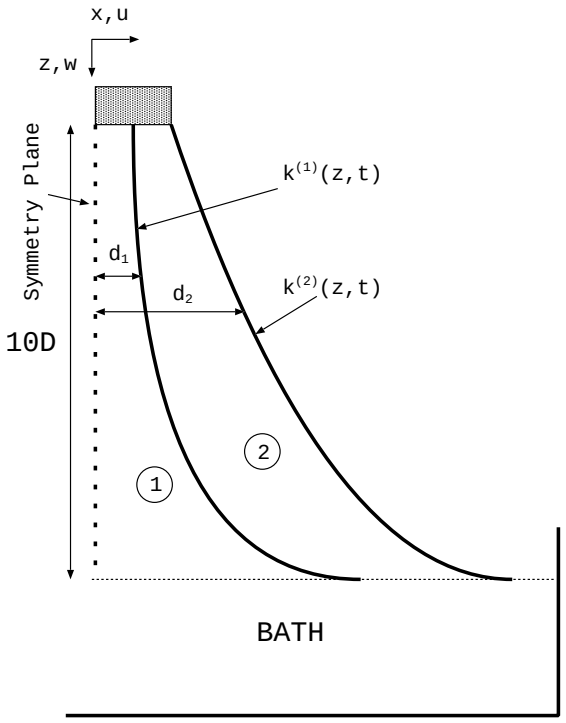

Figure 2.2: Thin film with two fluids.

2.3.1 Boundary Conditions

First, at the top of the film, where it is attached to the metal frame, we use no-slip boundary conditions $\vec{u}^{(i)} = 0$. At the symmetry line ($x = 0$), the fluid velocity should be parallel to the boundary and the vertical velocity should be an even function, leading to the boundary conditions

$$u^{(1)} = 0, \quad w_x^{(1)} = 0. \tag{2.9}$$

At the outer free boundary $x = k^{(2)}$, we choose the conditions

- $[\![\vec{n}^{(2)} \cdot \mathbf{T} \cdot \vec{n}^{(2)}]\!] = \sigma_2 \kappa^{(2)}$ (jump in normal stress),
- $\vec{\tau} \cdot \vec{u}^{(2)} = 0$ (tangential immobility),
- $k_t^{(2)} + k_z^{(2)} w^{(2)} = u^{(2)}$ (kinematic condition).

In the first condition, $\mathbf{T}$ is the stress tensor, $\vec{n}$ is the normal vector to the boundary, σ_2 is the surface tension at the fluid-air interface and $\kappa^{(2)}$ is the curvature of the boundary $k^{(2)}$. As such, this condition indicates that the jump in normal stress on the free surface is proportional to its curvature. In the second condition, we are assuming that the boundary $k^{(2)}$ cannot move along itself. Physically, this represents a limiting case when a lot of surfactant is put on the surface of the fluid, which can be thought of as a "dirty" surface. Finally, the kinematic boundary condition relates the velocity of the fluid with the movement of the boundary.

Now, for the inner free boundary $x = k^{(1)}$ between the two fluids, the conditions are similar

- $[\![\vec{n}^{(1)} \cdot \mathbf{T} \cdot \vec{n}^{(1)}]\!] = \sigma_1 \kappa^{(1)}$ (jump in normal stress),
- $k_t^{(1)} + k_z^{(1)} w^{(1)} = u^{(1)}$ (kinematic condition),

- $\vec{u}^{(1)} = \vec{u}^{(2)}$ at $x = k^{(1)}$ (continuous $\vec{u}$).

The first two conditions are analogous to the ones at the outer boundary. The third one requires the velocity to be continuous across $k^{(1)}$, since the velocity of the fluids on both sides of the boundary is dictated by the movement of $k^{(1)}$. However, the problem is not yet completely posed because one boundary condition remains to be imposed at the inner interface. We consider two relevant choices:

Case I: $k^{(1)}$ is tangentially immobile,

Case II: $k^{(1)}$ is tangentially mobile.

2.3.2 Case I: Tangential Immobility on $k^{(1)}$

In this case, we impose the boundary condition

$$\vec{\tau} \cdot \vec{u}^{(1)} \quad \text{at } x = k^{(1)}. \tag{2.10}$$

This means that the interface is not allowed to move along itself. This condition was imposed previously at the outer boundary $k^{(2)}$ and represents a limiting case when the interface is packed with surfactant.

The problem is now completely posed, as a sufficient number of boundary conditions have been specified. However, the resulting equations are non-linear and very complex, so we do not hope to find an exact solution analytically. Instead, we exploit the difference in scaling between the height of the film ($\sim$ 1 cm) and its width ($\sim$ 10 μm) to obtain simpler equations that we can solve.

We consider the case of the thin film

$$\frac{d}{D} = \delta^3 \ll 1, \tag{2.11}$$

where d and D are characteristic length scales for x and z, respectively. Scales for the other variables in the problem are as follows:

$$\begin{array}{lll} x \to \delta^3 D x & u \to \delta^3 W u & t \to \frac{D}{W} t \\ z \to D z & w \to W w & p \to \frac{W}{D\delta^6} \mu p \\ & k \to \delta^3 D k & \end{array} \tag{2.12}$$

Here $W = \frac{\rho g d^2}{\mu}$ is the characteristic velocity scale in which gravitational and viscous forces balance. Assuming that δ is a very small parameter, and writing the solution as a perturbation series in δ^3, then we can find very simple equations for the leading order terms by basically setting $\delta = 0$. Setting $\mu := \frac{\mu^{(1)}}{\mu^{(2)}}$ and $\rho := \frac{\rho^{(1)}}{\rho^{(2)}}$, the Navier-Stokes equations reduce to

$$\begin{array}{ll} p_x^{(1)} = 0, & w_{xx}^{(1)} - p_z^{(1)} + 1 = 0, \\ p_x^{(2)} = 0, & w_{xx}^{(2)} - p_z^{(2)} + \frac{\mu}{\rho} = 0. \end{array} \tag{2.13}$$

The divergence-free condition stays the same at leading order

$$u_x^{(i)} + w_z^{(i)} = 0. \tag{2.14}$$

Some of the boundary conditions are not affected, namely

$$\begin{array}{lll} w_x^{(1)} = 0 & \text{at} & x = 0, \\ w^{(1)} = w^{(2)} & \text{at} & x = k^{(1)}. \end{array} \tag{2.15}$$

The tangential immobility conditions translate to

$$w^{(i)} = 0 \quad \text{at } x = k^{(i)}, \quad i = 1, 2. \tag{2.16}$$

Finally, the leading order terms in the normal stress conditions give

$$\begin{array}{lll} p^{(2)} & = \mu p^{(1)} & \text{at } x = k^{(1)}, \\ p^{(2)} & = 0 & \text{at } x = k^{(2)}. \end{array} \tag{2.17}$$

These new equations and boundary conditions form a much easier problem that can be solved exactly for the pressures (equal to 0 everywhere) and velocities (quadratic polynomials in x) in the fluids. Hence, using the kinematic conditions, the problem reduces to equations for the free boundaries $k^{(i)}$ only. We then obtain the non-linear system of hyperbolic partial differential equations

$$k_t^{(1)} + [F^{(1)}]_z = 0, \tag{2.18}$$

$$k_t^{(2)} + [F^{(2)}]_z = 0, \tag{2.19}$$

where the flux functions $F^{(i)}$ are given by

$$\begin{array}{lll} F^{(1)} & = & \frac{(k^{(1)})^3}{3} \\ F^{(2)} & = & \frac{1}{3}(k^{(1)})^3 - \frac{\mu}{2\rho}\left[\frac{(k^{(2)})^3-(k^{(1)})^3}{3} - (k^{(1)}+k^{(2)})(\frac{(k^{(2)})^2-(k^{(1)})^2}{2}) + k^{(1)}k^{(2)}(k^{(2)}-k^{(1)})\right] \end{array} \tag{2.20}$$

It is worthwhile to point out a few properties of the flux functions at this stage. We have a non-linear system of equations for the $k^{(i)}$s where all terms in the flux function are cubic. Moreover, the equation for $k^{(1)}$ is completely independent of $k^{(2)}$ and thus can be solved first to uncouple the system. It is very similar to Burger's equation; we can use the method of characteristic to obtain an exact solution for (2.18)

$$k^{(1)}(z,t) = k_0^{(1)}(z - (k^{(1)}(z,t))^2 t), \tag{2.21}$$

where $k_0^{(1)}(z)$ is the initial shape for $k^{(1)}$. Recall that we arrived at the same equation in the one fluid case. Using a straight line as the initial shape ($k_0^{(1)} = sz$), the behaviour of the solution for large time is given by the power law

$$k^{(1)}(z,t) \sim \sqrt{z} t^{-\frac{1}{2}}. \tag{2.22}$$

2.3.3 Numerical Results for Case I

We compute a numerical solution to the system of equations (2.18), (2.19) by means of the method of lines. Using a backward finite difference discretization for the derivative of $F^{(i)}$, we get a system of ordinary differential equations

$$\frac{\partial k_j^{(i)}}{\partial t} = -\frac{F_j^{(i)} - F_{j-1}^{(i)}}{\Delta z}. \tag{2.23}$$

The initial conditions for this system are given by the discretization of the initial shape $k_0^{(i)}$. We also assume that both free boundaries stay attached to the metal frame at $z = 0$, giving the boundary conditions

$$k^{(i)}(0,t) = 0. \tag{2.24}$$

Then, we used Matlab built-in ODE stiff solver `ode15s` to obtain a numerical solution. From these results, we generated plots of the evolution of the free boundaries with time. A specific case where the fluid is the same on both sides of the inner boundary is shown below in Figure 2.3. By fixing z and looking at the resulting horizontal slice for varying time, we observe a decay rate of approximately $-\frac{1}{2}$ for both the inner and outer boundaries, as illustrated in Figure 2.4. The slope in that figure is a finite difference approximation to the slope of the thinning curve at fixed z near the bottom of the film and near the end of the evolution.

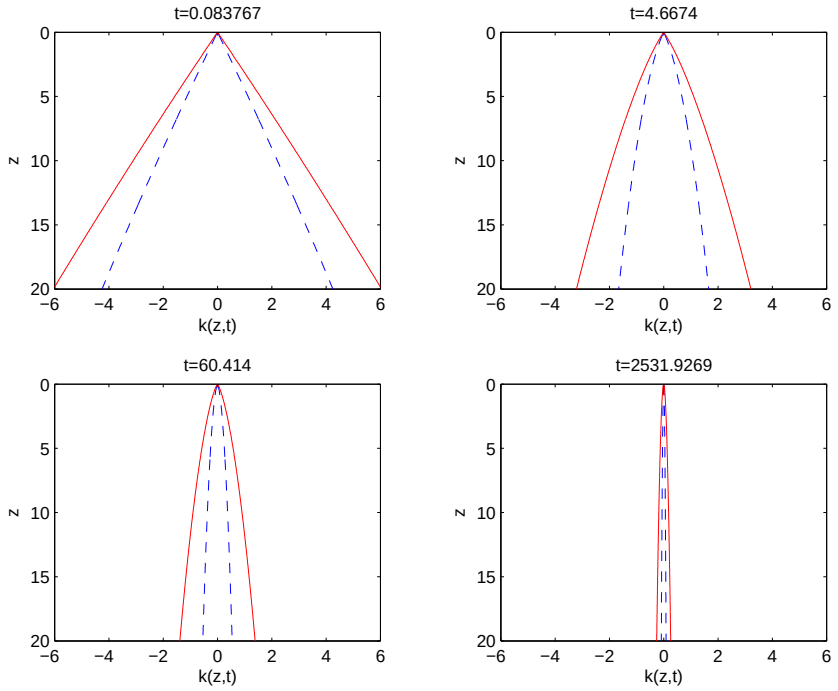

Figure 2.3: Free boundary evolution through time for $\mu = 1, \rho = 1$.

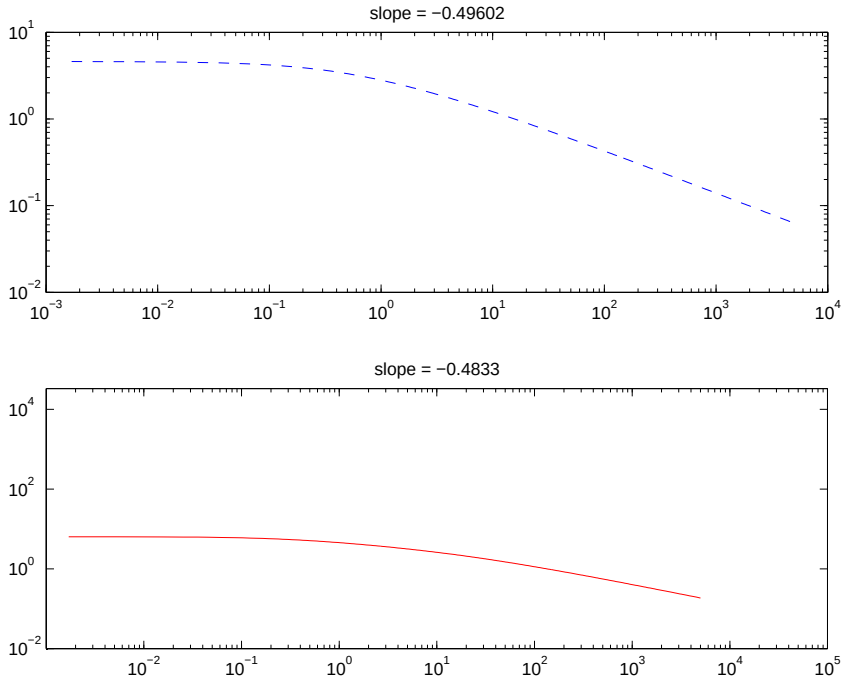

Figure 2.4: Thinning at fixed z: $k^{(i)}(z_0, t)$.

fluid combinations	μ	ρ	α	β
$\frac{\text{Polyurethane}}{\text{Silicone Oil}}$	0.0912	1.333	-0.49412	-0.40311
$\frac{\text{Water}}{\text{Polyurethane}}$	0.83333	0.0125	-0.49429	-0.49401
$\frac{\text{Same Fluid}}{\text{Same Fluid}}$	1	1	-0.49412	-0.47812
$\frac{\text{Mercury}}{\text{Water}}$	1361.4	13.546	-0.49402	-0.4942
$\frac{\text{Mercury}}{\text{Paraffin}}$	0.8168	16.9325	-0.49412	-0.38408

Table 2.1: Thinning rates of films. Boundary $k^{(1)}$ decays as t^{α}, and $k^{(2)}$ as t^{β}, for fixed z. Initial conditions: $k_0^{(1)}(z) = 0.23z$, and $k_0^{(2)}(z) = 0.32z$.

We investigated decay rates for different fluid combinations and summarized the results in Table 2.1. The results for all five cases were computed after 5000 time units. We find that the thinning rate is approximately $-\frac{1}{2}$ for every case except for the first and fifth ones. However, by letting the simulation run longer, we expect that the decay rate should approach $-\frac{1}{2}$. By increasing the time units to 20000 for the first case, we found that the decay rate for $k^{(2)}$ was -0.4622. From this result, we are confident that, in all cases, the thinning rate should eventually approach $-\frac{1}{2}$.

2.3.4 Similarity Solutions for Case I

The numerical results showed a decay rate of approximatively $-\frac{1}{2}$ for large time. Based on this observation, we now look for similarity solutions for (2.18) and (2.19) of the form

$$k^{(1)}(z,t) = f^{(1)}(z)t^{-\frac{1}{2}}, \tag{2.25}$$

$$k^{(2)}(z,t) = f^{(2)}(z)t^{-\frac{1}{2}}. \tag{2.26}$$

In addition, from the exact solution for $k^{(1)}$ obtained by the method of characteristics, we have

$$k^{(1)} \sim \sqrt{z}t^{-\frac{1}{2}} \quad \text{for } t \gg 1. \tag{2.27}$$

Thus we make an educated guess for the functions $f^{(i)}$:

$$f^{(1)}(z) = \sqrt{z}, \quad f^{(2)}(z) = C\sqrt{z}. \tag{2.28}$$

Substituting these into the equations and solving for the constant C, we obtain the value

$$C = 1 + 2\sqrt{\frac{\rho}{\mu}}. \tag{2.29}$$

This agrees with our numerical results, except in the first and fifth fluid combinations, where the fact that C is large could explain the slow convergence of $k^{(2)}$.

2.3.5 Case II: Mobile Boundary at $x = k^{(1)}$

In this second case, we replace the tangential immobility condition at the inner boundary by a condition that allows $k^{(1)}$ to move along itself. A comparison with case I is made below

$$\begin{array}{ll} \text{Case I} & \text{Case II} \\ \vec{\tau}\cdot\vec{u} = 0 & [\![\vec{\tau}\cdot\mathbf{T}\cdot\vec{n}]\!] = 0 \\ w^{(1)} = 0 & \mu w_x^{(1)} = w_x^{(2)} \end{array} \tag{2.30}$$

In case I, after changing the variables to dimensionless equivalents, and keeping only the leading order terms for $\delta^3 \ll 1$, the tangential immobility condition reduces to $w^{(1)} = 0$ at $x = k^{(1)}$. However, in case II, the condition we obtain at leading order from the tangential stress condition is a weighted matching of the x derivative of the vertical velocities ($\mu w_x^{(1)} = w_x^{(2)}$). The other equations and boundary conditions in the simplified problem are the same as in case I. Again, we can solve for the pressures and velocity components to obtain differential equations for the free boundaries only, using the kinematic condition

$$k_t^{(i)} + \left[F^{(i)}\right]_z = 0 \quad \text{for i = 1,2} \tag{2.31}$$

to get

$$F^{(1)} = \frac{k^{(1)}}{6\rho}[(2\rho + 3\mu - 6\mu\rho)(k^{(1)})^2 + 3\mu(k^{(2)})^2 + 6\mu(\rho-1)k^{(1)}k^{(2)}], \tag{2.32}$$

$$F^{(2)} = \frac{1}{6\rho}[(2\rho + \mu - 3\mu\rho)(k^{(1)})^3 + 3\mu(\rho-1)k^{(1)}(k^{(2)})^2 + 2\mu(k^{(2)})^3]. \tag{2.33}$$

Note that all the terms are still cubic in the expressions for the flux functions and that the equation for $k^{(1)}$ is now dependent on $k^{(2)}$.

2.3.6 Numerical Results for Case II

Using the method of lines introduced in case I, we can obtain numerical simulations of the free boundaries moving in time. Like in case I, we looked at the free boundary evolution of case II through time, as seen in Figure 2.5. A representative situation was that of mercury as the inner fluid and water as the outer one. For this choice of fluids, we see in Figure 2.6 that the decay rate of the inner fluid is faster than the outer one and that the decay rate of the outer fluid approaches $-\frac{1}{2}$.

2.3.7 Similarity Results for Case II

We once again analyzed the decay rate of different combinations of fluids and listed the results in Table 2.2. In all five cases, we observe that the inner fluid thins at a faster rate than the outer fluid. As we saw in case I, there is more variation in the values at $x = k^{(2)}$; however, it is worth noting that there is even more deviation in the values for this case than in case I.

From our experience with case I, we can try to find similarity solutions of the form

$$k^{(1)}(z,t) = f^{(1)}(z)t^{\alpha}, \tag{2.34}$$

$$k^{(2)}(z,t) = f^{(2)}(z)t^{\beta}, \tag{2.35}$$

where the values of α and β give the decay rates of the free boundaries for large time. It is not obvious from the table of decay rates, obtained from the numerical experiments, which values α and β should take. Substituting the similarity form for $k^{(i)}$ into the differential equations, and matching the most important terms, we get two different situations.

