

Symbolic Analysis of Design Requirements for Control Laws

Ruth Hardy

School of Computer Science
University of St Andrews

Abstract. In this document we present three different approaches to proving a curve lies outside a region on a Nichols plot. Our aim is to enhance current control engineering techniques for analysis of control systems using symbolic computation and computational logic techniques.

1 Introduction

The purpose of this document is to describe a recent project investigating machine assisted reasoning in the context of the Mathworks' Simulink tool.

The Simulink tool provides a powerful, high-level environment for the design, analysis, and simulation of control systems, however, its design verification methods are weak. It was the aim of this project to investigate ways in which computational logic and symbolic computation techniques could be included in the existing analysis in Simulink to allow some formal verification of the design to take place.

The existing analysis techniques within Simulink involve plotting graphs, such as Nichols plots, which are then visually inspected by the user to ensure certain design requirements are met. With Nichols plots it is usually a requirement that the curve does not pass through some specified region or that it stays within a specified region. Simulink provides no methods for formally verifying the results of this analysis. Symbolic computation and computational logic applied here would allow automation, extension and greater confidence in the results of the analysis.

There are many possible approaches to the problem of proving a curve lies outside or within a particular region on a Nichols plot. The problem can be reduced to proofs that a curve $f(x)$ lies either above or below lines in particular intervals; that is $f(x) > mx + c$ or $f(x) < mx + c$ in $[a, b]$.

We investigated three different approaches to this problem to determine the feasibility of each. The approaches all used the computer algebra system Maple (see Section 3.2) to do a form of pre-processing of a control system in order to generate verification conditions. The formal verification system PVS (see Section 3.3) was then used to discharge these conditions.

In the next section (Section 2) we introduce some basic terminology and notation that will be used throughout this document. Section 3 gives some background to this project. The three approaches we examined are outlined in Section

4, Section 5 and Section 6, along with an analysis of each method. Conclusions about the relative merits of the three approaches are drawn in Section 7 and possible future work is suggested in Section 8.

2 Notation

In this section we will introduce some terminology that will be used throughout this document. Wherever possible, we have tried to use existing, common terminology.

2.1 Control engineering terms.

In control engineering there are many different representations of control systems available. A generic control system can be represented as a function of time $g(t)$ or it can be represented in the frequency domain as a Laplace transform $G(s)$ or a Fourier transform $G(j\omega)$.

The Laplace transform maps a system in the continuous time domain $g(t)$ to $G(s)$ in the frequency domain. The Laplace transform is given by the following:

$$G(s) = \int_0^{\infty} f(t)e^{-st} dt$$

where it is assumed that $g(t)$ is zero for negative values of t . The Laplace transform is often referred to as the ‘transfer function’.

The Fourier transform is another representation of a system in terms of complex frequency. It can be obtained by substituting $s = j\omega$ in the Laplace transform. The variable ω represents frequency and is therefore real-valued and non-negative, and j represents the complex constant, as is the convention in control engineering.

Nichols Plots The Nichols plot (see Figure 1) is a common technique used by control engineers in the development of control systems. They are used for the analysis of open-loop systems in the frequency domain. They allow judgements to be made about the stability and performance of a system based on the path and the shape of the plot. This is done by defining an area of the graph which the plot of the system must not enter (or conversely must not leave). The region to be avoided (referred to as the exclusion region), or the desired region for the plot varies depending on the purpose of the control system. A line forming the boundary of the exclusion region may be referred to as the exclusion line. An example of an exclusion region is shown in Figure 1 and takes the form of a hexagon centred around the $(-180, 0)$ point.

The construction of a Nichols plot is quite complex and uses the fact that if a system expressed as a Laplace transform is given a sinusoid as an input it will output a sinusoid with the same frequency but different amplitude and phase angle. A comparison of these two sinusoids will give the gain and phase-shift of

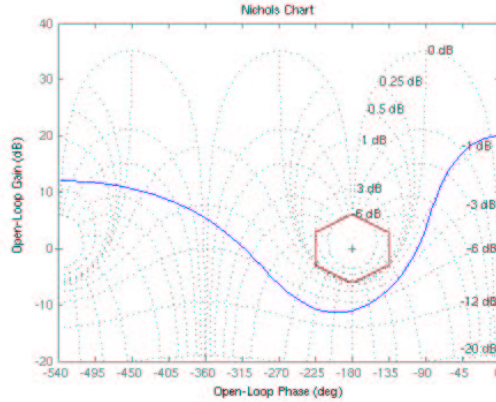


Fig. 1. Nichols plot for $\frac{-4s^4+48s^3-18s^2+250s+600}{s^4+30s^3+282s^2+525s+60}$.