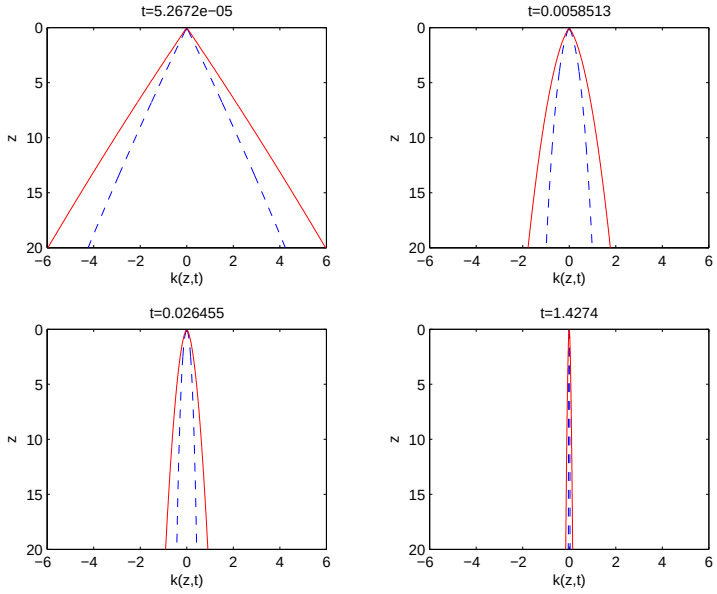

Figure 2.5: Free boundary evolution of case II through time for $\mu = 13.546, \rho = 1361.4$.

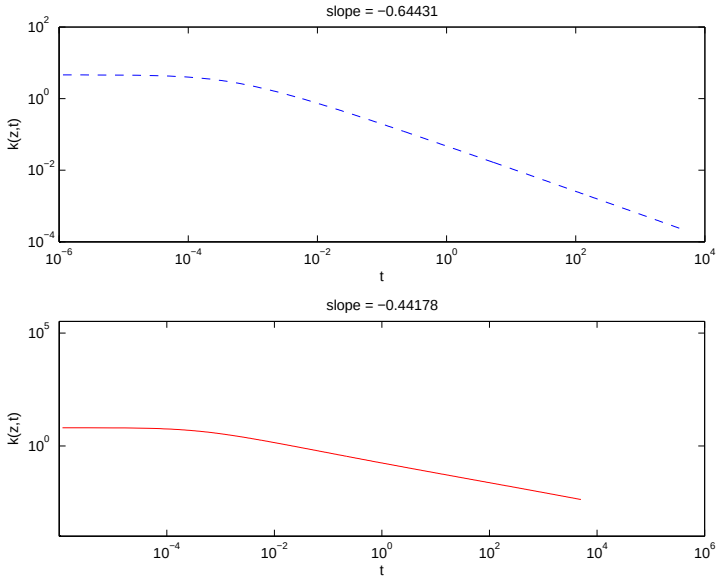

Figure 2.6: Thinning at fixed z of case II for $\mu = 13.546, \rho = 1361.4$: $k^{(i)}(z_0, t)$.

fluid combinations	μ	ρ	α	β
$\frac{\text{Polyurethane}}{\text{Silicone Oil}}$	0.0912	1.333	-0.65224	-0.4516
$\frac{\text{Water}}{\text{Polyurethane}}$	0.83333	0.0125	-0.66183	-0.5439
$\frac{\text{Same Fluid}}{\text{Same Fluid}}$	1	1	-0.65552	-0.49573
$\frac{\text{Mercury}}{\text{Water}}$	1361.4	13.546	-0.64431	-0.44178
$\frac{\text{Mercury}}{\text{Paraffin}}$	0.8168	16.9325	-0.62798	-0.41261

Table 2.2: Thinning rates of films. Boundary $k^{(1)}$ decays as t^{α}, and $k^{(2)}$ as t^{β}, for fixed z. Initial conditions: $k_0^{(1)}(z) = 0.23z$, and $k_0^{(2)}(z) = 0.32z$.

- Assuming the inner boundary decays faster than the outer boundary, $\alpha < \beta$, then by matching the exponents of t of the leading order terms, we obtain the conditions $\beta = -\frac{1}{2}$ and $\alpha \leq -\frac{1}{2}$. Hence, from this analysis, we are unable to find the specific value for α and can only conclude that it has to be smaller than $-\frac{1}{2}$.
- If the inner boundary decays slower than the outer boundary, $\alpha \geq \beta$, then by matching the exponents of t, we get the values $\alpha = \beta = -\frac{1}{2}$. This result indicates that the outer boundary cannot decay faster than the inner boundary, which makes sense since otherwise the two curves would eventually intersect.

2.4 Conclusions

For case I we obtained an exact analytical solution for the boundary $k^{(1)}$ and an approximate solution for the boundary $k^{(2)}$. Using our numerical results, we verified the thinning rate of fluids with the different physical properties, as shown in Figure 2.1.

We ran the numerical solver for case II and the results agreed with our predicted thinning rate for $k^{(2)}$. However, due to time constraints, we did not find any explicit decay rate for the boundary $k^{(1)}$. Although the $k^{(1)}$ boundary decay rate was hard to solve analytically, we were able to postulate a predicted value relative to $k^{(2)}$ and confirmed it with our numerical results.

2.5 Future Work

Our draining thin film model simplifies the physical dynamics, as we did not take into account the effects of surface tension. It would be interesting to check whether this simplification placed any restrictions on our model. It would also be interesting to compare our model with real world experimental results.

Bibliography

[1] D. J. Acheson. *Elementary fluid dynamics*. Oxford Applied Mathematics and Computing Science Series. The Clarendon Press Oxford University Press, New York, 1990.

[2] S. A. Braun, R. J. Snow and U. C. Pernisz. Gravitational drainage of a tangentially-immobile thick film. *Journal of Colloid and Interface Science*, 219(2):225–240, 1999.

[3] A. Davis S. H. Oron and S. G. Bankoff. Long-scale evolution of thin liquid films. *Rev. Modern Phys.*, 69:931–980, 1997.

Chapter 3

Stability of Hybrid Systems using a Sum of Squares Programming Approach: VCCR System Example

Participants: Sonja Glavaski (Mentor), Mark Braverman, Shengyuan Chen, Nadine Gärtner, Yujun Wu, Charles Bergeron.

PROBLEM STATEMENT: The objective of this project is to use SOSTOOLS to find a Lyapunov function and hence prove the stability of a system. The first part of this paper demonstrates the functioning of SOSTOOLS by means of an example. In the second part we propose a model for carbon dioxide removal (VCCR) in the cabin of a space station. We then propose a controller for the VCCR system and prove its stability.

3.1 Introduction to Hybrid Systems

Hybrid systems can be used to describe various human controlled dynamic processes. A hybrid system is characterized by having a finite number of *modes*. For each mode there are rules governing the continuous evolution of the *state* of the system. Furthermore there is some control structure determining when to switch between the individual modes. We will illustrate the concept with some examples before giving a formal definition of a hybrid system.

3.1.1 Examples

- One simple example of a hybrid system is an air conditioning system. The two modes of the air conditioning system correspond to the fan being on or off. The state of the system is the room temperature. A switching condition is reaching a certain temperature.
- In the automatic transmission in a car, the modes correspond to the gears. The state of the system and the switching conditions are expressed in terms of the number of revolutions in the motor.
- VCCR, a CO_2 removal system in a space station, is a more complex hybrid system which we will discuss in section 3.3.

3.1.2 Definition

A hybrid system is a dynamic system over $\mathbb{R}^n$, with a discrete set of modes $\{1, 2, \ldots, K\}$. At time t, let the system be in state $x = x(t) \in \mathbb{R}^n$, and in mode $i = \Phi(t) \in \{1, 2, \ldots, K\}$. In each mode, the system evolves according to the equation

$$\dot{x} = f_i(x),$$

where f_i is a given function for each $i \in \{1, 2, \ldots, K\}$.

For each pair of modes $(i, j) \in \{1, 2, \ldots, K\}^2$, there is a switching surface S_{ij}. The switching surface S_{ij} determines the states where the system flips from mode i to mode j. Formally the effect of a switching surface can be expressed as

$$\Phi(t) = j \text{ if } \Phi(t^-) = i \text{ and } x(t) \in S_{ij}.$$

3.2 Lyapunov Stability

3.2.1 The Lyapunov Theorem for Plain Dynamic Systems

Suppose we are given a dynamic system $\dot{x} = f(x)$. A point x_0 is an equilibrium point of the system if $f(x_0) = 0$. An equilibrium point x_0 is called *asymptotically stable* if the system will return to x_0 when slightly removed from there. An equilibrium point x_0 is called *stable* if the system will return to a bounded area containing x_0 after being slightly removed from x_0. Given an equilibrium point, we are interested in determining whether it is a stable equilibrium or not. One of the main tools for proving stability of a system is the Lyapunov theorem which is stated below.

Theorem:
Let $\dot{x} = f(x),\ x \in R^n$ be a system. Suppose $f(0) = 0$, i.e. 0 is an equilibrium point. If for some open region D with $0 \in D \subset R^n$ there is a continuously differentiable function $V : D \to \mathbb{R}^+$, such that

(a)$V(0) = 0$,
(b) $V(x) > 0$ for $x \in D \setminus \{0\}$,

then

if $\dot{V}(x) \leq 0$ for $x \in D$, the system is stable at the origin.

If $\dot{V}(x) < 0$ for $x \in D$, the system is asymptotically stable at the origin.

In this case, V is called a *Lyapunov function*.

In short, the theorem says that if we can find a Lyapunov function for a system, we know the system is stable. Intuitively, a Lyapunov function can be thought of as the energy of the system, which can decrease, but not increase.

3.2.2 A Lyapunov-like Theorem for Hybrid Systems

In order to apply the Lyapunov theorem to a hybrid system some extra precautions have to be taken. In contrast to a 'plain' dynamic system, in a hybrid system there are several possible modes of the system. Furthermore, in a hybrid system there can be states of the systems that do not correspond to a unique mode. Because of this, a 'Lyapunov like' function for a hybrid system not only depends on the state x of the system, but also on its mode i. We therefore need to introduce functions $V_i(x)$ for the individual modes of the system. Expressing the mode i of the system as a function Φ of the time we can formally combine the functions V_i into an overall function V:

$$V(t) = V_{\Phi(t)}(x(t))$$

Of course we need to ensure the function V satisfies certain conditions before we can claim that such a V can take the role of a Lyapunov function.

Given a hybrid system with modes $i = 1, 2, \ldots, K$, for each i we define a region X_i, such that

$$x(t) \in X_i, \quad \text{whenever} \quad i = \Phi(t).$$

We are now ready to formulate a Lyapunov like Theorem for hybrid systems. This is not the most general version of such a theorem, but it is strong enough for our purposes. See [3] and [5] for more details.

Theorem:
Suppose we are given a hybrid system with an equilibrium point at the origin (i.e. $f_i(0) = 0$ for each $i = 1, 2, \ldots, K$).
If for some open region D with $0 \in D \subset \mathbb{R}^n$, there are continuously differentiable functions $V_i : D \to \mathbb{R}^+,\ i = 1, 2, \ldots, K$, such that

(a)$V_i(0) = 0$ for each i,
(b) $V_i(x) > 0$ for $x \in X_i \cap D \setminus \{0\}$ for each i,
(c) $\dot{V}_i < 0$ on $X_i \cap D$ for each i,
(d) For each $i, j \in \{1, 2, \ldots, K\}$, $V_i(x) \geq V_j(x)$ for $x \in S_{ij}$,

then the system is asymptotically stable at the origin.

Note that the first conditions ensure that the functions V_i behave like Lyapunov functions on their associated regions X_i. The last condition ensures that the combined function V does not increase during mode switchings.

3.2.3 An Example

Consider a hybrid system $\dot{x} = f(x)$ with two modes. Let the mode dependent evolution subsystems $f_1(x)$ and $f_2(x)$ be given as follows:

$$f_1(x) = \begin{bmatrix} -x_1 & -100x_2 \\ 10x_1 & -x_2 \end{bmatrix}$$

$$f_2(x) = \begin{bmatrix} x_1 & +10x_2 \\ -100x_1 & +x_2 \end{bmatrix}.$$

Let the initial mode of the system be $\Phi(0) = 1$. Let the switchings between modes 1 and 2 be determined by the following switching rules:

$$\Phi(t) = \begin{cases} 1 & \text{if } i(t^-) = 2\,,\, 2x_1(t) - x_2(t) \geq 0 \\ 2 & \text{if } i(t^-) = 1\,,\, -px_1(t) - x_2(t) \geq 0. \end{cases}$$

We want to investigate for which parameter values $p > 0$ the system is stable. Figure 3.1 illustrates the evolution of the system for values of p that make the system unstable or stable, respectively.

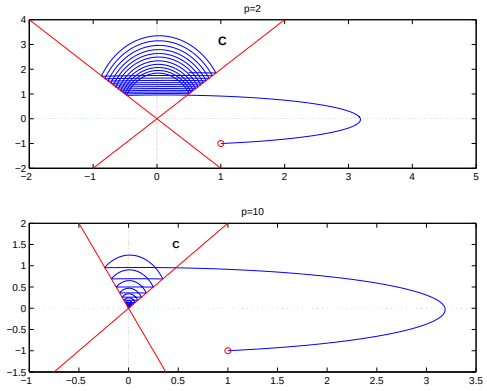

Figure 3.1: Values of p that make the system unstable (top) and stable (bottom).

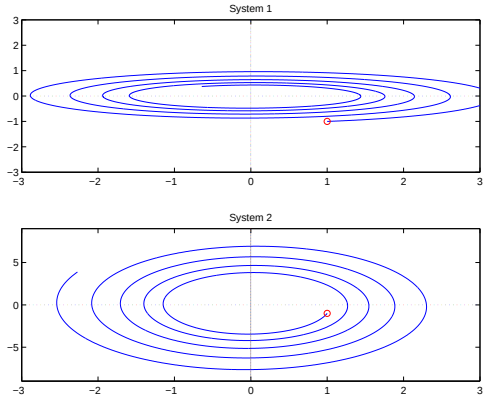

Figure 3.2: Evolution of the subsystems.

A formal analysis of the system reveals that there is a critical value of p that separates stable systems from unstable ones. We will not go into details of how to find this critical value analytically since this project is concerned with investigating the performance of SOSTOOLS on the stability problem.

For the moment we will ignore the switching rules in order to observe the behaviour of the subsystems under the assumption it remains in the same mode forever. We notice that in mode 1 the subsystem is stable and evolves counterclockwise whereas in mode 2 it evolves clockwise and is unstable (see figure 3.2).

Taking the switching conditions back into consideration we make some further observations: After starting in mode 1 and hence moving counterclockwise, the system stays in mode 1 until it hits the positive ray of the line $x_2 = -px_1$. Then it switches to mode 2, and starts moving clockwise. It stays in mode 2 until it hits the positive

ray of the line $x_2 = 2x_1$. This process continues, which means that after a certain time the state of the system always lies within the cone C determined by these two rays. (see figure 3.1).

We see that the regular Lyapunov theorem is not applicable in proving stability here, because the existence of a single Lyapunov function for both modes simultaneously would in particular imply stability of system 2, which we have identified to be unstable. Hence we invoke the modified Lyapunov theorem from above.
In our particular example, the conditions of the Lyapunov like theorem are:

(a) $V_1(0) = V_2(0) = 0$

(b) $V_1(x)$ is decreasing wherever mode 1 occurs. Denote by X_1 the set on which mode 1 occurs. So: $\dot{V_1}(x) < 0$ for $x \in X_1$. In our case $X_1 = C$.

(c) $\dot{V_2} < 0$ for $x \in X_2$. Here $X_2 = C$.

(d) For $x \in S_{12}$, $V_1(x) > V_2(x)$.

(e) For $x \in S_{21}$, $V_2(x) > V_1(x)$.

3.2.4 Using SOSTOOLS

Among other things, the SOSTOOLS software package [7] can be used to search for Lyapunov functions. As the name suggests, SOSTOOLS deals with sums of squares (of monomials). A sum of squares is always non-negative and hence a potential Lyapunov function. So using SOSTOOLS for finding Lyapunov functions corresponds to decreasing the search space from the space of non-negative functions to the space of sums of squares. So restricting the search space greatly reduces computation times — however at the cost of possibly missing out on finding Lyapunov functions.

In order to use the SOSTOOLS software on the example stated in the section above, we need to adjust its formulation. To this end we make the somewhat restrictive assumption that V_1 and V_2 are polynomials which then allows us to rewrite the problem in terms of polynomial inequalities.

As we have seen above the state of the system will eventually lie within the cone C. We can represent C parametrically as

$$C = \{\alpha^2(1,2) + \beta^2(-1,p) : \alpha, \beta \in \mathbb{R}\}.$$

Using this parametrical representation and rewriting the conditions (a)–(e) from above yields:

(a) $V_1(x) > 0$,

(b) $V_2(x) > 0$,

(c) $-\dot{V_1}(\alpha^2(1,2) + \beta^2(-1,p)) > 0$,

(d) $-\dot{V_2}(\alpha^2(1,2) + \beta^2(-1,p)) > 0$,

(e) $V_2(\alpha^2(1,2)) - V_1(\alpha^2(1,2)) > 0$,

(f) $V_1(\alpha^2(-1,p)) - V_2(\alpha^2(-1,p)) > 0$.

We can now use SOSTOOLS to test for which values of p it can find Lyapunov like functions V_1, V_2 for our example. SOSTOOLS allows us to specify the maximal degree of the polynomials used in the search. Of course, for any particular p the higher degree of polynomials we chose, the more candidates for Lyuapunov functions there are. We experiment with polynomials of degrees ranging from 2 to 6. From the analytical analysis we know that for any p below 2.16 the system is unstable and for any p above 2.16 the system is stable. Using SOSTOOLS with polynomials of degree 2 the lowest p for which we can find a Lyapunov function is 5.9. Using polynomials of degree 4 we find Lyapunov functions for p down to 2.5. With polynomials of degree 6 we get even closer to the critical value, namely down to 2.22.

3.3 Modelling the CO_2 Removal System in a Space Station

3.3.1 Introduction and a Mathematical Model

In the context of space travel arises the problem of oxygen retrieval. Among other things the process involves removal of carbon dioxide which can be accomplished via a *variable configuration carbon dioxide removal system* or VCCR. In the following we will concentrate just on the VCCR system itself and treat the rest of the oxygen retrieval system as an independent black box. The VCCR system roughly works as follows: carbon dioxide is removed from the crew cabin and absorbed into a so called *bed.* In the bed there are two modes of aggregation: the carbon dioxide enters the bed in gaseous form and then compounds with some gel present in the bed.

Whenever the connection between the crew cabin and the bed is cut off the bed can be partially depleted from the carbon dioxide.

So, in total there are two modes of the VCCR system. The mode when carbon dioxide leaves the crew cabin and enters the bed will be referred to as *adsorbing mode* or *mode A*. The mode when the bed is being depleted from carbon dioxide is called *desorbing mode* or *mode D.*

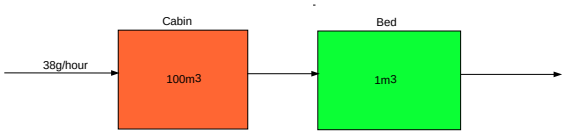

Figure 3.3: A sketch of the carbon dioxide removal system.

In mode A some carbon dioxide is produced in the cabin due to the breathing of the crew members. Also, some air (and hence carbon dioxide) leaves the cabin. At the same time, the carbon dioxide that leaves the cabin enters the bed. Introducing variables C_c and C_b representing the carbon dioxide concentration in the cabin and the bed respectively we can state these interrelationships in the form of two differential equations:

$$\text{System A} = \begin{cases} \dot{C}_c = k(C_b - C_c) + \frac{r}{v_c} \\ \dot{C}_b = k(C_c - C_b)\ \frac{v_c}{v_b}\ \frac{1}{s+1}. \end{cases}$$

With v_c standing for the volume of the cabin and r standing for the rate by which carbon dioxide is produced by the crew, the term $\frac{r}{v_c}$ represents the rate by which the carbon dioxide concentration increases in the cabin. We assume that the amount of carbon dioxide that can leave the cabin is proportional to the difference in concentrations in the cabin and the bed with some coefficient k. Those two factors lead to the first equation. We assume that the amount of CO_2 in the solid state is always in constant proportion to the amount of CO_2 in gaseous state in the bed with proportion factor s. So we have

$$\text{The amount of } CO_2 \text{ in solid state} = s \cdot C_b v_b.$$

In the adsorbing mode, the system remains closed. The only factor which affects the total amount of CO_2 in the system is the astronauts breathing. Hence the total amount of CO_2 in the system increases at a constant rate r. The total amount of CO_2 in the system at time t is

$$T(t) = CO_2^{cabin} + CO_2^{bed} = C_c v_c + C_b v_b + s \cdot C_b v_b = C_c v_c + (s+1) \cdot C_b v_b,$$

and we know that $\dot{T} = r$. Differentiating T we obtain

$$\dot{C}_c v_c + (s+1) \cdot \dot{C}_b v_b = r.$$

Combining with $\dot{C}_c = k(C_b - C_c) + \frac{r}{v_c}$ yields

$$\dot{C}_b = \frac{k(C_c - C_b)v_c}{(s+1)v_b} = k(C_c - C_b)\frac{v_c}{v_b}\frac{1}{s+1}.$$

Which proves the second equation in system A.

In mode D all that happens in the cabin is the crew breathing and hence increasing the carbon dioxide concentration linearly over time. The bed is being depleted from some of the carbon dioxide it holds and we assume this can be done in a rate proportional to the carbon dioxide concentration in the bed. We can then state the evolution of the system in mode D as follows:

$$\text{System D} = \begin{cases} \dot{C}_c = \frac{r}{v_c} \\ \dot{C}_b = -\beta C_b. \end{cases}$$

In our discussion of the stability of the system we will assume the parameters take the following (more or less arbitrarily chosen) values:

$$v_c = 100m^3 \quad k = 3 \quad r = 38\tfrac{g}{h} \quad v_b = 1m^3 \quad \beta = 1 \quad s = 99.$$

The need for a carbon dioxide removal system originates from a constraint on the acceptable carbon dioxide concentration in the air in the cabin:

$$5 \leq C_c \leq 8\ \frac{g}{m^3}.$$

So having specified the system parameters, variables, constraints, modes and the evolution rules, we can now move on to controlling the system, i.e. determining the switching conditions.

3.3.2 Suggesting a Controller

Common sense suggests that a relatively high carbon dioxide concentration in the cabin and a low concentration in the bed should take the system into the adsorbing mode. On the other hand, a low concentration in the cabin and a high concentration in the bed suggests emptying the bed. Hence we expect that a diagonal in the $C_c - C_b$ plane makes a sensible switching line which causes the state of the system to stay in the feasible region with respect to carbon dioxide concentrations. In order to increase the robustness of the controller we decide to introduce a buffer zone. To this end we introduce two parallel switching lines instead of only one. Within the buffer zone we allow the system to be in either mode. This concession reduces the dithering in the system. In figure 3.4 we give a sketch showing how the two switching lines divide the feasible region of the $C_c - C_b$ plane into three areas I, II and III. Area I represents the values of C_c and C_b for which we want the system to be in mode D, in area II we want the system to be in mode A. Area III is the buffer zone.

To be specific, we choose the switching lines l_1 and l_2 to be the following:

$$\begin{array}{ll} l_1: & C_b = p(C_c - 5.5) \quad \text{(switching from mode A to mode D)} \\ l_2: & C_b = p(C_c - 7) \quad \text{(switching from mode D to mode A)}. \end{array}$$

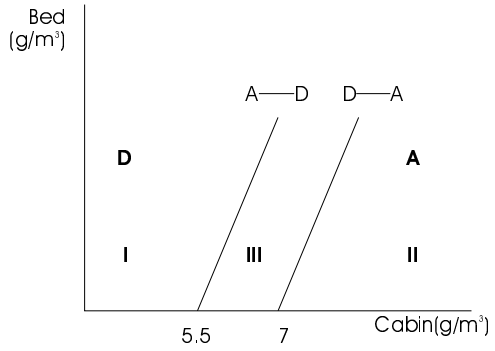

Figure 3.4: Dividing the plane into three regions.

The slope of the line, i.e. the parameter p, can be adjusted to keep the system stable in the feasible region. After some inspection, we choose p to be 0.66. Note that we choose the intersection points of the switching lines with the C_c axis to be 5.5 and 7, such that eventually the system will stay *strictly* within the feasible region $5 < C_c < 8$. The reason why we choose 5.5 and 7 rather than 5 and 8 will become clear shortly.

3.3.3 A Lyapunov Style Theorem

Theorem

Given a dynamical system $\dot{x} = f(x)$ (possibly hybrid) and a continuously differentiable function $V(x)$ such that

1. $V(x) \geq 0$ for all x.
2. For some closed bounded region K, $\dot{V}(x) < 0$ for $x \notin K$.

Denote $m = max_{x \in K} V(x)$ and $S = V^{-1}([0, m]) = \{x : V(x) \leq m\}$,

then for each $x_0 = x(0)$ there is T, such that $x(t) \in S$ for $t > T$.

The Intuition Behind the Theorem

Suppose the system starts at some arbitrary point $x_0 = x(0)$. Then as long as the system stays out of the region K, the value of V decreases. Since $V(x) \leq m$ on K, $V(t)$ is decreasing as long as $V(x(t)) > m$. Hence eventually we have $V(x(t)) \leq m$, and the system reaches the set S. Also, $K \cap S^c = \emptyset$, and hence V is nonincreasing when the system is on the boundary of S. So once the system is in S, $V(t)$ can never increase above m, and the system stays within S.

3.3.4 Stability of the VCCR System

Introducing a Lyapunov Function

In order to prove stability of the carbon dioxide system we want to make use of the Lyapunov technique. We claim that the following function V is a Lyapunov function for our system:

$$V(C_c, C_b) = \begin{cases} N(C_b - p(C_c - 7))^2 + N(C_b - p(C_c - 5.5))^2 + (C_b + \frac{C_c}{p})^2 & \text{in areas I, II} \\ N(1.5p)^2 + (C_b + \frac{C_c}{p})^2 & \text{in area III} \end{cases}$$

The branch of function V belonging to areas I and II consists of three terms: The first two terms are a multiple of a

point's distance from the switching lines. That is, the value of $V(C_c, C_b)$ increases with increasing distance from the switching lines. So the effect of the first two terms is to pull the state of the system towards acceptable values of C_c. The vigorousness of the distance penalty can be fine-tuned by choosing the parameter N accordingly. We choose N to be 200. The third part of the first branch of V resembles the distance from an imaginary line. The imaginary line is perpendicular to the switching lines and cuts through the origin. The purpose of adding the distance from the imaginary line is to pull the state of the system 'down' towards small values of C_b.

When we are in the buffer zone, i.e. area III, the carbon dioxide concentration in the cabin C_c is within the acceptable range. Hence we are mainly concerned with pulling the system towards low values of C_b. Again, this is accomplished by considering the distance $(C_b + \frac{C_c}{p})^2$ from the same imaginary line that was introduced above. In addition to that, we introduce the term $N(1.5p)^2$ to ensure the continuity of V.

Proving Stability of the System

We now apply the Lyapunov style theorem using the function V described above. First, we observe that V is continuously differentiable, and $V \geq 0$. We now check using Matlab, on which region the condition $\dot{V} < 0$ holds. First we verify that this condition holds on regions I and II. In region I, the system is always in mode D. Hence $\dot{V} = \frac{dV}{dx} \cdot \dot{x}_D$ in region I. In region II, the system is always in mode A. So here $\dot{V} = \frac{dV}{dx} \cdot \dot{x}_A$. From figures 3.5 and 3.6 below we see that $\dot{V} < 0$ both in region I and II.

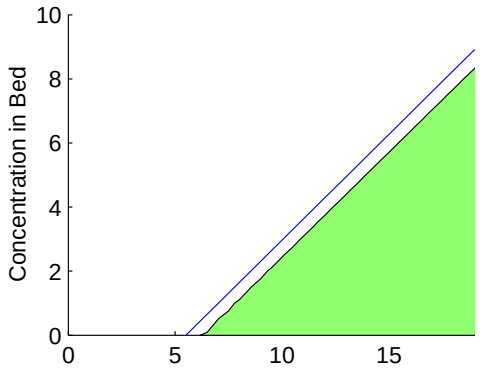

Figure 3.5: In region I, the system is in mode D. White region= $\dot{V} < 0$.

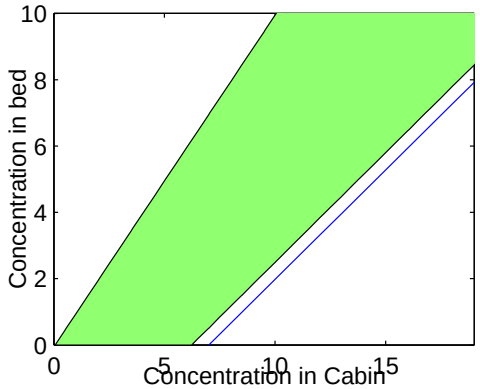

Figure 3.6: In region II, the system is in mode A. White region=$\dot{V} < 0$.

In region III the system could be in either mode A or D. To enforce the condition $\dot{V} < 0$, we need to have both $\frac{dV}{dx} \cdot \dot{x}_A < 0$ and $\frac{dV}{dx} \cdot \dot{x}_D < 0$. Figures 3.7, 3.8 and 3.9 illustrate that $\frac{dV}{dx} \cdot \dot{x}_A < 0$ holds on the entire region III, and that $\frac{dV}{dx} \cdot \dot{x}_D < 0$ holds for all points in region III, except for a subset K.

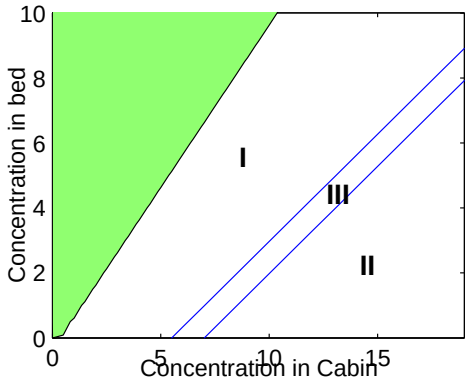

Figure 3.7: In region III, in the case the system is in mode A. White region=$\dot{V} < 0$.

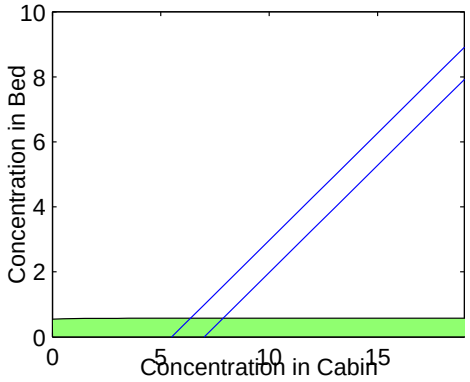

Figure 3.8: In region III, in the case the system is in mode D. White region = $\dot{V} < 0$.

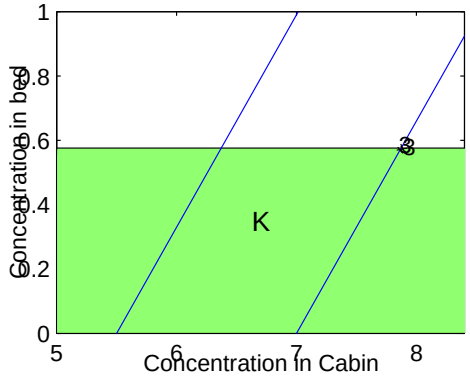

Figure 3.9: A close-up on set K.

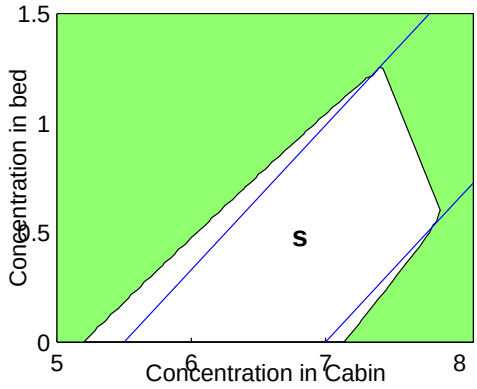

Figure 3.10: The set S=$V^{-1}([0, m])$.

The Lyapunov style theorem states that the system stably converges to the set S. Since $5 < C_c < 8$ for all the points in S, the constraints on the carbon dioxide concentration in the cabin will be satisfied.

3.4 Conclusions

As usual, there are plenty of opportunities for further investigations. Concerning the first part it would be desirable to run SOSTOOLS on the given example using polynomials of higher degrees in order to explore how closely SOSTOOLS advances the critical value of p, the switching surface parameter.

As to the second part, there are various possible extensions. Unrolling the possibilities from the end, first of all it would be interesting to express area III parametrically and use SOSTOOLS for the stability analysis. Also, we could experiment with different switching lines, in particular with non-parallel ones. In another step the controller could be optimized in order to minimize the amount of dithering in the system. Finally, a more sophisticated model of the carbon dioxide removal system could incorporate the transition of carbon dioxide between the solid and the gaseous parts of the bed.

Bibliography

[1] A. Papachristodoulou and S. Prajna, On the Construction of Lyapunov Functions using the Sum of Squares Decomposition, Proceedings of the 41st IEEE Conference on Decision and Control, Las Vegas, Nevada, USA, 2002.

[2] J. T. Malin et al., Multi-Agent Diagnosis and Control of an Air Revitalization System for Life Support in Space, IEEE Aerospace, 2000.

[3] S. Petterson and B. Lennartson, Stability and Robustness for Hybrid Systems, Proceedings of the 35th Conference on Decision and Control, Kobe, 1996, p. 1202–1207.

[4] P. A. Parrilo, Semidefinite Programming Relaxations for Semialgebraic Problems,Mathematical Programming Ser. B, Vol. 96, No. 2, pp. 293–320, 2003.

[5] M. S. Branicky, Stability of Switched and Hybrid Systems, Proceedings of the 33rd Conference on Decision and Control, Lake Buena Vista, 1994, p. 3498–3503.

[6] S. Boyd et al., Linear Matrix Inequalities in System and Control Theory, SIAM Studies in Applied mathematics, Vol. 15, 1994.

[7] S. Prajna et al., SOSTOOLS: User's Guide, 2002, http://www.cds.caltech.edu/sostools/.

Chapter 4

Modelling Polymer-Purification by Counter-Current Exchange

Participants: David Misemer (Mentor), Zhenlu Cui, Hua Li, Seung Lee, Qian Wang, Fabien Youbissi.

PROBLEM STATEMENT: Adhesives are commonly used for medical and electrical purposes, such as trans-thermal skin patches and black electrical tapes electricians carry with them. It is necessary that adhesive should have the least impurities because impurities would cause side-effect of patches or malfunction of electrical devices. That is, the material for adhesives, polymer melt, should be highly purified. The engineers at 3M use the counter-current exchange process to remove the residual monomers to obtain highly purified polymer melt.

While the co-current exchange process is limited due to equilibrium partition, the counter-current exchange process does not go to the state that the exchange stops. The exchange of O_2 and CO_2 at the gills of fish is an example of the counter-current exchange process.

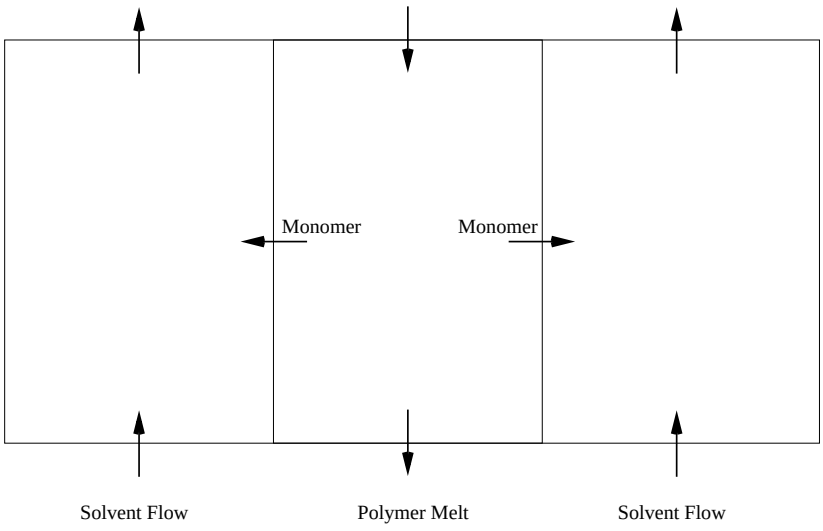

Figure 4.1: Schematic Picture of 2 Zone Model.

In this paper, we study the purification process using the counter-current exchange, build two models that fit experimental data, and explain the physics of the counter-current exchange process. In the end, we compare two models.

4.1 2 Zone Model

Here we assume that there are only two flows in the column where the exchange of residual monomer in the polymer solvent takes place. One is the flow of the polymer melt which goes down under gravity force and the other is the flow of the solvent which is pushed up. We also assume that two flows are concentric and that residual monomers in the flows are uniformly distributed at each level. Figure 4.1 is the schematic picture of the 2 zone model.

4.1.1 The Equation Set-up

In order to derive relationships between the concentrations of residual monomers in the polymer melt and the solvent, we define functions and symbols as follows: Let L be the length of the cylindrical column where the polymer melt and the solvent exchange monomers. Let r be the radius of the column of the polymer melt. Let R be the radius of the cylindrical column . Let $C_p(t,h)$ and $C_s(t,h)$ be the concentrations of residual monomers in the polymer melt and the solvent at time t and at height h, respectively. Here we set the bottom of the column $h = 0$. Then, at the interface of the polymer melt and the solvent, we have the monomer mass flux $j\text{mass} = k(C_p - p\,C_s)$, where p is a partition constant and k is an exchange efficiency constant. Let v_p and v_s be the velocity of the polymer melt and the solvent, respectively. Relating to the velocity, we also have flow rates of the polymer melt and the solvent, denoted by f_p and f_s, respectively. We assume that they are constant; i.e., the polymer melt and the solvent are put into the main column with fixed rates. Let A_p and A_s be the cross-sectional area of the columns of the polymer melt and the solvent, respectively. Then we have

$$A_p = \pi r^2, \qquad A_s = \pi(R^2 - r^2).$$

From time t to time $t + \triangle t$, the amount

$$v_p \cdot \triangle t \cdot C_p(t,h) \cdot A_p = v_p \cdot \triangle t \cdot C_p(t,h) \cdot \pi r^2$$

of the mass of the polymer melt comes in at height h and the amount

$$v_p \cdot \triangle t \cdot C_p(t, h - \triangle h) \cdot A_p = v_p \cdot \triangle t \cdot C_p(t, h - \triangle h) \cdot \pi r^2$$

of the mass of the polymer melt comes out at height $h - \triangle h$. And in time $\triangle t$, the polymer melt loses to the solvent the amount of residual monomers that is

$$j_{mass} \cdot \triangle t \cdot \text{(Area of Exchange)} = k(C_p(t, h) - p\, C_s(t, h)) \cdot \triangle t \cdot 2\pi r \cdot \triangle h.$$

Due to the conservation law of mass, we have the following:

$$\begin{aligned}
\triangle M_p &= M_p(t + \triangle t) - M_p(t) \\
&= v_p \cdot \triangle t \cdot C_p(t, h) \cdot A_p - v_p \cdot \triangle t \cdot C_p(t, h - \triangle h) \cdot A_p - j_{mass} \cdot \triangle t \cdot \text{(Area of interface)} \\
&= v_p \cdot \triangle t \cdot C_p(t, h) \cdot \pi r^2 - v_p \cdot \triangle t \cdot C_p(t, h - \triangle h) \cdot \pi r^2 - k(C_p - p\, C_s) \cdot \triangle t \cdot 2\pi r \cdot \triangle h.
\end{aligned}$$

Dividing the above by $A_p \cdot \triangle t \cdot \triangle h$, we have

$$\frac{C_p(t + \triangle t, h) - C_p(t, h)}{\triangle t} = v_p \frac{C_p(t, h) - C_p(t, h - \triangle h)}{\triangle h} - \frac{2k}{r}(C_p(t, h) - p\, C_s(t, h)).$$

Letting $\triangle t$ and $\triangle h$ tend to zero, we have

$$\frac{\partial C_p}{\partial t}(t, h) = v_p \frac{\partial C_p}{\partial h}(t, h) - \frac{2k}{r}(C_p(t, h) - p\, C_s(t, h)),$$

or

$$\frac{1}{v_p}\frac{\partial C_p}{\partial t}(t, h) = \frac{\partial C_p}{\partial h}(t, h) - \frac{2\pi r k}{f_p}(C_p(t, h) - pC_s(t, h)).$$