the system for the given frequency. To construct the Nichols plot for a system $g(t)$ with the Laplace transform $G(s)$, the system is exposed sinusoidal signals with a range of frequencies. The output sinusoids are then compared to the input sinusoids. The Nichols plot is a parametric plot (see Section 2.2) of the gain (in decibels) of the output sinusoid against the phase-shift of the output sinusoid. The parameter for this plot is the frequency ω . The formula for the gain of a system exposed to sinusoidal inputs is $20\log_{10}(|G(j\omega)|)$ and the phase-shift is $\text{argument}(G(j\omega))$. The equations for the x and y co-ordinates are rather complex as they contain inverse trigonometric functions and logarithms, respectively. Nichols plots are usually done using computers so this level of complexity is often hidden.

2.2 Parametric equations.

Sometimes instead of expressing the formula for a curve as an equation that gives y in terms of x , i.e. $y = f(x)$, one wishes to express y and x in terms of a third variable, i.e. $y = Y(\omega)$ and $x = X(\omega)$. This is a parametric representation of the curve. The pair of equations $y = Y(\omega)$ and $x = X(\omega)$ are referred to as parametric equations and ω is the parameter.

It is not possible with parametric equations to directly differentiate y with respect to x . In order to obtain $\frac{dy}{dx}$ one must first differentiate the two parametric equations separately and then combine them in the following manner:

$$\frac{dy}{dx} = \frac{dy/d\omega}{dx/d\omega} \quad (1)$$

To find the second derivative of y with respect to x , one must again differentiate two equations and combine them. This time one must differentiate $\frac{dy}{dx}$ with respect to ω and X with respect to ω . One combines these equations in the

following manner:

$$\frac{d^2y}{dx^2} = \frac{d}{d\omega} \left(\frac{dy}{dx} \right) / \frac{dx}{d\omega} \quad (2)$$

Though these equations will give the derivative of y with respect to x the derivative will be in terms of ω .

There are two methods available to find the tangent to a parametric curve at a point $\omega = \omega_0$. The first method is similar to finding the tangent of a non-parametric curve:

$$y = \left. \frac{dy}{dx} \right|_{\omega=\omega_0} (x - X(\omega_0)) + Y(\omega_0) \quad (3)$$

The second method gives the tangent as a pair of parametric equations:

$$x = X(\omega_0) + \left. \frac{dx}{d\omega} \right|_{\omega=\omega_0} (\omega - \omega_0) \text{ and } y = Y(\omega_0) + \left. \frac{dy}{d\omega} \right|_{\omega=\omega_0} (\omega - \omega_0) \quad (4)$$

3 Background

3.1 Mathwork's Simulink

Mathworks' Simulink tool provides a high level environment based on numeric techniques. It is a powerful tool for the design, analysis and simulation of embedded software control systems. In particular it is commonly used for automotive and flight control systems. Simulink provides a graphical user interface in which users can build up a model of a dynamical system from primitive blocks representing mathematical functions, matrix transforms, and so on (see Figure 2). Simulink can use this model to drive a variety of simulations.

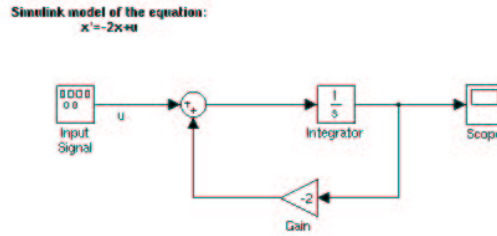


Fig. 2. Simulink model.

Simulink provides a number of block libraries containing inputs, outputs, continuous mathematical elements, and discrete mathematical elements, and since MATLAB and Simulink are so commonly used for the development of control systems a specific “control system toolbox” is available, containing various elements required in control system development.

The block representation is an intuitive way of representing the control systems and allows a hierarchical representation so the design can be layered depending on the analysis level required.

Although MATLAB and Simulink provide useful tools for the development and simulation of systems they do not provide any means of formally verifying results. Since systems are becoming larger and more complex (a modern braking system can be made up of many thousands of blocks), it is more important to ensure that they are correct and more difficult to do so through simulation and numeric analysis. MATLAB does provide a link to the computer algebra system Maple to allow limited symbolic rather than numeric techniques to be employed.

3.2 Symbolic Computation

Computer algebra systems like Maple incorporate a wide variety of symbolic computation techniques, for example for factoring polynomials or computing Grobner bases. They increasingly incorporate some numerical elements also. They can in principle do continuous mathematics “symbolically”, that is to say the query “integrate $\tan(ax)$ between 0 and b ” should return a symbolic answer in a and b .

In practise current computer algebra systems are not always reliable for continuous mathematics, as they do not handle side conditions well and many continuous mathematics problems do not have an accessible symbolic solution [Kal00].

Maple provides some methods for the analysis of formulae, such as the method “iscont”, which checks for continuity of a function in an interval. However, Maple does not use the actual analytic definition of continuity, since this is outside its capabilities, and so may return the wrong answer.

There has been research into improving the reliability of Maple by linking it with the computational logic system PVS to provide additional support [ADG⁺01]. In this system Maple can make calls such as ‘prove positive f [a,b]’ to PVS, which returns either yes, no or fail. In the case of ‘yes’ being returned there is a complete rigorous argument to justify the result.

3.3 Computational Logic

The aim of computational logic is to provide greater assurance in the design and development of software. Formal proofs are produced in a given logical system to prove certain properties of a system hold. This requires both symbolic and logical techniques for manipulating and reasoning about the mathematics of designs and software. This is not a capability of numeric software, and a much wider task than that associated with symbolic computation systems like Maple.

There have been many computational logic tools developed to assist in the production of formal proofs. They range from the highly automated systems, for user with limited experience, to those with a very low level of automation, for expert users.

Proofs developed in systems which lack automation are often long and difficult to produce. For this reason many systems try to include higher levels of automation, such as procedures or collections of strategies, which may be able to automatically find proofs. These systems need to balance high levels of automation with performance and the ability to work at a lower level when the high level devices fail.

Advanced computational logic systems such as HOL [GM93], Isabelle (Paulson, Cambridge) and PVS [ORSvH95] address both foundational issues and practical ones and are supported by a mass of theoretical and practical work. They have been used for a variety of major proof developments in the design of both discrete digital systems and more recently in the continuous domain.

NASA Langley [CM00] use PVS in their project studying aircraft avoidance algorithms in free-flight air-traffic control. This work has relied upon developing precise formulations of the necessary analysis in computational logic systems. NASA Langley have a large public domain library of such material, building on the work of Gottlieb [Got00] at St Andrews university.

PVS. PVS (Prototype Verification System) is a computational logic system which consists of a specification language, some predefined theories and a theorem prover. It has a high level of built-in automation along with a powerful strategy language for developing further automation. The specification language is based on a classical, higher-order logic. It is typed and supports predicate and dependent sub-typing. The core of PVS is implemented in Allegro Common Lisp.

PVS can either be run in batch mode, via a Tcl/Tk interface, or, most commonly, via an emacs interface, which has support for short-cut commands and supports a graphical representation of proof trees.

Much of the what is needed for this project is already built into PVS; the real numbers are built into PVS, much real analysis has already been implemented in it, the transcendental functions have been implemented in it along with various lemmas about their properties. PVS has a number of predefined strategies, such as grind and field. These strategies attempt to apply rewrite rules and lemmas for an automatic proof. Although these strategies may not always be able to complete the proof for a given theory they will often simplify it and split it into various subgoals, making it easier for the user to complete the proof.

4 Approach 1

This was the first approach investigated. The aim was to test whether a curve $y = g(x)$ does not meet a line $y = mx + c$ in a region $[a, b]$. In this approach sets of verification conditions were created for different cases of curves. The conditions were determined using various properties of curves, such as whether it was convex or concave, and increasing or decreasing.

This approach uses knowledge about the properties of convex and concave curves. The gradient of a convex curve decreases as x increases (and conversely

increases with decreasing x). The gradient of a concave curve increases as x increases. These properties can be used to determine the relative position of the curve and the line, and then one or both of the end points can be used to determine whether the curve lies entirely below or above the line. Eventually, these case splits would be hidden from the user through automation of the process. Maple can be used to determine which case of curve a given system has and then PVS can be used to discharge the appropriate verification conditions.