Now we turn our attention to C_s. Like the case of C_p, we use the conservation law of mass. From time t to time $t + \triangle t$, the amount

$$v_s \cdot \triangle t \cdot C_s(t, h - \triangle h) \cdot A_s = v_s \cdot \triangle t \cdot C_s(t, h - \triangle h) \cdot \pi(R^2 - r^2)$$

of the mass of the solvent comes in at height h and the amount

$$v_s \cdot \triangle t \cdot C_s(t, h) \cdot A_s = v_s \cdot \triangle t \cdot C_s(t, h) \cdot \pi(R^2 - r^2)$$

of the mass of the solvent comes out at height $h - \triangle h$. And in $\triangle t$, the solvent gains the amount of residual monomers that is

$$j_{mass} \cdot \triangle t \cdot \text{(Area of Exchange)} \approx k(C_p(t, h) - p\, C_s(t, h)) \cdot \triangle t \cdot 2\pi r \cdot \triangle h.$$

Thus

$$\begin{aligned}
\triangle M_s &= M_s(t + \triangle t) - M_p(t) \\
&= v_s \cdot \triangle t \cdot C_s(t, h - \triangle h) \cdot A_s - v_s \cdot \triangle t \cdot C_s(t, h) \cdot A_s + j_{mass} \cdot \triangle t \cdot \text{(Area of Exchange)} \\
&= v_s \cdot \triangle t \cdot C_s(t, h - \triangle h) \cdot \pi(R^2 - r^2) - v_s \cdot \triangle t \cdot C_s(t, h) \cdot \pi(R^2 - r^2) \\
&\quad + k(C_p(t, h) - p\, C_s(t, h)) \cdot \triangle t \cdot 2\pi r \cdot \triangle h.
\end{aligned}$$

Dividing the above by $A_s \cdot \triangle t \cdot \triangle h$, we have

$$\frac{C_s(t+\triangle t,h)-C_s(t,h)}{\triangle t} = v_s \frac{C_s(t,h-\triangle h)-C_s(t,h)}{\triangle h} + \frac{2kr}{R^2-r^2}(C_s(t,h)-p\,C_s(t,h)).$$

Letting $\triangle t$ and $\triangle h$ tend to zero, we have

$$\frac{\partial C_s}{\partial t}(t,h) = -v_s \frac{\partial C_s}{\partial h}(t,h) + \frac{2rk}{R^2-r^2}(C_p(t,h)-p\,C_s(t,h)),$$

or

$$-\frac{1}{v_s}\frac{\partial C_s}{\partial t}(t,h) = \frac{\partial C_s}{\partial h}(t,h) - \frac{2\pi rk}{f_s}(C_p(t,h)-pC_s(t,h)).$$

Therefore, we have a system of partial differential equations

$$\begin{cases} \frac{1}{v_p}\frac{\partial C_p}{\partial t}(t,h) &= \frac{\partial C_p}{\partial h}(t,h) - \frac{2\pi rk}{f_p}(C_p(t,h)-pC_s(t,h)) \\ -\frac{1}{v_s}\frac{\partial C_s}{\partial t}(t,h) &= \frac{\partial C_s}{\partial h}(t,h) - \frac{2\pi rk}{f_s}(C_p(t,h)-pC_s(t,h)) \end{cases}$$

with boundary conditions $C_p(t,L)=\alpha$ and $C_s(t,0)=0$ for all $t>0$, where α is given.

4.1.2 Steady-State Phenomena

The steady-state of the system means that the concentrations do not depend on time t; i.e., the system has gone through a certain amount of time to have the time-independent concentrations of residual monomers in the polymer melt and the solvent.

Quantitative Analysis

Here we denote the steady-state concentrations of residual monomers in the polymer melt and the solvent by $C_{p,\infty}$ and $C_{s,\infty}$, respectively. So we have the following system of first-order ordinary differential equations:

$$\begin{cases} 0 &= \frac{\partial C_{p,\infty}}{\partial h}(t,h) - \frac{2\pi rk}{f_p}(C_{p,\infty}(t,h)-pC_{s,\infty}(t,h)) \\ 0 &= \frac{\partial C_{s,\infty}}{\partial h}(t,h) - \frac{2\pi rk}{f_s}(C_{p,\infty}(t,h)-pC_{s,\infty}(t,h)) \end{cases} \tag{4.1}$$

or

$$\begin{pmatrix} \frac{dC_{p,\infty}}{dh}(h) \\ \frac{dC_{s,\infty}}{dh}(h) \end{pmatrix} = 2\pi rk \begin{pmatrix} \frac{1}{f_p} & -\frac{p}{f_p} \\ \frac{1}{f_s} & -\frac{p}{f_s} \end{pmatrix} \begin{pmatrix} C_{p,\infty}(h) \\ C_{s,\infty}(h) \end{pmatrix},$$

with boundary conditions $C_{p,\infty}(L)=\alpha$ and $C_{s,\infty}(0)=0$. Let $\mathbf{A}$ be the matrix

$$\begin{pmatrix} \frac{1}{f_p} & -\frac{p}{f_p} \\ \frac{1}{f_s} & -\frac{p}{f_s} \end{pmatrix}.$$

If we let $\mu = \frac{2k}{rv_p}$, $\eta = \frac{2kr}{(R^2-r^2)v_s}$ and $M_f = \frac{\mu}{\eta} = \frac{f_p}{f_s}$, then the eigenvalues of $\mathbf{A}$ are 0 and $\mu-\eta p$.

Case 1. Assume that the second eigenvalue is not zero and denote it by λ. Then the solution of the system of the ordinary differential equations is

$$\begin{cases} C_{p,\infty}(h) &= \alpha \dfrac{e^{\lambda h} - M_f p}{e^{\lambda L} - M_f p} \\ \\ C_{s,\infty}(h) &= \alpha \dfrac{2\pi r k}{f_s} \dfrac{e^{\lambda h} - 1}{\frac{1}{M_f p} e^{\lambda L} - 1} \end{cases} \tag{4.2}$$

for $0 \leq h \leq L$. It is easy to see that the effectiveness of the process can be characterized by the ratio E of the initial concentration and the exit concentration of the polymer melt; i.e.,

$$E = \frac{C_{p,\infty}(0)}{C_{p,\infty}(L)} = \frac{\frac{1}{M_f p} - 1}{\frac{1}{M_f p} e^{\lambda L} - 1} < \alpha.$$

Case 2. We assume $\lambda = 0$; i.e., $\mu = \eta p$. Then it is easy to see that $\mathbf{A}^2 = \mathbf{0}$. The solution of (4.1) is

$$\begin{cases} C_{p,\infty}(h) &= \alpha \dfrac{f_p + 2\pi r k h}{f_p + 2\pi r k L} \\ \\ C_{s,\infty}(h) &= \alpha \dfrac{f_p}{1 + 2\pi r k L} \end{cases}$$

for $0 \leq h \leq L$. So we have

$$C_{p,\infty}(0) = \alpha \frac{f_p}{1 + 2\pi r k L}.$$

Qualitative Analysis

From the solutions above, we notice that in either case the exchange process removes the residual monomers from the polymer melt. From the experimental data, we can find the material constant k that is the exchange efficiency constant in the ordinary differential equations. Here is one set of the experimental data. For this set of parameters, we have a non-zero eigenvalue. The column has height 16 feet and diameter 1.5 inch. The polymer melt comes into the column through 140 holes of radius 3 mil. The mass ratio of the solvent to the polymer melt is 3.95:1 and the flow rate of the solvent is 104 gram per minute. The effective removal ratio E defined by $\frac{C_{p,\infty}(0)}{C_{p,\infty}(L)}$ is 0.0034. Plugging all data in (4.2), we have $k = 1.8607$ by computation. So we use the exponentially increasing concentrations which are given in (4.2).

In Figure 4.2, as the polymer melt flows down the column, the concentration of residual monomers in the polymer decreases. On the other hand, that of the monomers in the solvent increases as the solvent is pushed up.

E gives us an idea about how long the column should be. Taking $k = 1.8607$ and using the solution given in (4.2), we get Figure 4.3.

4.2 3 Zone Model

As we all experience when we have a shower, water spray from the shower-head carries nearby air with itself. Likewise, in the column, the polymer goes down with the solvent nearby. Thus, unlike the previous system, we have three layers of the polymer melt and the solvent. Two layers are the ones of the polymer melt and the solvent, and in the middle of the two there is a layer of the solvent that goes down with the polymer melt. See Figure 4.4.

4.2.1 The Equation Set-up

We start with defining functions and constants to describe the system. We inherit functions and constants used in the previous section for the polymer melt. Since we have two layers of the solvent, we need to have two sets of

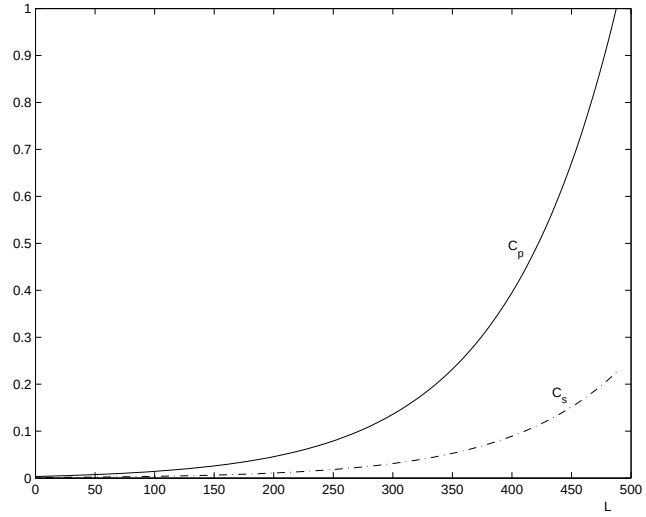

Figure 4.2: Change of Concentrations

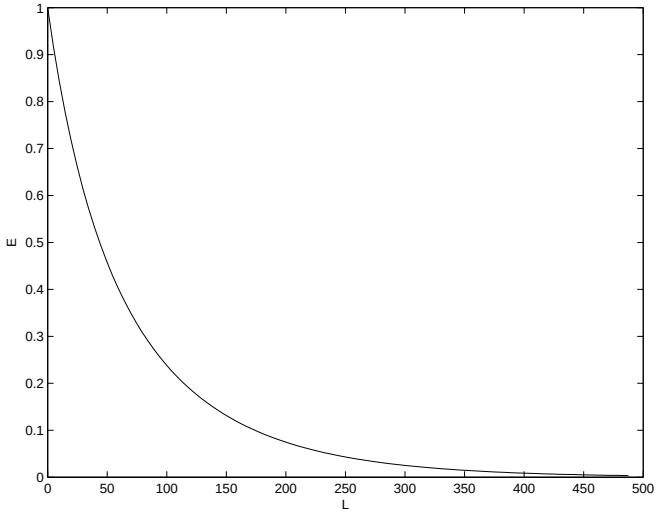

Figure 4.3: E, Effective Removal Ratio of L.

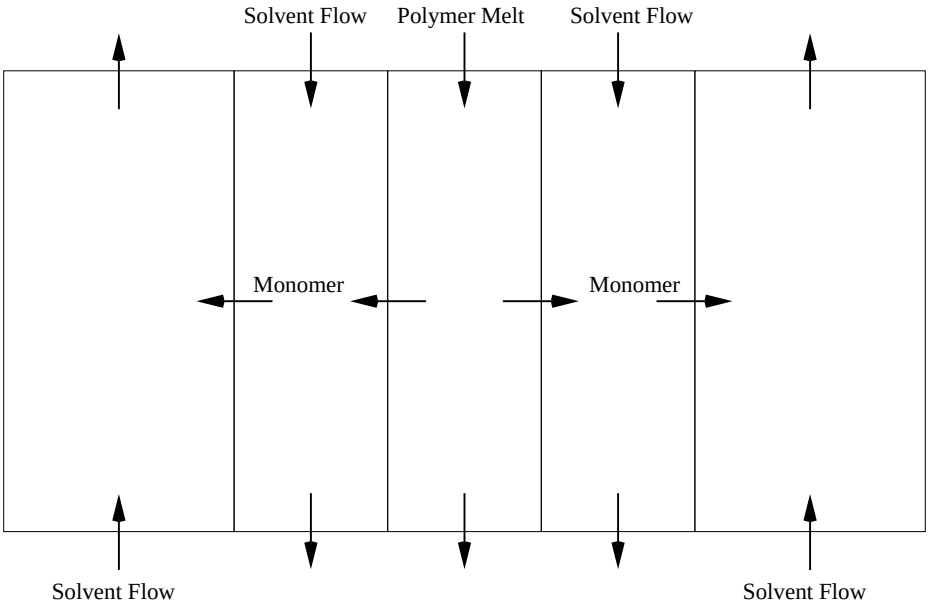

Figure 4.4: Schematic Picture of 3 Zone Model.

functions and constants used for the solvent. We denote functions and constants for the inner layer of the solvent by symbols used in the previous section with subscript 1 and for the outer layer by those with subscript 2. For example, the concentration of the inner solvent is denoted by $C_{s,1}$ while that of the outer solvent is denoted by $C_{s,2}$.

Just like the simple system, from the polymer melt, the impurity or the residual monomer moves into the inner layer of the solvent. From time t to time $t + \triangle t$, the amount

$$v_p \cdot \triangle t \cdot C_p(t,h) \cdot A_p = v_p \cdot \triangle t \cdot C_p(t,h) \cdot \pi r^2$$

of the mass of the polymer melt comes in at height h and the amount

$$v_p \cdot \triangle t \cdot C_p(t,h-\triangle h) \cdot A_p = v_p \cdot \triangle t \cdot C_p(t,h-\triangle h) \cdot \pi r^2$$

of the mass of the polymer melt comes out at height $h - \triangle h$. And in time $\triangle t$, the polymer melt loses to the solvent some amount of residual monomers that is

$$j_{mass} \cdot \triangle t \cdot (\text{Area of Exchange}) = k_1(C_p(t,h) - p\,C_{s,1}(t,h)) \cdot \triangle t \cdot 2\pi r \cdot \triangle h,$$

where k_1 is the exchange efficiency constant of the polymer melt and the solvent. Due to the conservation law of mass, we have the following:

$$\begin{aligned}
\triangle M_p &= M_p(t+\triangle t) - M_p(t) \\
&= v_p \cdot \triangle t \cdot C_p(t,h) \cdot A_p - v_p \cdot \triangle t \cdot C_p(t,h-\triangle h) \cdot A_p \\
&\quad - j_{mass} \cdot \triangle t \cdot (\text{Area of Exchange}) \\
&= v_p \cdot \triangle t \cdot C_p(t,h) \cdot \pi r^2 - v_p \cdot \triangle t \cdot C_p(t,h-\triangle h) \cdot \pi r^2 \\
&\quad - k_1(C_p - p\,C_{s,1}) \cdot \triangle t \cdot 2\pi r \cdot \triangle h.
\end{aligned}$$

Dividing $\triangle M_p$ by $A_p \cdot \triangle h \cdot \triangle t$ and letting $\triangle t$ and $\triangle h$ tend to zero, we have

$$\frac{\partial C_p}{\partial t}(t,h) = v_p \frac{\partial C_p}{\partial h}(t,h) - \frac{2k_1}{r}\left(C_p(t,h) - pC_{s,1}(t,h)\right),$$

or

$$\frac{1}{v_p}\frac{\partial C_p}{\partial t}(t,h) = \frac{\partial C_p}{\partial h}(t,h) - \frac{2\pi r k_1}{f_p}(C_p(t,h) - pC_{s,1}(t,h)).$$

The inner layer of the solvent exchanges the residual monomer with the polymer melt and the outer layer. From time t to time $t+\triangle t$, the amount

$$v_{s,1} \cdot \triangle t \cdot C_{s,1}(t, h-\triangle h) \cdot A_{s,1} = v_{s,1} \cdot \triangle t \cdot C_{s,1}(t, h-\triangle h) \cdot \pi(r_1^2 - r^2)$$

of the mass of the solvent comes in at height h and the amount

$$v_{s,1} \cdot \triangle t \cdot C_{s,1}(t, h) \cdot A_{s,1} = v_{s,1} \cdot \triangle t \cdot C_{s,1}(t, h) \cdot \pi(r_1^2 - r^2)$$

of the mass of the solvent comes out at height $h - \triangle h$. And in $\triangle t$, the inner solvent layer gains from the polymer melt the amount of residual monomers that is

$$j_{mass} \cdot \triangle t \cdot \text{(Area of Exchange)} = k_1(C_p(t,h) - p\, C_{s,1}(t,h)) \cdot \triangle t \cdot 2\pi r \cdot \triangle h$$

and loses to the outer layer the amount

$$j_{mass} \cdot \triangle t \cdot \text{(Area of Exchange)} = k_2(C_{s,1}(t,h) - C_{s,2}(t,h)) \cdot \triangle t \cdot 2\pi r_1 \cdot \triangle h.$$

Thus

$$\begin{aligned}
\triangle M_{s,1} &= M_{s,1}(t+\triangle t) - M_{s,1}(t) \\
&= -v_{s,1} \cdot \triangle t \cdot C_{s,1}(t, h-\triangle h) \cdot A_{s,1} + v_{s,1} \cdot \triangle t \cdot C_{s,1}(t,h) \cdot A_{s,1} \\
&\quad + j_{mass} \cdot \triangle t \cdot \text{(Area of Exchange)} - j_{mass} \cdot \triangle t \cdot \text{(Area of Exchange)} \\
&= -v_s \cdot \triangle t \cdot C_s(t, h-\triangle h) \cdot \pi(r_1^2 - r^2) + v_s \cdot \triangle t \cdot C_s(t,h) \cdot \pi(r_1^2 - r^2) \\
&\quad + k_1(C_p(t,h) - p\, C_{s,1}(t,h)) \cdot \triangle t \cdot 2\pi r \cdot \triangle h \\
&\quad - k_2(C_{s,1}(t,h) - C_{s,2}(t,h)) \cdot \triangle t \cdot 2\pi r_1 \cdot \triangle h.
\end{aligned}$$

Dividing $\triangle M_{s,1}$ by $A_{s,1} \cdot \triangle h \cdot \triangle t$ and letting $\triangle t$ and $\triangle h$ tend to zero, we have

$$\frac{\partial C_{s,1}}{\partial t}(t,h) = v_{s,1}\frac{\partial C_{s,1}}{\partial h}(t,h) + \frac{2k_1 r}{r_1^2 - r^2}\left(C_p(t,h) - pC_{s,1}(t,h)\right) - \frac{2r_1 k_2}{r_1^2 - r^2}\left(C_{s,1}(t,h) - C_{s,2}(t,h)\right),$$

or

$$\frac{1}{v_{s,1}}\frac{\partial C_{s,1}}{\partial t} = \frac{\partial C_{s,1}}{\partial h}(t,h) + \frac{2\pi r k_1}{f_{s,1}}(C_p(t,h) - pC_{s,1}(t,h)) - \frac{2\pi r_1 k_2}{f_{s,1}}(C_{s,1}(t,h) - C_{s,2}(t,h)).$$

Lastly, we look into the outer layer. From time t to time $t+\triangle t$, the amount

$$v_{s,2} \cdot \triangle t \cdot C_{s,2}(t, h-\triangle h) \cdot A_{s,2} = v_{s,2} \cdot \triangle t \cdot C_{s,2}(t, h-\triangle h) \cdot \pi(R^2 - r_1^2)$$

of the mass of the solvent comes out at height h and the amount

$$v_{s,2} \cdot \triangle t \cdot C_{s,2}(t, h) \cdot A_{s,2} = v_{s,2} \cdot \triangle t \cdot C_{s,2}(t, h) \cdot \pi(R^2 - r_1^2)$$

of the mass of the solvent comes in at height $h - \triangle h$. And in $\triangle t$, the outer solvent layer gains an amount of residual monomers that is

$$j_{mass} \cdot \triangle t \cdot \text{(Area of Exchange)} = k_2(C_{s,1}(t,h) - C_{s,2}(t,h)) \cdot \triangle t \cdot 2\pi r_1 \cdot \triangle h.$$