As well as a needing a different theorem for each of the different cases of curves, a different theorem is also needed depending on whether the line denoting the exclusion region has a positive or a negative gradient. Twenty different cases were identified. We developed a suite of verification conditions for the different types of curves. In practise curves are often not entirely convex or entirely concave in the interval of interest, so curves would need to be split into regions according to whether they are concave or convex and monotonic increasing or decreasing.

The next sections show an example of the theorems associated with a monotonic increasing convex curve lying below a line. The theorem is first given in a general form for the line $y = mx + c$ and the curve $y = f(x)$ in the interval $[a, b]$, and then in the specific parametric form. How the theorem for parametric curves translates into a practical method is then shown, followed by a worked example of this method in Maple and PVS.

4.1 An Example Theorem

In this section we consider the the real-valued curve $y = f(x)$ and line $y = g(x) = mx + c$ in the interval $[a, b]$.

Theorem. Suppose we have $f(x)$ and $g(x)$ as above, and the following hold:

1. $f(x)$ is continuous in $[a, b]$
2. $f(x)$ is twice differentiable; that is $\frac{df}{dx}$ and $\frac{d^2f}{dx^2}$ exist
3. $f(x)$ is monotonic increasing in $[a, b]$; that is $\frac{df}{dx} > 0$
4. $f(x)$ is convex in $[a, b]$; that is $\frac{d^2f}{dx^2} < 0$

We have the following three cases:

- Case 1 (see Figure 3): If the following holds then $f(x) < g(x)$ in $[a, b]$
 5. $g(x)$ has a positive gradient; that is $m > 0$
 6. $f(x)$ is steeper than $g(x)$ at the end point b ; that is $\frac{df}{dx}|_{x=b} > m$
 7. $f(x)$ is below $g(x)$ at the end point b ; that is $f(b) < g(b)$
- Case 2 (see Figure 4): If the following hold then $f(x) < g(x)$ in $[a, b]$
 8. $g(x)$ has a positive gradient; that is $m > 0$
 9. $f(x)$ is less steep than $g(x)$ at the end point a ; that is $\frac{df}{dx}|_{x=a} < m$
 10. $f(x)$ is below $g(x)$ at the end point a ; that is $f(a) < g(a)$
- Case 3 (see Figure 5): If the following hold then $f(x) < g(x)$ in $[a, b]$
 11. $g(x)$ does not have a positive gradient; that is $m \leq 0$

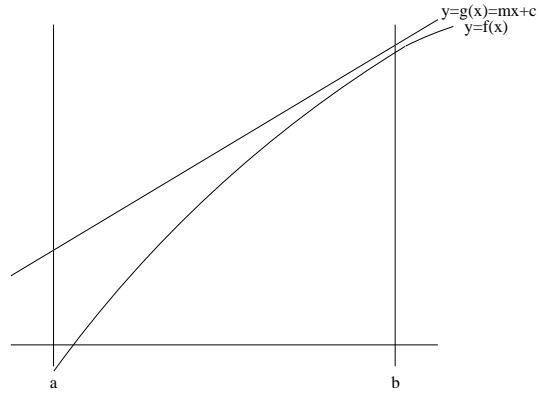


Fig. 3. Monotonic increasing convex curve (case 1).

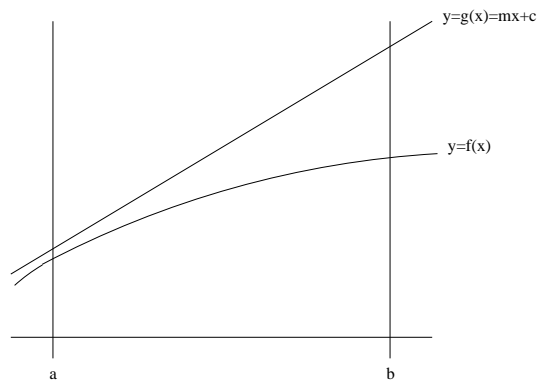


Fig. 4. Monotonic increasing convex curve (case 2).

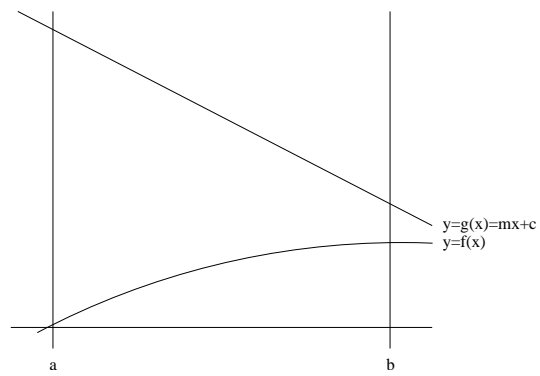


Fig. 5. Monotonic increasing convex curve (case 3).

12. $f(x)$ is below $g(x)$ at the end point b : $f(b) < g(b)$

This theorem is representative of the theorems for several different cases of curves and lines. There are similar theorems for concave curves lying below lines, convex curves lying above lines, and concave curves lying above lines. There is a great deal of symmetry in these theorems which reflect the symmetry of the configuration of curves and lines, for example, an increasing convex curve lying below a line with a positive gradient is a reflection of a decreasing convex curve lying below a line with a negative gradient, and so on (see Figure 6).

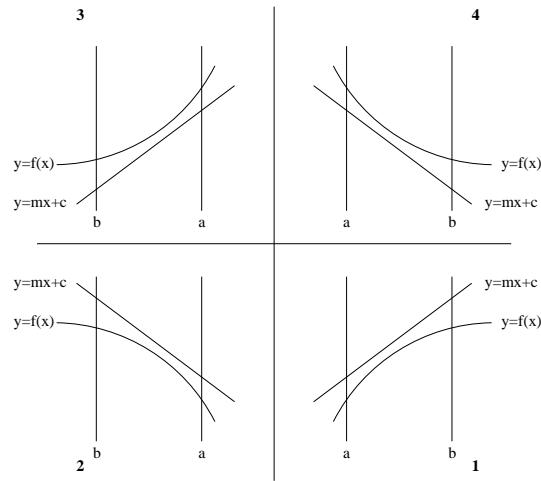


Fig. 6. Symmetry of Curves.

4.2 An Example Theorem for Parametric Curve

This section shows the same theorem as in the previous section only now we consider the real-valued parametric curve $y = Y(\omega)$, $x = X(\omega)$, along with the line $y = g(x) = mx + c$. The interval of interest is $[\omega_a, \omega_b]$, where $a = X(\omega_a)$ and $b = X(\omega_b)$.

Theorem. Suppose we have $Y(\omega)$, $X(\omega)$ and $g(x)$ as above, and the following hold:

1. $Y(\omega)$ and $X(\omega)$ are continuous in $[\omega_a, \omega_b]$
2. $Y(\omega)$ and $X(\omega)$ are twice differentiable; that is $\frac{dy}{d\omega}$, $\frac{d^2y}{d\omega^2}$, $\frac{dx}{d\omega}$ and $\frac{d^2x}{d\omega^2}$ exist
3. curve is monotonic increasing in $[\omega_a, \omega_b]$; that is $\frac{dy}{dx} > 0$
4. curve is convex in $[\omega_a, \omega_b]$; that is $\frac{d^2y}{dx^2} < 0$

We have the following three cases:

- Case 1: If the following hold then $Y(\omega) < g(x)$ in $[\omega_a, \omega_b]$
 5. $g(x)$ has a positive gradient; that is $m > 0$
 6. curve is steeper than $g(x)$ at the end point ω_b ; that is $\frac{dy}{dx}|_{\omega=\omega_b} > m$
 7. $Y(\omega)$ is below $g(x)$ at the end point ω_b ; that is $Y(\omega_b) < g(b)$
- Case 2: If the following hold then $Y(\omega) < g(x)$ in $[\omega_a, \omega_b]$
 8. $g(x)$ has a positive gradient; that is $m > 0$
 9. curve is less steep than $g(x)$ at the end point ω_a ; that is $\frac{dy}{dx}|_{\omega=\omega_a} < m$
 10. $Y(\omega)$ is below $g(x)$ at the end point ω_a ; that is $Y(\omega_a) < g(a)$
- Case 3: If the following hold then $Y(\omega) < g(x)$ in $[\omega_a, \omega_b]$
 11. $g(x)$ does not have a positive gradient; that is $m \leq 0$
 12. $Y(\omega)$ is below $g(x)$ at end point ω_b ; that is $Y(\omega_b) < g(b)$

4.3 Method

The theorem shown in Section 4.2 can be translated into a method using Maple and PVS. In this method a transfer function is given to Maple which does manipulation of the formula, along with symbolic and numeric computation. The results of this computation determine which case the curve falls into and which conditions must be met. PVS is used to discharge the verification conditions.