Thus

$$\begin{aligned} \triangle M_{s,2} &= +v_{s,2} \cdot \triangle t \cdot C_{s,2}(t, h - \triangle h) \cdot \pi(R^2 - r_1^2) - v_{s,2} \cdot \triangle t \cdot C_{s,2}(t,h) \cdot \pi(R^2 - r_1^2) \\ &\quad + k_2(C_{s,1}(t,h) - C_{s,2}(t,h)) \cdot \triangle t \cdot 2\pi r_1 \cdot \triangle h \end{aligned}$$

Dividing $\triangle M_{s,2}$ by $A_{s,2} \cdot \triangle h \cdot \triangle t$ and letting $\triangle t$ and $\triangle h$ tend to zero, we have

$$\frac{\partial C_{s,2}}{\partial t}(t,h) = -v_{s,2}\frac{\partial C_{s,2}}{\partial h}(t,h) + \frac{2r_1k_2}{R^2 - r_1^2}\left(C_{s,1}(t,h) - C_{s,2}(t,h)\right),$$

or

$$-\frac{1}{v_{s,2}}\frac{\partial C_{s,2}}{\partial t}(t,h) = \frac{\partial C_{s,2}}{\partial h}(t,h) - \frac{2\pi r_1 k_2}{f_{s,2}}(C_{s,1}(t,h) - C_{s,2}(t,h)).$$

As in the 2 zone model, we have a boundary condition $C_p(t,L) = \alpha$ for all $t > 0$. Regarding $C_{s,1}$, we see that, at the top of the column, some of the solvent goes out of the system while the rest stays in the system and moves along with the polymer melt. Thus we have

$$C_{s,1}(t,L) = C_{s,2}(t,L) \qquad \text{for all } t > 0.$$

Lastly, since mass of residual monomers in the inner solvent layer must be equal to mass of residual monomers in the outer solvent layer,

$$(r_1^2 - r^2)v_{s,1}C_{s,1}(t,0) = (R^2 - r_1^2)v_{s,2}C_{s,2}(t,0).$$

Therefore, we have

$$\begin{cases} \dfrac{1}{v_p}\dfrac{\partial C_p}{\partial t}(t,h) &= \dfrac{\partial C_p}{\partial h}(t,h) - \dfrac{2\pi r k_1}{f_p}\left(C_p(t,h) - pC_{s,1}(t,h)\right), \\ \dfrac{1}{v_{s,1}}\dfrac{\partial C_{s,1}}{\partial t}(t,h) &= \dfrac{\partial C_{s,1}}{\partial h}(t,h) + \dfrac{2\pi k_1 r}{f_{s,1}}\left(C_p(t,h) - pC_{s,1}(t,h)\right) \\ &\quad - \dfrac{2\pi r_1 k_2}{f_{s,1}}\left(C_{s,1}(t,h) - C_{s,2}(t,h)\right), \\ -\dfrac{1}{v_{s,2}}\dfrac{\partial C_{s,2}}{\partial t}(t,h) &= \dfrac{\partial C_{s,2}}{\partial h}(t,h) + \dfrac{2\pi r_1 k_2}{f_{s,2}}\left(C_{s,1}(t,h) - C_{s,2}(t,h)\right), \end{cases}$$

with the boundary conditions $C_p(t,L) = \alpha$, $(r_1^2 - r^2)v_{s,1}C_{s,1}(t,0) = (R^2 - r_1^2)v_{s,2}C_{s,2}(t,0)$, $C_{s,1}(t,L) = C_{s,2}(t,L)$ for all $t > 0$.

4.2.2 Steady-State Phenomena

Quantitative Analysis

We denote the steady-state concentrations of residual monomers in the polymer melt, the inner layer and the outer layer by $C_{p,\infty}$, $C_{s,1,\infty}$ and $C_{s,2,\infty}$, respectively. So we have to solve the following system of first-order ordinary

differential equations:

$$\begin{cases} 0 &= \dfrac{dC_{p,\infty}}{dh}(t,h) - \dfrac{2\pi r k_1}{f_p}\left(C_{p,\infty}(t,h) - pC_{s,1,\infty}(t,h)\right), \\ 0 &= \dfrac{dC_{s,1,\infty}}{dh}(t,h) + \dfrac{2\pi r k_1}{f_{s,1}}\left(C_{p,\infty}(t,h) - pC_{s,1,\infty}(t,h)\right) \\ & \quad -\dfrac{2\pi r_1 k_2}{f_{s,1}}\left(C_{s,1,\infty}(t,h) - C_{s,2,\infty}(t,h)\right), \\ 0 &= \dfrac{dC_{s,2,\infty}}{dh}(t,h) - \dfrac{2\pi r_1 k_2}{f_{s,2}}\left(C_{s,1,\infty}(t,h) - C_{s,2,\infty}(t,h)\right), \end{cases} \tag{4.3}$$

or

$$\begin{pmatrix} \dfrac{dC_{p,\infty}}{dh}(t,h) \\ \dfrac{dC_{s,1,\infty}}{dh}(t,h) \\ \dfrac{dC_{s,2,\infty}}{dh}(t,h) \end{pmatrix} = \begin{pmatrix} \dfrac{2\pi r k_1}{f_p} & -\dfrac{2\pi r k_1 p}{f_p} & 0 \\ -\dfrac{2\pi r k_1}{f_{s,1}} & \dfrac{2\pi r k_1 p + 2\pi r_1 k_2}{f_{s,1}} & -\dfrac{2\pi r_1 k_2}{f_{s,1}} \\ 0 & \dfrac{2\pi r_1 k_2}{f_{s,2}} & -\dfrac{2\pi r_1 k_2}{f_{s,2}} \end{pmatrix} \begin{pmatrix} C_{p,\infty}(t,h) \\ C_{s,1,\infty}(t,h) \\ C_{s,2,\infty}(t,h) \end{pmatrix}$$

with the boundary conditions $C_{p,\infty}(L) = \alpha$, $f_{s,1}C_{s,1,\infty}(0) = f_{s,2}C_{s,2,\infty}(0)$, $C_{s,1,\infty}(L) = C_{s,2,\infty}(L)$.

If we let

$$\mathbf{A} = \begin{pmatrix} \dfrac{2\pi r k_1}{f_p} & -\dfrac{2\pi r k_1 p}{f_p} & 0 \\ -\dfrac{2\pi r k_1}{f_{s,1}} & \dfrac{2\pi r k_1 p + 2\pi r_1 k_2}{f_{s,1}} & -\dfrac{2\pi r_1 k_2}{f_{s,1}} \\ 0 & \dfrac{2\pi r_1 k_2}{f_{s,2}} & -\dfrac{2\pi r_1 k_2}{f_{s,2}} \end{pmatrix},$$

then the solution of the above equations is of form

$$\begin{pmatrix} C_{p,\infty}(h) \\ C_{s,1,\infty}(h) \\ C_{s,2,\infty}(h) \end{pmatrix} = c_1\xi_1 + c_2\xi_2 e^{\lambda_1 h} + c_3\xi_3 e^{\lambda_2 h},$$

where λ's are eigenvalues of $\mathbf{A}$, ξ's are eigenvectors $\mathbf{A}$, and c's are constants satisfying the initial conditions.

Qualitative Analysis

The solution of (4.3) is too long to write. And, since the experimental data does not give us the values of parameters introduced for the 3 zone model, it is hard to analyze its behaviour. Here we investigate the 3 zone model using one set of parameters.

We use the same data used for analyzing the 2 zone model. The column has height 16 feet and radius 1.5 inch. The polymer melt comes into the column through 140 holes of radius 3 mil. The effective radius of the region under the influence of the polymer melt is assumed to be $2/3R$. The mass ratio of the solvent to the polymer melt is 3.95:1 and the flow rate of the solvent is 104 gram per minute. Let $E = 60/17581$.

Figures 4.5, 4.6 and 4.7 show the change of concentrations for different k_1 and k_2. It is hard to tell which choice of k_1 and k_2 fits the data. Figure 4.8 shows the behaviour of E's with different sets of data.

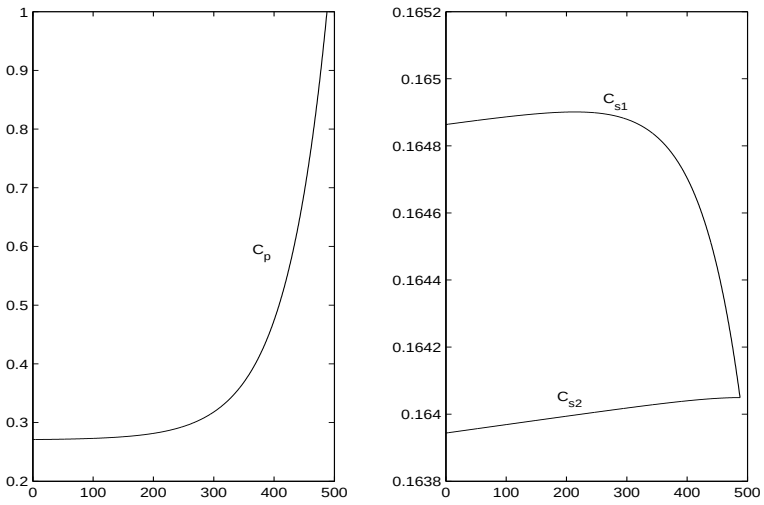

Figure 4.5: $k_1 = 1.5$ and $k_2 = 0.8$.

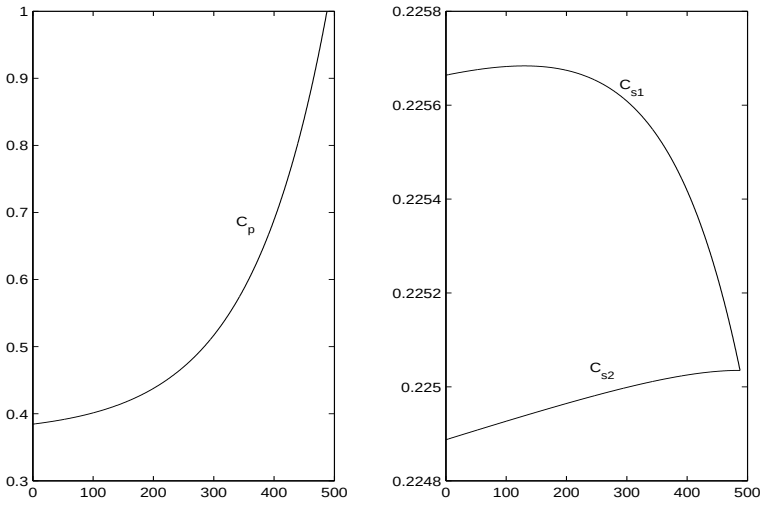

Figure 4.6: $k_1 = 0.8$ and $k_2 = 1.5$.

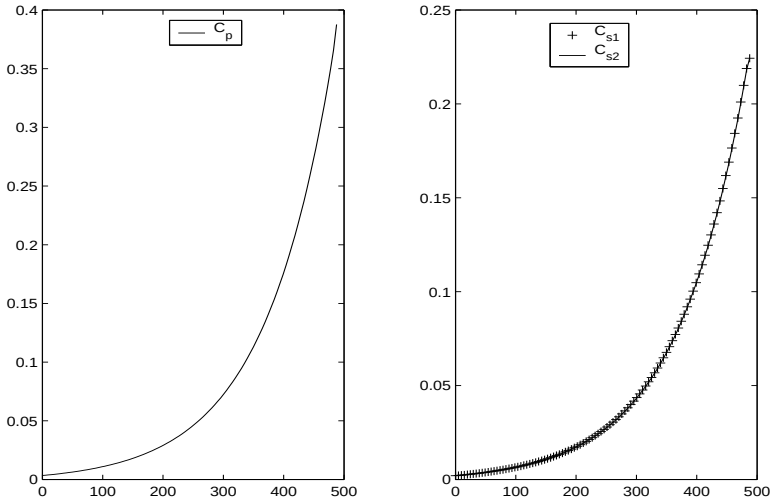

Figure 4.7: $k_1 = 55$ and $k_2 = 7293$.

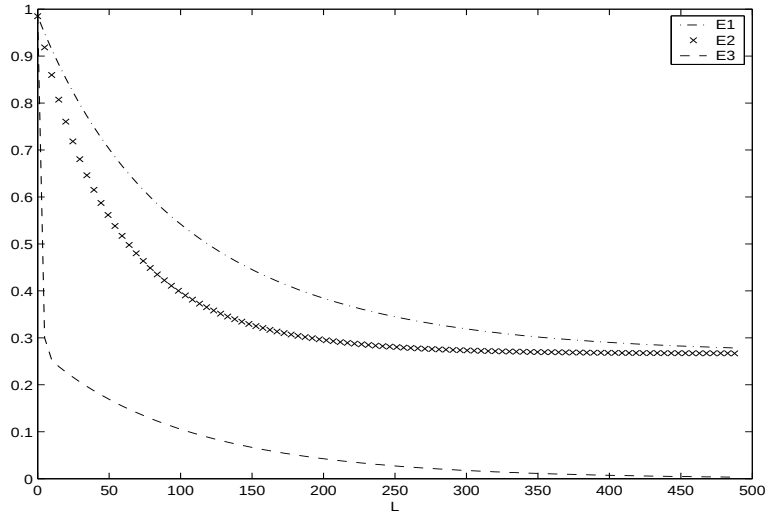

Figure 4.8: Behaviour of E.

4.3 Conclusion

While the 2 zone model is simple to analyze and fits the data quite well, it is not easy to analyze the 3 zone model. We speculate that the difficulty lies in the complexity of the solution as well as insufficient data for the model. Regarding the data, it is hard to determine the effective radius of the region under the influence of the polymer melt, and it might affect the analysis significantly. At this stage, it is inappropriate to judge the validity of the model. However, we infer that the 3 zone model is ingenious since it is more appealing to our intuition.

Chapter 5

Solar Car Racing Strategies

Participants: Fadil Santosa (Mentor), Hongbin Guo, Christian Ketelsen, Ilona Kletskin, Qingguo Li, Alfonso Limon, Yuriy Mileyko.

PROBLEM STATEMENT: An optimal solar car racing strategy, one which will account for different racing rules and track conditions, must be formulated.

In today's world of diminishing natural energy resources, advances in technology and scientific innovation have fuelled researchers' interest in exploring the potential of solar cars. In recent years, these researchers have expanded their science labs to include the race track. Around the world, teams are constantly trying to develop new materials, aerodynamic designs, and electrical systems that will help them win the various solar car races held globally [1]. These races vary in their objectives from maximizing distance over a prescribed time period to maximizing average speeds over the course of a multi-day race. The race cars begin with a fully charged battery, and must rely on additional energy absorbed through the solar panels during the race. While the design and construction process is important, a racing team cannot hope for success without a well developed racing strategy that takes into consideration such things as weather and topography.

Racing teams must adhere to several constraints including total combined weight of vehicle and driver and maximum battery capacity. Apart from these initial specifications, the problem then becomes an exercise in optimal control in order to determine the optimal speed that should be used at any given time to maximize overall performance. In previous races, strategies have included maintaining a constant speed throughout the race under clear skies and flat terrain, but either increasing or decreasing speed under cloud cover. The question arises then, as to whether or not these strategies are actually the most desirable; would it perhaps be preferable to drive at a considerably faster speed, and then stop to recharge the batteries? Under cloudy conditions, is it better to try to "outrun" the cloud as quickly as possible, or to drive slowly, since available solar power is reduced within this region? In this paper, we will explore the answers to these questions regarding various racing scenarios.

5.1 Introduction

The paper has been organized into six sections and two appendices. The five sections that following consist of a solar car model, racing strategies dictated by optimal control theory, numerical results for both cloudy and sunny days, conclusions, and future work. The appendices contain theorems important to this paper and a series of Matlab m-files used to compute the numerical results.

5.2 Model

As in all types of racing, the main objective in the majority of solar car races is to attain the maximum average speed throughout. For solar cars, the ability to achieve this goal is directly related to the amount of energy in the system at any given time. Solar cars begin a race with an initial amount of energy stored in their batteries, an amount prescribed by the rules of the particular race. Cars must then run the course of the race and, without exceeding battery capacity, obtain all additional energy from the sun through their photovoltaic panels. Assuming a single day race beginning shortly after sunrise, all additional solar power generation including recharging must take place during race time.

In the formulation of the model, several additional simplifying assumptions are made. First, because a regenerative braking system (one that allows braking to produce energy) is not a standard feature of solar cars, it is assumed that the car is not equipped with such a system. Furthermore, it is assumed that the amount of energy required to accelerate the solar car is small compared to the amount of energy dissipated throughout the race day. Thus, the force necessary to accelerate the vehicle is ignored, and the acting forces that hinder the vehicle's motion are the aerodynamic drag and the rolling friction associated with the tires. Using the relation that power is the product of force and speed, and since power is being consumed only to overcome resistive forces, the following power balance is obtained:

$$P_b + P_s - v(c_r + c_a v^2) = 0$$

The aerodynamic drag scales as the square of the speed, with a coefficient, c_a, that is dependent upon such quantities as the frontal area of the car and the density of air. The force of rolling friction c_r is a function of the speed of the vehicle and its mass; for purposes of simplification, this force is assumed to be constant. The values of the coefficients are: $c_r = 36.7875, c_a = 0.05795$ [2]. The solar power function P_s depends on such things as latitude on the surface of the earth, day of the year, and time of the day. However, for any given race, latitude and day of the year will not change greatly; thus P_s is taken only as a function of the time of the day. An example of a power profile for January 1st along the equator can be seen in Figure 5.1.

This power profile shall be used for all future calculations. While the efficiency of the solar panels is accounted for in the function P_s, the transformation of energy from the battery to the drive system is assumed to be one hundred percent efficient. Because the power available from the battery P_b can be written as the negative time derivative of the energy in the battery, the power balance can be rewritten as the first equation in the following first order system. The second governing equation defines the speed.

$$\begin{aligned} \dot{E} &= P_s - v(c_r + c_a v^2) \\ \dot{x} &= v \end{aligned} \tag{5.1}$$

The racing strategy is further restricted by the conditions that the initial energy should be equal to the maximum energy that can be stored in the battery or the maximum energy allowed by the terms of the race. Likewise, in order to ensure that the most efficient use of energy is implemented, the energy in the system at the end of the race should be zero. Finally, it is clear that the energy available from the battery must be non-negative at all times. Other conditions of the system can be defined depending on the nature of the racing objective.

5.3 Strategies

The racing strategy can be influenced by many factors, such as weather conditions, racing terrain, type of race and number of racing days. We consider two types of races: fixed distance and fixed time. For the first type of race, we are trying to minimize the total time, and for the second type, we maximize the distance.

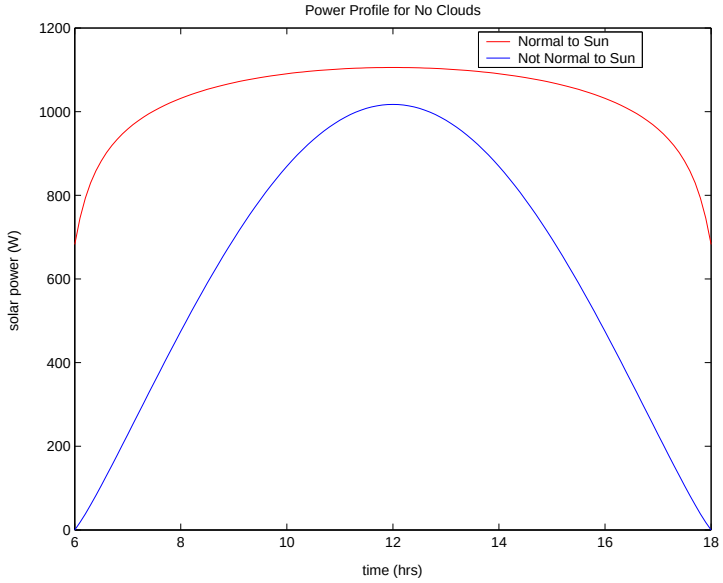

Figure 5.1: Power Profile.

In this paper we assume that we have flat terrain and a single day race. We also assume the possibility of only two weather conditions: a perfectly sunny day or a day with one cloud located at a known position.