The method can be considered to be split into different phases: the definition phase; the classification phase; and the proof phase. The definition phase should always be the same for all cases of curve. The classification phase will follow different paths depending on the case of the curve. The proof phase is determined by the classification phase. Considering a curve fitting all the criteria for case 1 in the theorem given in Section 4.2. The method for such a curve would be as follows:

Definition phase.

1. Define equations for the transfer function and the exclusion line, along with the end points a and b (in terms of x). This information should be provided by the user. Maple needs this information.
2. Define equations for gain and phase-shift. Maple is used to compute these equations; PVS is then given them.
3. Evaluate numerically end points a and b in terms of ω . Maple is used to compute the values; PVS is then given them.
4. Define $\frac{dy}{dx}$. Maple is used to compute this; PVS is then given it.
5. Define $\frac{d^2y}{dx^2}$. Maple is used to compute this; PVS is then given it.

Classification phase.

1. Solve $\frac{dy}{dx} > 0$. Maple is used to solve this. This corresponds to determining whether condition 3 of the theorem holds.
2. Solve $\frac{d^2y}{dx^2} < 0$. Maple is used to solve this. This corresponds to determining whether condition 4 of the theorem holds.
3. Solve $\frac{dy}{dx}|_b > m$. Maple is used to solve this. This corresponds to determining whether condition 6 of the theorem holds.
4. Evaluate $Y(\omega_b) < g(b)$. Maple is used to solve this. This corresponds to determining whether condition 7 of the theorem holds.

Proof phase.

1. Prove $Y(\omega)$ and $X(\omega)$ are continuous. PVS is used for this. This corresponds to condition 1 of the theorem.
2. Prove $\frac{dy}{dx}$ and $\frac{d^2y}{dx^2}$ exist. This corresponds to condition 2 of the theorem.
3. Prove $\frac{dy}{dx} > 0$ in $[a, b]$. PVS is used for this. This corresponds to condition 3 of the theorem.
4. Prove $\frac{d^2y}{dx^2} < 0$ in $[a, b]$. PVS is used for this. This corresponds to condition 4 of the theorem.
5. Prove $m > 0$. PVS can be used for this. This corresponds to condition 5 of the theorem.
6. Prove $\frac{dy}{dx}|_b > m$. PVS is used for this. This corresponds to condition 6 of the theorem.
7. Prove $Y(\omega_b) < g(b)$. PVS is used for this. This corresponds to condition 7 of the theorem.

4.4 Example

In this section we will show a complete worked example of the approach described above. This example is for a very simple control system with the transfer function $G(s) = \frac{1}{s^3+6s^2+5s+2}$, the line of the exclusion region $y = \frac{12x}{\pi} + 6$, and the interval $[-\pi, \frac{3\pi}{4}]$ (see Figure 7). We show a Maple worksheet demonstrating the functions Maple requires for the pre-processing of the transfer function. Then we show the PVS input file along with a discussion of how the verification conditions were discharged.

Maple – Definition phase. In this section input into Maple is preceded by the symbol $>$ and output is shown on the following lines. Input can be made up of several statements, one after the other, and by nested calls. If an input statement is followed by a semi-colon Maple displays the result; if it is followed by a colon the result is not displayed. Maple represents the complex constant as I , and since Maple can not represent ω , w has been used as a substitute.

The first task in the Maple file is to set up various variables that are needed, such as the end points of the interval of interest, the equation for the line of

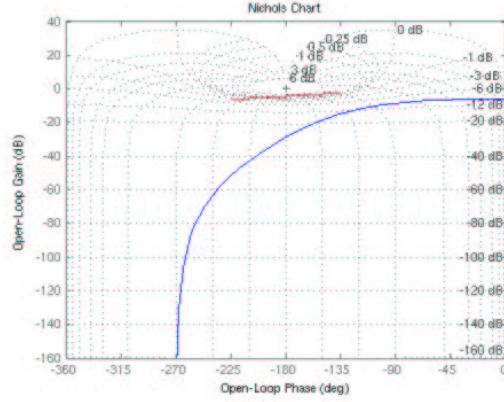


Fig. 7. Nichols plot for $\frac{1}{s^3+6s^2+5s+2}$.