5.3.1 Minimizing the Total Time (*no clouds*)

In this section we consider the problem of minimizing the total time of the race given its length. Using the governing equations (5.1), we arrive at the following problem:

$$\dot{E} = P_s - v(c_r + c_a v^2), \tag{5.2}$$
$$\dot{x} = v, \tag{5.3}$$

with conditions

$$\begin{aligned} E(t_0) &= E_0, \\ x(t_0) &= 0, \\ E(T) &= 0, \\ x(T) &= X, \end{aligned}$$

where T denotes the end time which we should minimize, and X denotes the length of the race. Notice that the energy must be zero at the end of the race; if it is greater than zero, we could have raced faster.

In the above problem, we have two state-variables E and x, and one control variable v. The optimal control can be found using the Pontryagin maximum principle [3], [4]. In order to use it, we should first set up the Hamiltonian. Equations (5.2), (5.3) imply that

$$H(v,t) = 1 + \lambda_1(P_s(t) - v(c_r + c_a v^2)) + \lambda_2 v$$

Applying theorem (1) in Appendix A, we get the following:

$$\dot{\lambda}_1 = 0,\ \dot{\lambda}_2 = 0$$

and

$$H(v^*(T),T) = 0,\ H(v^*(t),t) = -\int_t^T \frac{\partial H}{\partial t}(v^*(\tau),\tau)d\tau \tag{5.4}$$

where $v^*(t)$ denotes the optimal speed. The first two equations imply λ_1 and λ_2 are constants, thus $\frac{\partial H}{\partial t} = \lambda_1 \frac{dP}{dt}$. So, the integral in the equation (5.4) can be calculated

$$H(v^*(t), t) = -\lambda_1(P_s(T) - P_s(t))$$

This implies that

$$\lambda_1 P(T) = \lambda_1 v(c_r + c_a v^2) - 1 - \lambda_2 v$$

Note that the left hand side does not depend on t, so the right hand side should also be independent of t. Therefore, the speed v should be a constant. Having a constant optimal speed, we can integrate equation (5.2) directly and find that

$$E(t) = \int_{t_0}^{t} P_s(\tau)d\tau - v(c_r + c_a v^2)t + E_0$$

A constant speed also implies that $x = tv$, and applying the end condition $E(T) = 0$, we get the following nonlinear equation for v.

$$\int_{t_0}^{\frac{X}{v}} P_s(\tau)d\tau - X(c_r + c_a v^2) + E_0 = 0$$

Solving this equation gives us the optimal speed v^* for the minimum time problem.

5.3.2 Maximum Distance Problem (*No Clouds*)

Now we change our objective from minimum time to maximum distance. The end time T in this case in known and fixed. The problem itself looks similar to the previous one:

$$\dot{E} = P_s - v(c_r + c_a v^2), \quad (5.5)$$
$$\dot{x} = v, \quad (5.6)$$

with conditions

$$\begin{aligned} E(t_0) &= E_0, \\ x(t_0) &= 0, \\ E(T) &= 0, \\ x(T) &- free, \end{aligned}$$

Again, we use the Pontryagin maximum principle to find the optimal speed [3], [4]. The Hamiltonian in this case is:

$$H(v, t) = -v + \lambda_1(P_s(t) - v(c_r + c_a v^2)) + \lambda_2 v$$

We apply theorem (2) in Appendix A to obtain

$$\begin{aligned} \dot{\lambda}_1 &= 0, \\ \dot{\lambda}_2 &= 0, \\ \lambda_1(T) &= 0, \\ \lambda_2(T) &= 0. \end{aligned}$$

This implies that λ_1 and λ_2 are zero. So, the Hamiltonian becomes

$$H(v, t) = -v \quad (5.7)$$

Theorem (2) also states that the optimal speed $v^*(t)$ should minimize the Hamiltonian. Therefore equation (5.7) implies that v^* should be the largest constant in the admissible set. As in the previous section, we can integrate the equation (5.5) for E directly

$$E(t) = \int_{t_0}^{t} P_s(\tau)d\tau - v(c_r + c_a v^2)t + E_0$$

Figure 5.2: Cloud over the track.

Applying the end condition, we get the following cubic equation for v.

$$\int_{t_0}^{T} P_s(\tau)d\tau - v(c_r + c_a v^2)T + E_0 = 0$$

This equation has only one real root, which gives us the optimal speed.

Equivalence of the minimum time and the maximum distance problems

As you can see, both problems give a constant speed as the optimal strategy. This makes these two problems equivalent in the following way. Given a distance X, we can solve the minimum time problem and find the minimum time T_{min}. Then, solving the maximum distance problem with the fixed time $T = T_{min}$, we obtain the maximum distance $X_{max} = X$. This equivalence can be seen from the following simple relation: $x = vt$.

5.3.3 Race with a Single Cloud

The minimum time problem we solve in this section considers a single cloud starting at $x = x_1$ and ending at $x = x_2$ (Figure 5.2). In this case, due to the cloud, the energy from the sun decreases. In order to find the optimal speed, we used the following idea. Assume the car is before or after the cloudy period. Then, we have the same energy input as in the problem without the cloud. From the previous analysis, we know the optimal strategy is to race with a constant speed. The same strategy can also be implemented in the region under the cloud, since the power is scaled.

$$P(t) = \begin{cases} P_s(t) & 0 < t < t_1, \\ 0.25P_s(t) & t_1 < t < t_2, \\ P_s(t) & t_2 < t < T, \end{cases}$$

where t_1 and t_2 are the times the car enters and leaves the cloudy region, respectively.

Denoting the speed in each of the regions by v_1, v_2, v_3, the dynamics of the system can be written in the following way:

$$\begin{aligned} \dot{E} &= P_s - v_1(c_r + c_a v_1^2) \\ \dot{x} &= v_1, \text{ where } x \in [0, x_1]. \end{aligned} \tag{5.8}$$

$$\begin{aligned} \dot{E} &= \tfrac{1}{4}P_s - v_2(c_r + c_a v_2^2) \\ \dot{x} &= v_2, \text{ where } x \in [x_1, x_2] \end{aligned} \tag{5.9}$$

$$\begin{aligned} \dot{E} &= P_s - v_3(c_r + c_a v_3^2) \\ \dot{x} &= v_3, \text{ where } x \in [x_2, X] \end{aligned} \tag{5.10}$$

The boundary conditions are

$$E(0) = E_0, \quad E(T) = 0 \tag{5.11}$$

If we know the energy in the battery at times t_1 and t_2, we can calculate the total time T by solving one minimum time problem per region. In this way, the time T becomes a function of $E_1 = E(t_1)$ and $E_2 = E(t_2)$. The objective is to find an optimal strategy such that the total time T is minimized. Thus, the problem of finding a minimum time can be formulated as an optimization problem of this form:

$$\min_{E_1, E_2} T \quad s.t. \quad 0 < E_2 < E_1 < E_0. \tag{5.12}$$

5.4 Numerical Results

We use the approach developed in the previous section to compute the optimal speed and energy profiles for various racing conditions. To make results comparable, an initial energy of $5kWhr$ was used for all numerical computations. Since it has been shown that the minimum time and the maximum distance problems are equivalent, the calculations focus on minimizing racing time using a fixed distance of $600km$.

The results were calculated and visualized using the software package Matlab. The following Matlab subroutines were used during the computation of the optimal speed profile: fzero and fmincon. Information on these routines can be found at http://www.mathworks.com /access /helpdesk /help /toolbox /optim /optim.shtml. The routines developed for this project, also written in Matlab, can be found in Appendix B.

Under clear sky conditions it was found that the optimal racing speed is $v^* = 79.8km/hr$. The corresponding energy dissipation profile as a function of time is shown in Figure 5.3. An interesting question arises as to how the optimal racing speed varies as a function of the race length; these numerical results are illustrated in Figure 5.4. As expected, this is a decreasing function, but one should also note that it is nonlinear.

When considering a race with a single fixed cloud, the solar power profile changes due to the cloud and is illustrated in Figure 5.5. There are several important facts illustrated in the cloudy day optimal speed profile; Figure 5.6. Note that the speed profile in this case is piecewise constant and that the speed is the highest under the cloud. This suggests that one should always try to "outrun" a cloud. The energy profile is consistent with expectations and is illustrated in Figure 5.7.

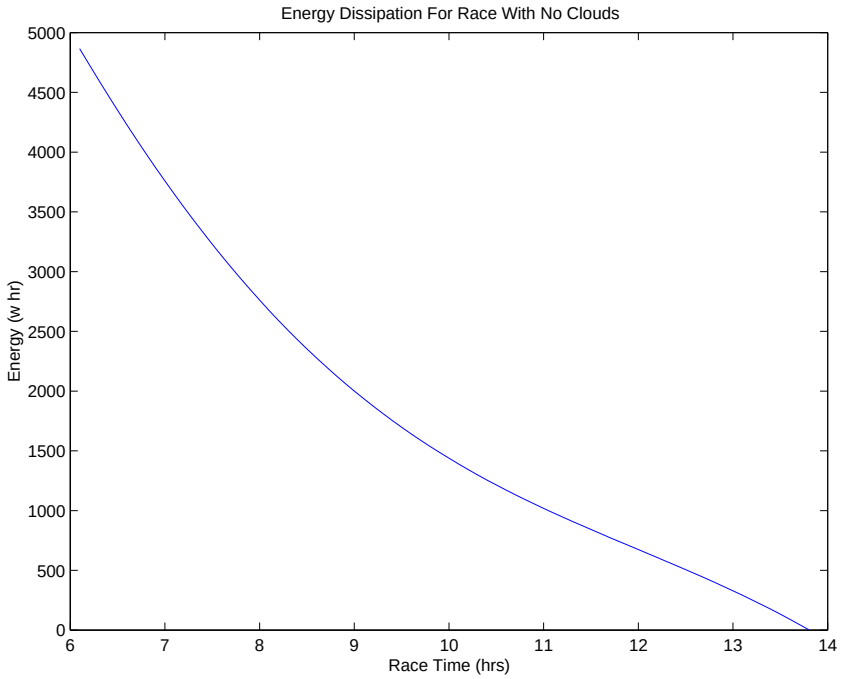

Figure 5.3: Energy dissipation for sunny day race.

5.5 Conclusion

In this paper, optimal racing strategies have been developed for solar car races of various types. For ideal racing conditions involving flat terrain and no cloud cover, it has been shown that maintaining a constant speed throughout the race is the optimal strategy for both the maximum distance and the minimum time problems. In fact, these

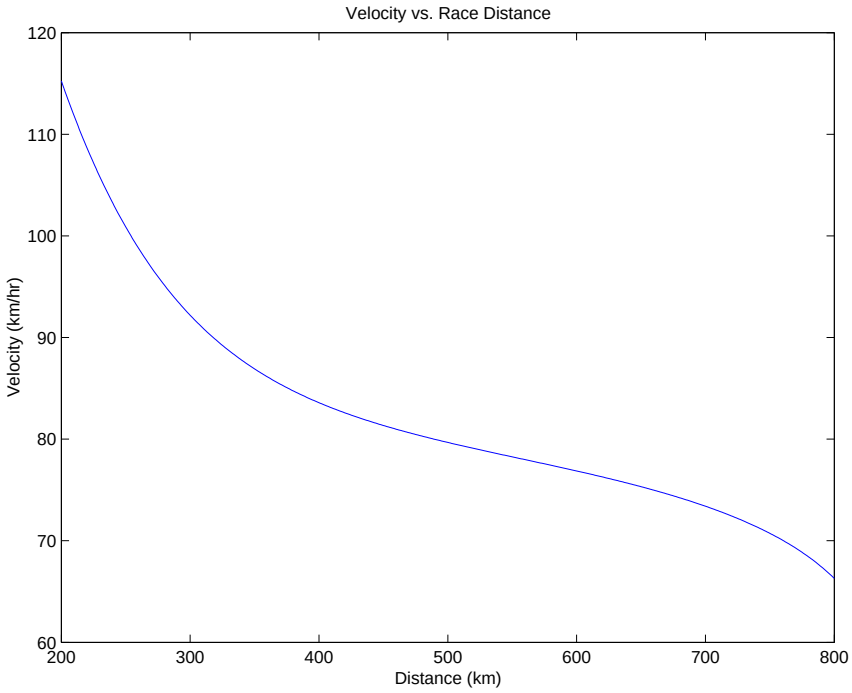

Figure 5.4: Race speed for different distance.

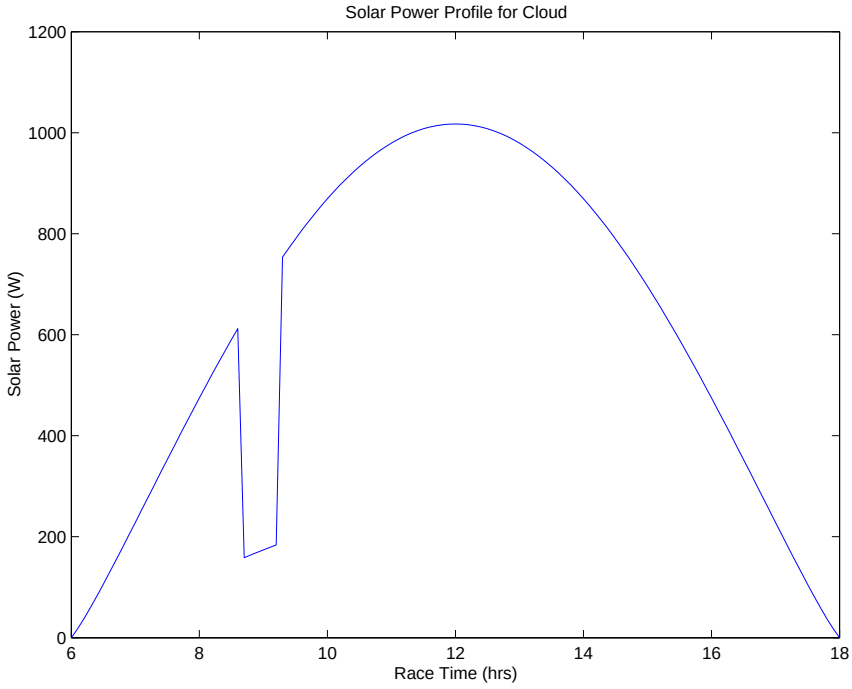

Figure 5.5: Solar power profile for cloudy day.

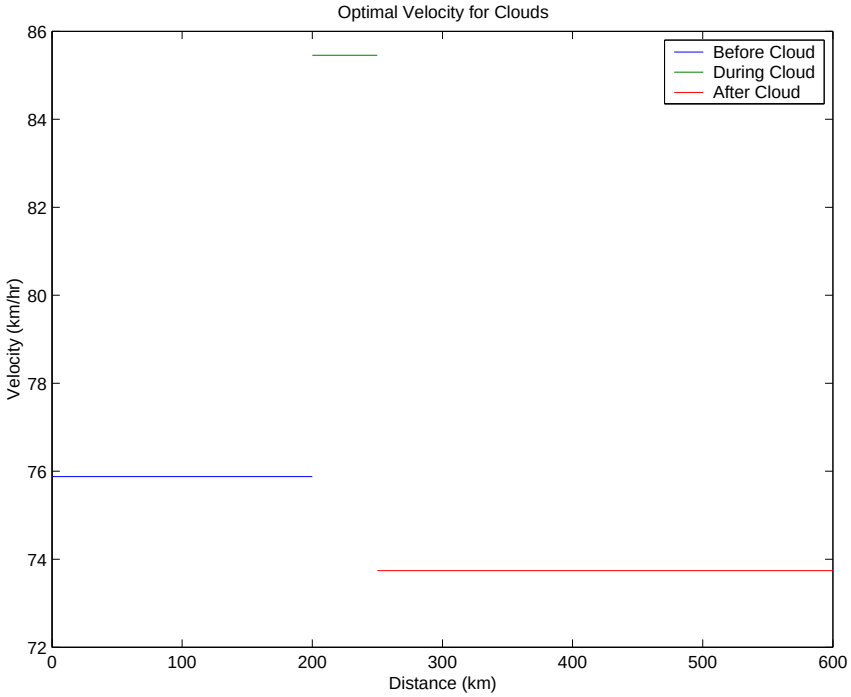

Figure 5.6: Optimal speed for cloudy day race.

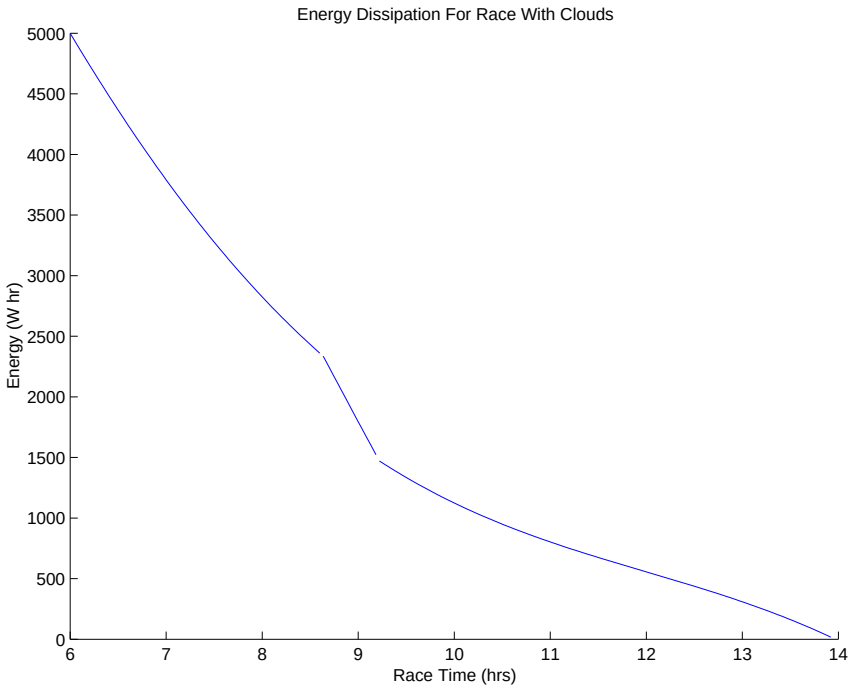

Figure 5.7: Energy dissipation for cloudy day race.

two racing objectives have been shown to be equivalent. Since poor weather is a constant concern for cars utilizing solar power, optimal strategies must also be devised for races subject to cloudy conditions. In the simplified case: given a single fixed cloud, the optimal speed profile is piecewise constant, with the greatest speed being achieved while traversing the portion of road under the cloud. While the racing conditions considered here may be simplified versions of the actual environment, the model is a useful tool for determining optimal strategies over various portions of otherwise more complicated environment.

5.6 Future Work

In actual races, numerous road, terrain, and weather conditions impact the selection of an optimal strategy. Some of the most critical concerns are the possibility of multi-day races, sporadic cloud cover, and non-level landscape. Since multi-day races are simply a combination of single-day races, the model can be easily extended to account for the possibility of racing over several days. Likewise, the generalization to multiple fixed clouds is straightforward, and extending the algorithm to consider sporadic cloud cover would involve the implementation of a stochastic element. One of the non-trivial extensions involves topography of the raceway; while our assumption of no-cost acceleration is valid for the flat terrain conditions, the inclusion of an acceleration term in the power balance becomes necessary for non-level road.

Thus far, the implication has been that all strategic decisions are made prior to race time; however, in actuality, the most successful racing strategies will be those which can be adapted in real time to account for various terrain and weather conditions as they appear. The implementation of this dynamical programming approach, together with extensions accounting for varying weather and landscape in the existing model, should provide a more robust optimal strategy.