the exclusion region, and the Fourier transform of the transfer function of the system:

```
> assume(w, real):  ax := -Pi:  bx := -3 * Pi/4:  m := 12/Pi:
> c := 6:  l := m * x + c;  tf := subs(s = I * w, 1/(s^3 + 6 * s^2 + 5 * s + 2));
```

$$l := \frac{12x}{\pi} + 6$$

$$tf := \frac{1}{-I w^{-3} - 6 w^{-2} + 5 I w^{-1} + 2}$$

This is a very simple task for Maple and involves only assignment and substitution. Maple allows the user to impose certain properties on a variable using the “assume” command. Here we assume w is a real number. A variable that has assumptions made about it is followed by a tilde in Maple output.

The next task is to find the equations for the gain and phase-shift of the transfer function:

```
> X := evalc(argument(tf));  Y := 20 * log10(abs(tf));
```

$$X := \arctan(w^{-3} - 5 w^{-1}, -6 w^{-2} + 2)$$

$$Y := \frac{20 \ln\left(\frac{1}{\sqrt{26 w^{-4} + w^{-2} + 4 + w^{-6}}}\right)}{\ln(10)}$$

This is slightly more complex than the first step as it involves manipulation of complex numbers and transcendental functions. Maple has a built in \arctan function which takes two parameters in order to keep the result within $[-\pi, \pi]$ instead of $[\frac{-\pi}{2}, \frac{\pi}{2}]$.

Once the equation for the phase-shift has been found it can be used to evaluate numerically the end points a and b in terms of ω (or w) instead of x :

> $bw := \text{fsolve}(X = bx, w);$ $aw := \text{fsolve}(X = ax, w);$

$bw := 1.000000000$

$aw := 2.236067977$

Next, Maple is used to find the derivative and double derivative of the curve given by the parametric equations:

> $dydx := \text{simplify}(\text{diff}(Y, w)/\text{diff}(X, w));$

$$dydx := \frac{10 w^{\sim} (52 w^{\sim 2} + 1 + 3 w^{\sim 4})}{(\ln(2) + \ln(5)) (3 w^{\sim 4} + 12 w^{\sim 2} + 5)}$$

> $d2ydx2 := \text{simplify}(\text{diff}(dydx, w)/\text{diff}(X, w));$

$$d2ydx2 := -\frac{5(-48 w^{\sim 6} + 690 w^{\sim 4} + 768 w^{\sim 2} + 5 + 9 w^{\sim 8})(26 w^{\sim 4} + w^{\sim 2} + 4 + w^{\sim 6})}{(\ln(2) + \ln(5)) (3 w^{\sim 4} + 12 w^{\sim 2} + 5)^3}$$

This is again a fairly straightforward task for Maple, using well established methods.

So far in this example, all of the work done in Maple has been symbolic computation, numeric evaluation, and manipulation of the formula. This was essentially setting up everything needed for the classification of the curve to begin. The next steps are used to determine whether the curve is increasing or decreasing, and concave or convex.

Maple – Classification phase. Maple provides a function to solve inequalities, and return a range in which the inequalities are true. This means that not only will Maple determine whether the curve is concave or convex, but it can also determine the intervals in which these properties hold:

> $\text{solve}(dydx > 0, w);$

$\text{RealRange}(\text{Open}(0), \infty)$

> $\text{solve}(d2ydx2 < 0, w);$

w^{\sim}

These results show that, according to Maple, the curve is monotonic increasing and convex over all non-negative values of w . Maple is not guaranteed to return a result for this method.

This next step was determined by the result of the last two and corresponds to condition 6 in the theorem given in Section 4.2

> $\text{is}(\text{subs}(w = bw, dydx) > m);$

true

In this case the result of this is *true* so we continue onto condition 7 in the theorem, if it had been *false* we would have gone onto condition 9.

$> is(subs(w = bw, Y) < subs(x = bx, l));$

true

The result of this calculation is also *true*. These last four calculations were in an area where Maple is not very strong, for this reason the next step is to use PVS to formally verify the results.