Appendix A: Theorems

Theorem 1 ***The Minimum Principle: Time-dependent Problem (Fixed End Point)***

Let $\boldsymbol{u}^*(t)$ be an admissible control which transfers $(\boldsymbol{x}_0, t_0)$ to $S = \{\boldsymbol{x}_1\} \times (T_1, T_2)$. Let $\boldsymbol{x}^*(t)$ be the trajectory corresponding to $\boldsymbol{u}^*(t)$, originating at $(\boldsymbol{x}_0, t_0)$, and meeting S for the first time at t_1 [that is, $\boldsymbol{x}^*(t_1) = \boldsymbol{x}_1$]. In order that $\boldsymbol{u}^*(t)$ be optimal, it is necessary that there exist a function $\boldsymbol{p}^*(t)$ such that:

1. *$\boldsymbol{p}^*(t)$ corresponds to $\boldsymbol{u}^*(t)$ and $\boldsymbol{x}^*(t)$ so that $\boldsymbol{p}^*(t)$ and $\boldsymbol{x}^*(t)$ are solutions of the canonical system*

$$\begin{aligned} \boldsymbol{x}^*(t) &= \frac{\partial H}{\partial \boldsymbol{p}}[\boldsymbol{x}^*(t), \boldsymbol{p}^*(t), \boldsymbol{u}^*(t), t] \\ \boldsymbol{p}^*(t) &= -\frac{\partial H}{\partial \boldsymbol{x}}[\boldsymbol{x}^*(t), \boldsymbol{p}^*(t), \boldsymbol{u}^*(t), t] \end{aligned}$$

   satisfying boundary conditions

$$\boldsymbol{x}^*(t_0) = \boldsymbol{x}_0 \quad \boldsymbol{x}^*(t_1) = \boldsymbol{x}_1$$

2. *The function $H[\boldsymbol{x}^*(t), \boldsymbol{p}^*(t), \boldsymbol{u}, t]$ has an absolute minimum as a function of $\boldsymbol{u}$ over Ω at $\boldsymbol{u} = \boldsymbol{u}^*(t)$ for t in $[t_0, t_1]$; that is,*

$$\min_{\boldsymbol{u} \in \Omega} H[\boldsymbol{x}^*(t), \boldsymbol{p}^*(t), \boldsymbol{u}, t] = H[\boldsymbol{x}^*(t), \boldsymbol{p}^*(t), \boldsymbol{u}^*(t), t]$$

   or, equivalently,

$$H[\boldsymbol{x}^*(t), \boldsymbol{p}^*(t), \boldsymbol{u}^*(t), t] \leq H[\boldsymbol{x}^*(t), \boldsymbol{p}^*(t), \boldsymbol{u}, t] \quad \textit{for all} \quad \boldsymbol{u} \quad \textit{in} \quad \Omega$$

3. *The function $H[\boldsymbol{x}^*(t), \boldsymbol{p}^*(t), \boldsymbol{u}^*(t), t]$ satisfies the relations*

$$H[\boldsymbol{x}^*(t), \boldsymbol{p}^*(t), \boldsymbol{u}^*(t), t] = -\int_t^{t_1} \frac{\partial H}{\partial t}[\boldsymbol{x}^*(\tau), \boldsymbol{p}^*(\tau), \boldsymbol{u}^*(\tau), \tau] d\tau$$

$$H[\boldsymbol{x}^*(t_1), \boldsymbol{p}^*(t_1), \boldsymbol{u}^*(t_1), t_1] = 0$$

Theorem 2 ***The Minimum Principle: Fixed-terminal-time Problem***

Let $\boldsymbol{u}^*(t)$ be an admissible control which transfers $(\boldsymbol{x}_0, t_0)$ to:

a. The target set $S = \{\boldsymbol{x}_1\} \times \{t_1\}$ or

b. The target set $S = S_1 \times \{t_1\}$ or

c. The target set $S = R_n \times \{t_1\}$

Let $\boldsymbol{x}^*(t)$ be the trajectory corresponding to $\boldsymbol{u}^*(t)$, originating at $(\boldsymbol{x}_0, t_0)$, and meeting S at t_1. In order that $\boldsymbol{u}^*(t)$ be optimal, it is necessary that there exist a function $\boldsymbol{p}^*(t)$ such that:

1. *$\boldsymbol{p}^*(t)$ corresponds to $\boldsymbol{u}^*(t)$ and $\boldsymbol{x}^*(t)$ so that $\boldsymbol{p}^*(t)$ and $\boldsymbol{x}^*(t)$ are solutions of the canonical system*

$$\begin{aligned} \boldsymbol{x}^*(t) &= \frac{\partial H}{\partial \boldsymbol{p}}[\boldsymbol{x}^*(t), \boldsymbol{p}^*(t), \boldsymbol{u}^*(t), t] \\ \boldsymbol{p}^*(t) &= -\frac{\partial H}{\partial \boldsymbol{x}}[\boldsymbol{x}^*(t), \boldsymbol{p}^*(t), \boldsymbol{u}^*(t), t] \end{aligned}$$

   satisfying boundary conditions

   (i) $\boldsymbol{x}^*(t_0) = \boldsymbol{x}_0 \quad \boldsymbol{x}^*(t_1) = \boldsymbol{x}_1$

   (ii) $\boldsymbol{x}^*(t_0) = \boldsymbol{x}_0 \quad \boldsymbol{x}^*(t_1) \in S_1$

   (iii) $\boldsymbol{x}^*(t_0) = \boldsymbol{x}_0 \quad \boldsymbol{x}^*(t_1)$ *free*

according to the target set, S is of the form (a),(b), or (c).

2. *The function $H[\boldsymbol{x}^*(t), \boldsymbol{p}^*(t), \boldsymbol{u}, t]$ has an absolute minimum as a function of $\boldsymbol{u}$ over Ω at $\boldsymbol{u} = \boldsymbol{u}^*(t)$ for t in $[t_0, t_1]$; that is,*

$$\min_{\boldsymbol{u}\in\Omega} H[\boldsymbol{x}^*(t), \boldsymbol{p}^*(t), \boldsymbol{u}, t] = H[\boldsymbol{x}^*(t), \boldsymbol{p}^*(t), \boldsymbol{u}^*(t), t]$$

or, equivalently,

$$H[\boldsymbol{x}^*(t), \boldsymbol{p}^*(t), \boldsymbol{u}^*(t), t] \leq H[\boldsymbol{x}^*(t), \boldsymbol{p}^*(t), \boldsymbol{u}, t] \quad \textit{for all} \quad \boldsymbol{u} \quad \textit{in} \quad \Omega$$

3. *The vector $\boldsymbol{p}^*(t_1)$ is:*

   (i) Transversal to S_1 at $\boldsymbol{x}^*(t_1)$

   (ii) Zero vector; that is, $\boldsymbol{p}^*(t_1) = 0$

   according to the set, S is of the form (b) or (c).

Appendix B: Optimal Speed Algorithm

The scripts and functions created to calculate the optimal speed profile have been separated into two sections: No Clouds Over the Track and One Cloud Over the Track. The first assumes there are no clouds, so the amount of energy collected by the solar panels is only affected by the time of day. The second assumes there is one cloud over the race course, and its position is known and fixed.

No Clouds Over the Track

In this section, the race is assumed to be "ideal". Ideal is defined in the following way: there are no clouds to diminish the energy available to the solar car's collector and the energy required to accelerate the solar car is negligible, implying a level race track. The subsections include the necessary Matlab m-files to compute and visualize the optimal speed profiles with these assumptions. The code segments are organized in the following way:

Scripts and Functions

en_plot.m	—	script: calculates optimal speed and plots results.
set_vars.m	—	script: sets global variables for important constants.
eq_fun.m	—	function: calculates energy given a velocity.
int_power.m	—	function: calculates energy collected by the solar array.
array_power.m	—	function: calculates power generated by the solar array.
energy.m	—	function: calculates energy given a race time.

en_plot.m

```
% This script calculates optimal velocity for a race with no hills, no clouds.
clear;clc;close all;

% initializing physical parameters
set_vars;
% find optimal v st. v=constant and E(T) ≈ 0
V=fzero(@eq_fun,20);
% total time required to finish the race in hours
T=X/V/3600;
```

```
% results
% ———————————————————————————
% optimal velocity
{'optimal velocity = ', V*3.6, 'km/hr'}

% energy dissipation plot
dt=0.5; t=t0+dt:dt:T+t0;
y=energy(t);
plot(t,y)
ylabel('Energy (w hr)')
xlabel('Race Time (hrs)')
title('Energy Dissipation During the Race')
```

set_vars.m

```
% This function sets global variables for various physical parameters
clear;clc;close all;

% setting global variables
global alpha;
global tau;
global Cf;
global Ca;
global X;
global E0;
global t0;
global V;

% variables used for solar power, friction and drag coefficient, and race parameters
alpha=0.2;      % exponent in array_power function
tau=0.4;        % Parameter in array_power function

Cf=36.787;      % Cf=m*g*Cr; m=250(kg), g=9.81(m/s^2), Cr=0.005*3
Ca=0.0579;      % Ca=0.5*rho*A*Cd; rho=1.22(kg/m^3), A=1(m^2), Cd=0.095

X=600e3;        % length of one day race (m)
E0=5e3;         % initial energy in battery (watt*hr)
t0=6;           % start of race (hr)
```

eq_fun.m

```
function y=eq_fun(v)
% function y=eq_fun(v)
%
% calculates energy given a velocity; routine is used in conjuction with
% fzero to find the optimal velocity given that y(T)=0 and v=const.
%
% Inputs:
% v - solar car velocity
%
% Outputs:
% y - amount of energy created and dissipated while racing
```

```
global t0;      % initial race time
global X;       % length of race
global E0;      % initial energy in battery
global Cf;      % friction coefficient for solar car
global Ca;      % drag coefficient for solar car

% unit change (seconds to hours)
unit = 1/3600;

% energy balance:
% y = energy stored + energy collected from sun - energy used to race
y = E0 + int_power(t0,X./v*unit+t0) - X.*(Cf+Ca.*v.^ 2)*unit;
```

int_ power.m

```
function y=int_ power(t_i,t_f)
% function y=int_ power(t_i,t_f)
%
% calculates energy collected by the solar array from t_ i to t_ f
%
% Inputs:
% t_ i - initial time of day in hours: range 6 < t < 18
% t_ f - final time of day in hours: range 6 < t < 18
%
% Outputs:
% y - amount of energy collected from t_i to t_f
%
% Note:
% t_ i < t_ f and can be vectors

% initializing array
y=zeros(1,length(t_ f));

% taking into account t_ i and t_ f can be vectors
for k=1:length(t_ f)
    % calculating the integral of the array_power
    y(k)=quadl(@array_ power,t_ i,t_ f(k));
end;
```

array_ power.m

```
function [Pa_ h,Pa_ n]=array_ power(t)
% function [Pa_ h,Pa_ n]=array_ power(t)
%
% calculates power generated by the solar array at a specific time of day
%
% Inputs:
% t - time of day in hours: range 6 < t < 18 {t can be a vector}
%
% Outputs:
% Pa_ h - power generated while car is racing {array is not perp. to sun}
% Pa_ n - power generated while car is recharging {array is normal to sun}
```

```
global alpha;
global tau;

% defining physical constants
phi=0;      % zenith angle
n=1;        % day of the year
T=12;       % noon (hr)
Pse=1376;   % solar flux outside of earth (w/m^ 2)
Ea=0.16;    % solar cell efficiency (%16)
A=8;        % solar array area (m^ 2)

% calculating the power generated by the solar collector
% ———————————————————————
% theta angle with respect to noon (max solar power)
theta=pi/12*((t-T));
% solar declination (depends on day of the year)
dels=-0.4093*cos((2*pi/365.24)*(n+10));
cos_ angle=sin(phi)*sin(dels)+cos(phi)*cos(dels)*cos(theta);
% amount of atmospheric interference
AM=1./(cos_ angle+0.50572*(96.08-acos(cos_angle)*180/pi).^ (-1.6364));

% solar flux collected by array while recharging {array is normal to sun}
Ps_ n=(Pse*tau.^ ((AM.^ alpha)/2));
% solar flux collected by array while racing
Ps_ h=(Pse*tau.^ ((AM.^ alpha)/2)).*cos_angle;

% outout power by array while recharging {array is normal to sun}
Pa_ n=Ps_ n*Ea*A;
% outout power by array while racing
Pa_ h=Ps_ h*Ea*A;
```

energy.m

```
function y=energy(t)
% function y=energy(t)
%
% calculates energy given a race time; routine is used to confirm that the
% final energy y(T) ≈ 0 and that for t < T the energy is never zero.
%
% Inputs:
% t - race time interval 6 < ti < 18 for all i's (can be a vector)
%
% Outputs:
% y - amount of energy in the battery at some time t

global t0;  % initial race time
global X;   % length of race
global E0;  % initial energy in battery
global Cf;  % friction coefficient for solar car
global Ca;  % drag coefficient for solar car
global V;   % optimal velocity

% energy balance:
% y = energy stored + energy collected from sun - energy used to race
```

```
y = E0 + int_power(t0,t) - V.*(Cf+Ca.*V.^2).*(t-t0);
```

One Cloud Over the Track

In this section, the race is assumed to be "ideal", but the no cloud assumption is dropped. Ideal is defined in the same way as in the previous section. The energy required to accelerate the solar car continues to be assumed negligible. The subsections include the necessary Matlab m-files to compute the optimal speed profiles. The code segments are organized as:

Scripts and Functions

en_plot2.m	—	script: calculates optimal speed and plots results.
set_vars2.m	—	script: sets global variables for important constants.
eq_fun2.m	—	function: calculates energy given a velocity.
int_power.m	—	function: calculates energy collected by the solar array.
array_power.m	—	function: calculates power generated by the solar array.
energy2.m	—	function: calculates energy given a race time.
get_time.m	—	function: calculates the time it takes to cover a set distance.
total_t.m	—	function: calculates total time needed to finish the race.

en_plot2.m

```
% script calculates optimal velocity for a race with no hills, and one cloud.
clear;clc;close all;

% initializing physical parameters
set_vars2;
% find optimal v st. v=piecewise constant and E(0)>E(1)>E(2)>E(T)=0.
E=fmincon(@total_t,[3000;2000],[-1 1;1 0],[0;El],[],[],[0;0],[El;El]);

% results
% ————————————————————————————————————
% optimal velocity
v0=X0/t1/3600;
v1=(X1-X0)/t2/3600;
v2=(Xt-X1)/T/3600;
{'optimal velocities = ', v0*3.6, 'km/hr', v1*3.6, 'km/hr', v2*3.6, 'km/hr'}
% speed profile plot
dt=0.05;
figure(5.1);
plot([0;X0],[v0;v0],[X0;X1],[v1;v1],[X1;Xt],[v2;v2]);

% energy dissipation plot
figure(2);
hold on
t=th:dt:t1+th;
V=v0;
E0=El;
t0=th;
```

```
Coef=1;
plot(t,energy2(t));

t=t1+th:dt:t2+t1+th;
V=v1;
E0=E(1);
t0=th+t1;
Coef=0.25;
plot(t,energy2(t));

t=t2+t1+th:dt:T+t2+t1+th;
V=v2;
E0=E(2);
t0=th+t1+t2;
Coef=1;
plot(t,energy2(t));
hold off;
```

set_vars2.m

```
% This script sets global variables for various physical parameters
clear;clc;close all;

global alpha;
global tau;
global Cf;
global Ca;
global X;
global E0;
global t0;
global V;
global Ef;
global El;
global X0;
global X1;
global Xt;
global th;
global Coef;
global t1;
global t2;
global T;

% variables used for solar power, friction and drag coefficient, and race parameter
```

```
alpha=0.2;          % exponent in array_ power function
tau=0.4;            % Parameter in array_ power function

Cf=36.7875;         % Cf=m*g*Cr; m=250(kg), g=9.81(m/sˆ2), Cr=0.005*3
Ca=0.05795;         % Ca=0.5*rho*A*Cd; rho=1.22(kg/mˆ3), A=1(mˆ2), Cd=0.095

X=600e3;Xt=X;       % length of one day race (m)
X0=200e3;           % beginning position of the cloud
X1=250e3;           % end position of cloud
E0=5e3;El=E0;       % initial energy in battery (watt*hr)
t0=6;th=t0;         % start of race (hr)
Coef=1;             % start with no cloud (ie. 100% power)
```

eq_ fun2.m

```
function y=eq_ fun2(v)
% function y=eq_ fun(v)
%
% calculates energy given a velocity; routine is used in conjuction with
% fzero to find the optimal velocity given that y(T)=0 and v=const.
%
% Inputs:
% v - solar car velocity
%
% Outputs:
% y - amount of energy created and dissipated while racing

global t0;      % initial race time
global X;       % length of race
global E0;      % initial energy in battery
global Cf;      % friction coefficient for solar car
global Ca;      % drag coefficient for solar car
global Ef;      % amount of energy in the battery
global Coef;    % power from the sun depends on cloud cover

% unit change (seconds to hours)
unit = 1/3600;

% energy balance:
% y = energy stored + energy collected from sun - energy used to race
y = E0 - Ef + Coef*int_ power(t0,X./v*unit+t0) - X.*(Cf+Ca.*v.ˆ 2)*unit;
```

int_ power.m

```
function y=int_ power(t_i,t_f)
% function y=int_ power(t_i,t_f)
%
% calculates energy collected by the solar array from t_ i to t_ f
%
% Inputs:
% t_ i - initial time of day in hours: range 6 < t < 18
% t_ f - final time of day in hours: range 6 < t < 18
%
```

```
% Outputs:
% y - amount of energy collected from t_i to t_f
%
% Note:
% t_i < t_f and can be vectors



% initializing array
y=zeros(1,length(t_f));

% taking into account t_i and t_f can be vectors
for k=1:length(t_f)
    % calculating the integral of the array_power
    y(k)=quadl(@array_power,t_i,t_f(k));
end;
```

array_power.m

```
function [Pa_h,Pa_n]=array_power(t)
% function [Pa_h,Pa_n]=array_power(t)
%
% calculates power generated by the solar array at a specific time of day
%
% Inputs:
% t - time of day in hours: range 6 < t < 18 {t can be a vector}
%
% Outputs:
% Pa_h - power generated while car is racing {array is not perp. to sun}
% Pa_n - power generated while car is recharging {array is normal to sun}

global alpha;
global tau;

% defining physical constants
phi=0;      % zenith angle
n=1;        % day of the year
T=12;       % noon (hr)
Pse=1376;   % solar flux outside of earth (w/m^2)
Ea=0.16;    % solar cell efficiency (%16)
A=8;        % solar array area (m^2)

% calculating the power generated by the solar collector
% ————————————————————————————
% theta angle with respect to noon (max solar power)
theta=pi/12*((t-T));
% solar declination (depends on day of the year)
dels=-0.4093*cos((2*pi/365.24)*(n+10));
cos_angle=sin(phi)*sin(dels)+cos(phi)*cos(dels)*cos(theta);
% amount of atmospheric interference
AM=1./(cos_angle+0.50572*(96.08-acos(cos_angle)*180/pi).^(-1.6364));

% solar flux collected by array while recharging {array is normal to sun}
```

```
Ps_ n=(Pse*tau.^ ((AM.^ alpha)/2));
% solar flux collected by array while racing
Ps_ h=(Pse*tau.^ ((AM.^ alpha)/2)).*cos_angle;

% outout power by array while recharging {array is normal to sun}
Pa_ n=Ps_ n*Ea*A;
% outout power by array while racing
Pa_ h=Ps_ h*Ea*A;
```

energy2.m

```
function y=energy2(t)
% function y=energy(t)
%
% calculates energy given a race time; routine is used to confirm that the
% final energy y(T)≈0 and that for t < T the energy is never zero.
%
% Inputs:
% t - race time interval 6 < ti < 18 for all i's (can be a vector)
%
% Outputs:
% y - amount of energy in the battery at some time t

global t0;      % initial race time
global X;       % length of race
global E0;      % initial energy in battery
global Cf;      % friction coefficient for solar car
global Ca;      % drag coefficient for solar car
global V;       % optimal velocity
global Coef;    % power from the sun depends cloud cover

% energy balance:
% y = energy stored + energy collected from sun - energy used to race
y = E0 + Coef*int_ power(t0,t) - V.*(Cf+Ca.*V.^ 2).*(t-t0);
```

get_ time.m

```
function y=get_ time(x,t,E)
% function y=get_ time(x,t,E)
%
% Inputs:
% x - distance
% t - time
% E - energy
%
% Outputs:
% y - total time

global V;
global Ef;
global E0;
global X;
global t0;
```

```
E0=E(1);    % initial energy
Ef=E(2);    % final energy
t0=t;       % time
X=x;        % distance

% finds the optimal speed taking an intial guess of 20 m/s
V=fzero(@eq_fun2,20);

% total time required to finish the race in hours
y=X/V/3600;
```

total_time.m

```
function y=total_t(E)
% function y=total_t(E)
%
% Input:
% E - energy
%
% Output:
% total time

global X0;
global X1;
global Xt;
global El;
global th;
global Coef;
global t1;
global t2;
global T;

% calculates the time required to run first segment (no cloud)
t1=get_time(X0,th,[El E(1)]);

% calculates the time required to run second segment (cloud)
Coef=0.25;
t2=get_time(X1-X0,t1+th,E);

% calculates the time required to run third segment (no cloud)
Coef=1;
T=get_time(Xt-X1,t2+t1+th,[E(2) 0]);

% total time
y=100*(t1+t2+T);
```

Bibliography

[1] Y. SHIMIZU, Y. KOMATSU, M. TORII AND M. TAKAMURO, *Solar Car Cruising Strategy and its Supporting System*. JASE Review 19(1998), pp. 143–149.

[2] P. HOWLETT AND P. PUDNEY, *Optimal Driving Strategy for a Solar Car on a Level Road*. IMA Journal of Mathematics Applied in Business & Industry(1997), **Vol. 8**, pp. 59–81.

[3] M. ATHANS AND P.L. FALB, *Optimal Control: An Introduction to the Theory and Its Applications*. McGraw-Hill, Inc. 1966.

[4] E.R. PINCH, *Optimal Control and the Calculus of Variations*. Oxford University Press, 1993.

Chapter 6

Converting Machine Tool Measurements into a CAD Model

Participants: Robert Piché (Mentor), Samet Kadioglu, Lin Zhou, Ying Han, Tzvetalin Vassilev, Thalya Burden, Xinghua Deng.

PROBLEM STATEMENT: For a machine tool remanufacturer, the geometric data of old machines are needed in a CAD package to rebuild the machine. Measuring data "by hand" is time consuming and error prone. Therefore, our task is to design an algorithm to translate measurement data from a portable articulated measurement arm into a solid model which can be understood by a CAD package. The arm can provide global locations of points in space. The orientation of the arm is also available to determine the outward direction of the measured surface.

6.1 Introduction

The problem was introduced by a machine tool remanufacturer. They rebuild and modify existing machine tools, incorporating new technology to meet customer requirements at significantly lower cost than new machines. The first step in the machine redesign is to measure the old machine and enter its geometry into a CAD package. Doing this "by hand" is time consuming and error prone. The client would like to develop a method to improve this process.

The idea from the client is to use an articulated measurement arm to substitute the need for a human in the measurement. Therefore an algorithm is needed to translate the measurement data from the arm into a solid model which can be understood by the CAD package.

The machine parts we are dealing with are static and consist of polygons, cylinders and cones. Therefore, in order to develop an algorithm to meet the client's needs, we posed the following questions:

1. What is the minimum number of points needed to define a plane? a cylinder? a cone?

2. How are edges defined?

3. Can additional points be used to improve accuracy?

To simplify our problem, we have made the following assumptions. First we assume that the CAD package has the capability to add and subtract objects. Thus we can begin by considering convex objects only. In this manner, any non-convex object can be treated as the addition or subtraction of a series of convex objects. Second, from information gathered on the portable articulated measurement arm, we assume that the arm has 5–6 levels of freedom. Third, we suppose that, once measurements are taken, a tentative solid model will be displayed on a laptop connected to the measurement arm. Due to this supposition, the operator will be able to interact with the program.

We begin by providing solutions to the three questions aforementioned. Once this is accomplished, we are able to construct an algorithm to translate the measurements taken into a mathematical description that can then be input into a CAD package. After we have defined the algorithm, we will describe various features of the algorithm and ways it can be improved. Finally, we will summarize our project and suggest paths for future development.

6.2 Minimum Number of Points

We begin by considering the simplest geometrical object in question 1, the unbounded plane. By defining three distinct points (x_1, y_1, z_1), (x_2, y_2, z_2) and (x_3, y_3, z_3) on the plane, the plane can be determined provided the points do not lie along a straight line (figure 6.1). The equation of the plane is:

$$\begin{vmatrix} x - x_1 & y - y_1 & z - z_1 \\ x_2 - x_1 & y_2 - y_1 & z_2 - z_1 \\ x_3 - x_1 & y_3 - y_1 & z_3 - z_1 \end{vmatrix} = 0. \tag{6.1}$$

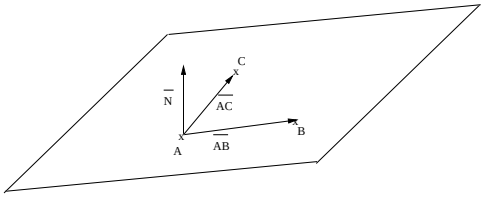

Figure 6.1: 3-D planar surface.

In addition, the arm has orientation. Thus, the outward normal of the plane can be determined. Using this, we can determine in what direction the machine lies.

For the unbounded right circular cylinder, three points, which lie along the side of the cylinder, are required to define the surface. For example, if we consider a right cylinder along the z-coordinate, by ignoring the z-coordinates of the points, we can construct a circle in the xy-plane and expand it in the z direction (figure 6.2). For this case, since the circle is on the xy-plane, we have the equation $(x - x_0)^2 + (y - y_0)^2 = r^2$. Therefore three points are enough to determine the three parameters, the center (x_0, y_0) and radius r.

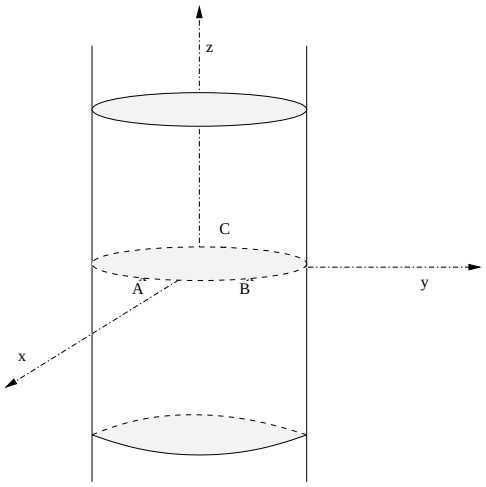

Figure 6.2: Unbounded right cylinder constructed by three points.

For the unbounded right circular cone, we have the following equation:

$$(x - x_0)^2 + (y - y_0)^2 = a^2(z - z_0)^2. \tag{6.2}$$

We notice that the equation has four parameters x_0, y_0, z_0 and a. Therefore, we require at least four distinct points to define such an object.

For general unbounded quadratic surfaces, we have the following equation with ten parameters

$$ax^2 + by^2 + cz^2 + dxy + eyz + fxz + gx + hy + iz + j = 0. \tag{6.3}$$

Thus, ten distinct points are needed to define these surfaces.

6.3 Defining Edges (Level Set Method)

In order to define the solid object, we must first find the edges of the object. Most edges on the machine tools can be identified as the intersections of two planes, which is represented by a line. For example, if we define two planes by their implicit functions as $f_1(x, y, z) = 0$ and $f_2(x, y, z) = 0$, respectively, then the equation of the edge line can be expressed as:

$$\begin{cases} f_1(x, y, z) = 0 \\ f_2(x, y, z) = 0. \end{cases} \tag{6.4}$$

Once this is obtained, we must eliminate all extraneous sections of the planes. The intersection line separates each plane into two half-planes. One half-plane contains the measurement points taken by the arm, the other half-plane does not contain material (metal). The section without the mesurement points is the section we wish to eliminate. To provide a mathematical procedure of deciding which part of the plane is unwanted, we borrow some ideas from the "Level Set Method" which is first introduced in [1] for interface propagation. The premise behind this procedure is illustrated by figure 6.3. Consider plane 1. When we plug in any of the measurement points of plane 1 into the implicit function $f_2(x, y, z)$, we will obtain a number. If the number is positive, we will accept the sign of f_2 as positive; if the number is negative, we will mutiply f_2 by -1. By this convention, f_2 will always

be positive for the area of plane 1 where the solid object remains and negative over the area we wish to eliminate. Therefore, the part of plane 1 that we eliminate satisfies:

$$\begin{cases} f_1(x,y,z) = 0 \\ f_2(x,y,z) < 0. \end{cases} \tag{6.5}$$

Similarly, $f_1(x,y,z)$ will change signs as it crosses the intersection along plane 2. This time the part we eliminate is:

$$\begin{cases} f_2(x,y,z) = 0 \\ f_1(x,y,z) < 0. \end{cases} \tag{6.6}$$

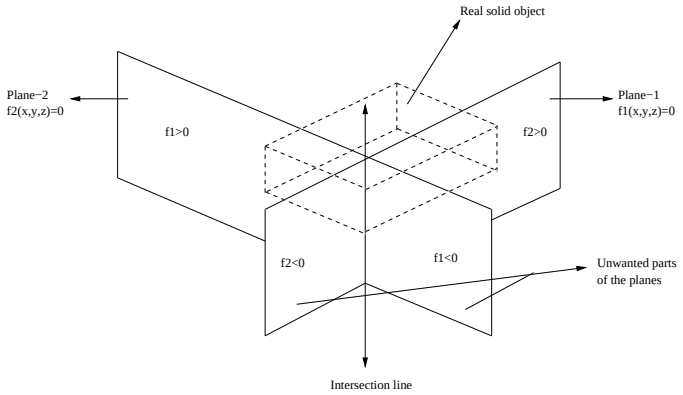

Figure 6.3: Convex solid object construction by the Level Set Method.

Note that this method will work for convex objects only. In general, for the i^{th} surface which is defined by $f_i(x,y,z) = 0$, we will eliminate all sections such that $f_j(x,y,z) < 0$ and $f_i(x,y,z) = 0$. This procedure will give the edges of the solid object.

Finally, the intersections of $f_j(x,y,z) \geq 0$ will yield the mathematical description of the solid object.

6.4 Using Additional Points

Up to now, we have only used the minimum number of points needed to define each surface. To improve accuracy, we can include additional data points. In general, we apply least squares fitting to do so. Here we will discuss the application of this method to different shapes.

For a plane perpendicular to one of the axes, the equation of the plane can be simply written as a function of one variable, such as $x = a$, $y = b$ or $z = c$. In this case, the data we have gathered on this plane has only one coordinate that is meaningful. Then the constant in the plane expression can be obtained by taking the mean of the values of that variable, which is equivalent to the method of linear least square fitting.

For any arbitrary plane, we may use total least square fitting, which is nonlinear. In this standard method, we minimize the sum of the orthogonal distances from the sample points to the plane. The distance from a sample point $(x_i, y_i, z_i$ to the plane is expressed as:

$$d_i = \frac{Ax_i + By_i + Cz_i + D}{\sqrt{A^2 + B^2 + C^2}} \tag{6.7}$$

where the plane equation is $Ax + By + Cz + D = 0$. Therefore we are actually minimizing $\sum_{i=1}^{n} d_i^2$.
The iterative equation that we use to find the parameter $\mathbf{u} = (A, B, C, D)$ is:

$$\mathbf{u}^{k+1} = \mathbf{u}^k - \mathbf{J^{-1}}(\mathbf{u}^k)\mathbf{d}(\mathbf{u}^k) \tag{6.8}$$

where $\mathbf{J}(\mathbf{u}) = \frac{\partial d_i(\mathbf{u})}{\partial u_j}$ is the Jacobian of $\mathbf{d} = (d_1, d_2, \cdots, d_n)$ [2].

The iterative method requires an approximate solution to be an initial guess. This will be obtained from any three points on the plane. In this manner, Newton's method will converge quickly to an exact solution.

A method simliar to this can be applied to circles in order to improve the accuracy for cylindrical objects [2]. We define the distances that we wish to minimize for the circle as follows:

$$d_i = \|\mathbf{x_0} - \mathbf{x}_i\| - r \tag{6.9}$$

where $\mathbf{x_0}$ is the center, r is the radius of the circle, and $\mathbf{x}_i's$ $(i = 1, \cdots, n)$ are the measured points. The iterative method will be the same as above with the unknown parameter $\mathbf{u} = (\mathbf{x_0}, r)$. The initial guess in this case will be obtained by fitting the data to a circle using the minimization of the algebraic distance method [3], which is a non-iterative fitting.

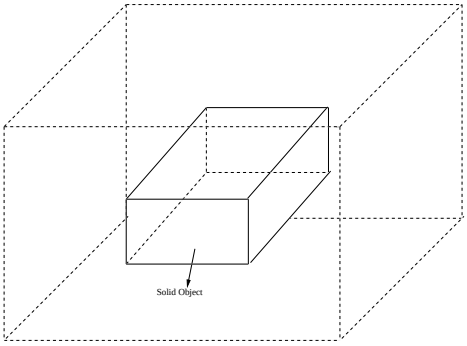

Figure 6.4: Bounded reference space.

6.5 Algorithm

Using the information provided above, we now can describe the algorithm to construct a solid convex object. We begin by setting a maximum boundary for the object (figure 6.4). This boundary can be the size of the room where the machine is located. With this convention, we eliminate the need to consider unbounded surfaces. We then prompt the operator to use the arm to mark a point as origin. This will be used to create our global coordinate system. Input also includes the type of surface (planar, cylindrical, conical). Once this information is obtained, the arm will measure at least the minimum number of data points needed to define the corresponding surface as well as the surface orientation. Depending on the surface, a subroutine is then called to determine the equation of the surface. Once a new surface is introduced, we identify all the intersections that may result between this new surface and the existing surfaces. Then a subroutine is called to define the edges and eliminate the unwanted sections by the "Level Set Method" described previously. After each addition of a surface, we repeat this procedure until the operator indicates that the object is complete. Note that after each addition of a surface, a laptop connected to the measurement arm will display the current shape that has been formed. In this manner, the operator will be able to detect any errors made, i.e. if the operator skips measuring a particular surface, he or she will not get a solid object.

Once all subobjects are defined, the CAD package will merge them to form the geometry of the machine tool being modelled. The flow chart of the algorithm is shown in figure 6.5.

6.6 Conclusions

To summarize, we have addressed basic questions in order to develop an algorithm which will translate measured data obtained from the portable measurement arm into a CAD model. The algorithm has been described in detail. The idea of constructing a convex solid object is based on the "Level Set Method". To utilize additional data points

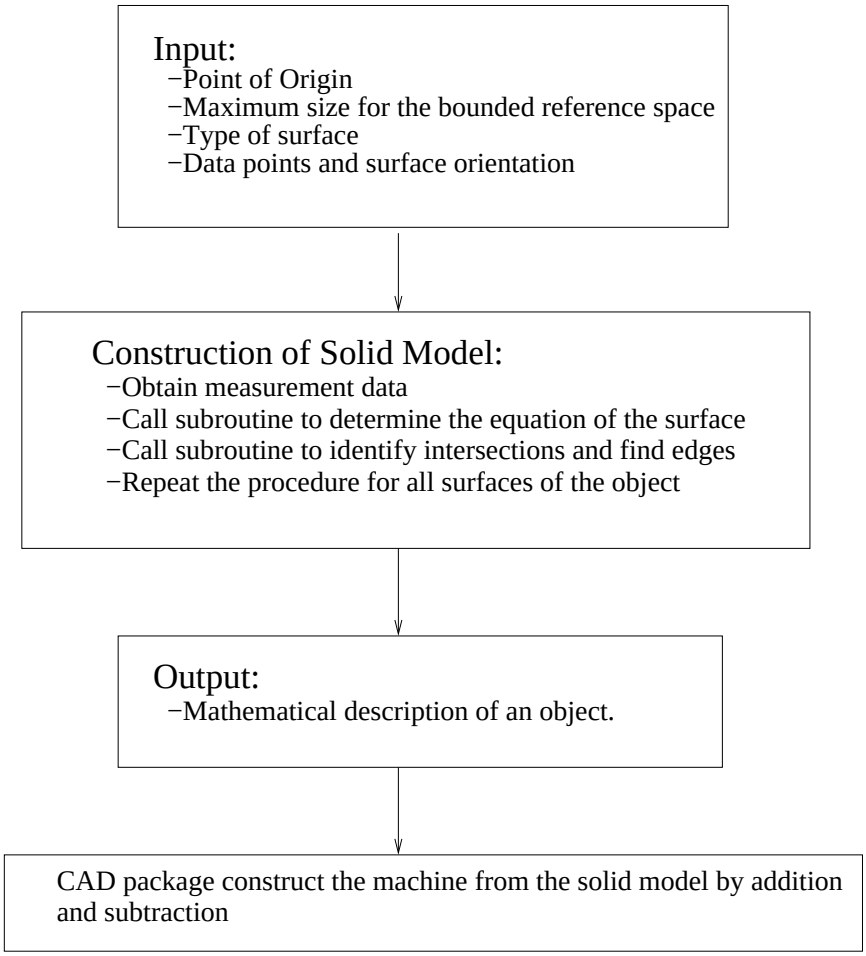

Figure 6.5: Flow chart of the algorithm.

and improve accuracy, the least square techniques are used. Algorithms for these least square fitting procedures are readily available.

6.7 Future Work

In order to expand and improve the solution to this problem, we suggest several directions. First, the algorithm should be implemented in the work environment in order to determine its efficiency and accuracy. Second, there are several ways in which the algorithm itself could be improved. For example, it can be extended so that the mathematical description of non-convex objects are defined directly. Also, the algorithm may be extended to include other geometrical surfaces, such as splines. In addition, fitting the data of general cylinders and cones can be explored. We also suggest that the client investigate other technological advances.

Bibliography

[1] Osher, S. and Sethian, J., Fronts Propagating with Curvature Dependent Speed: Algorithms Based On Hamiliton-Jacobi Formulations, J. Comp. Phys. 79, 12–49 (1988).

[2] Walter Gander, Gene H. Golub, Rolf Strebel, *Least-Squares Fitting of Circles and Ellipses.*

[3] Åke Bjorck, Solving Least Squares Problems, SIAM.

List of Participants

Organising Committee

Chris Bose	University of Victoria
Ian Frigaard	University of British Columbia
Huaxiong Huang	York University
Rachel Kuske	University of British Columbia
Jack Macki	University of Alberta
Fadil Santosa	Institute for Mathematics and its Applications

Mentors

Richard Braun	University of Delaware
Sonja Glavaski	Honeywell
David Misemer	3M
Robert Piché	Tampere University of Technology, Finland
Fadil Santosa	IMA/University of Minnesota
Emily Stone	Utah State University

Students

Qingguo Li	Simon Fraser University
Haris Widjaya	Simon Fraser University
Xinghua Deng	University of Alberta
Hongbin Guo	University of Alberta
Qian Wang	University of Alberta
Kerianne Yewchuk	University of Alberta
Shengyuan (Michael) Chen	University of British Columbia
Hua Li	University of Calgary
Andrew C. Taylor	University of Calgary
Alfonso L. Limon	Claremont Graduate University
Nadine Gärtner	Clemson University
Charles Bergeron	Ecole Polytechnique de Montreal
Samet Y. Kadioglu	Florida State University
Zhenlu Cui	Florida State University
Marcio F. Gameiro	Georgia Institute of Technology
Katharina Baamann	Georgia Institute of Technology
Fabien Mesmin Youbissi	Laval University
Olivier Dubois	McGill University
Ying Han	McGill University
Mingfei Li	Michigan State University
Lin Zhou	New Jersey Institute of Technology

Yuriy Mileyko	New Jersey Institute of Technology
Seung Y. Lee	Ohio State University
Serguei Lapin	University of Houston
Yasong Jin	University of Kansas
Yujun Wu	University of Kentucky
Thalya N. Burden	University of Kentucky
Tzvetalin S. Vassilev	University of Saskatchewan
Anuj Mubayi	University of Texas, Arlington
Mark Braverman	University of Toronto
Ilona Kletskin	University of Toronto
Christian W. Ketelsen	Washington State University
Rongsong Liu	York University

PIMS Contact Information

email: pims@pims.math.ca
http://www.pims.math.ca

- **Director: N. Ghoussoub**
  Phone: 604-822-3922
  Fax: 604-822-0883
  email: directorpims.math.ca
- **SFU-Site Director: M. Trummer**
  email: sfu@pims.math.ca
- **UAlberta-Site Director: J. Muldowney**
  email: ua@pims.math.ca
- **UBC-Site Director: D. Rolfsen**
  email: ubc@pims.math.ca
- **UCalgary-Site Director: G. Margrave**
  email: uc@pims.math.ca
- **UVic-Site Director: F. Diacu**
  email: uvic@pims.math.ca
- **UWashington-Site Director: S. P. Smith**
  email: uw@pims.math.ca