PVS – Definition phase. Firstly, this theory must import library files, which contain various lemmas and strategies to do with the transcendental functions, in particular the natural logarithm, and continuity of functions. Then, several variables and formulae must be defined.

In PVS, as with Maple, we also represents ω as w . The variables a and bw represent the end points of the exclusion region in terms of ω , and bx represents the end point b in terms of x . The formulae $Y(w)$, $X(w)$, $d(w)$ and $d2(w)$ represent the equations for the y coordinate of the parametric curve, the x coordinate for the parametric curve, the first derivative of the curve, and the second derivative of the curve respectively. They are all taken from the Maple. In this theory π and the natural logarithms of 2 and 5 are given a numeric value.

```
mapleCubic: THEORY
BEGIN
```

```
IMPORTING transcendentals@top_analysis
```

```
w : VAR nreal
```

```
a : real = 23/10
```

```
bw : real = 1
```

```
bx : real = -3*(31416/10000)/4
```

```
ln2 : real = 69315/100000
```

```
ln5 : real = 16094/10000
```

```
X(w) : real = atn((w^3-5*w)/(2-6*w^2))
```

```
Y(w) : real = 20*(ln(1/((w^6+26*w^4+w^2+4)^(1/2))))/ln(10)
```

```
d(w) : real = 10*w*(3*w^4+52*w^2+1)/((ln2+ln5)*(3*w^4+12*w^2+5))
```

```
d2(w) : real = -5*(9*w^8-48*w^6+690*w^4+768*w^2+5)*(w^6+26*w^4+w^2+4)/
((ln2+ln5)*(3*w^4+12*w^2+5)^3)
```

PVS – Proof phase. The PVS input file contains a theory which is divided into a number of lemmas. These lemmas roughly correspond to the theorem given as case 1 in section 4.2. Lemmas A and A2 correspond to condition 1 in the theorem described in Section 4.2. Lemmas B and B2 correspond to condition 3 and lemmas C and C2 correspond to condition 4. Conditions 3 and 4 are split

into two lemmas since they are true for all values of w , as well as being true for the specific interval of interest. Lemma D corresponds to condition 6 and lemma E to condition 7

```

lemmaA : LEMMA
  FORALL (y:posreal) : continuous(lambda (w:posreal) : Y(w), y)

lemmaA2 : LEMMA
  FORALL (x:posreal) : continuous(lambda (w:posreal) : X(w), x)

lemmaB : LEMMA
  FORALL w : 0 < w IMPLIES d(w) > 0

lemmaB2 : LEMMA
  FORALL w : 1 < w AND w < 23/10 IMPLIES d(w) > 0

lemmaC : LEMMA
  FORALL w : 0 < w IMPLIES d2(w) < 0

lemmaC2 : LEMMA
  FORALL w : 1 < w AND w < 23/10 IMPLIES d2(w) < 0

lemmaD : LEMMA
  d(bw) - 12/(31416/10000) > 0

lemmaE : LEMMA
  12/(31416/10000)*bx+6 - Y(bw) > 0

END mapleCubic

```

Proof. PVS has a number of high level strategies built in, so the proofs of these lemmas try to make use of them where ever possible. Lemmas B and D could be proved automatically, using the in-built strategies, grind and field. Lemma C needed some lower level interaction as well as the high level strategies for the proof. Case splits needed to be introduced along with the application and instantiation of a lemma to do with multiplying both sides of an inequality by a positive number. Lemmas B2 and C2 are proved using lemmas B and C respectively. Lemmas A, A2 and E are unproved.

4.5 Benefits and Issues

This approach tries to make use of the existing support in Maple and PVS. The Maple features that are used are all fairly basic functions, such as differentiation, and the proofs in PVS try to exploit the existing strategies. Since inverse trigonometric functions and logarithms disappear when differentiated, and this

method deals primarily with the derivative and double derivative of the curve, PVS can prove several of the lemmas for each case without the need for further support to be built in. However, due to the limited support for the inverse trigonometric functions and logarithms in PVS, lemmas A, A2 and E all require further support to be built in before they can be proved.

It was found that as the order of the polynomials in the transfer function increased so did the complexity of the proofs for lemmas B and C. More lower level user intervention was required, however, in each case the intervention seemed to be of a similar kind, involving forced case splits and the application of certain lemmas. This seems to indicate that it may not be too difficult to automate the proofs for transfer functions of a low order.

The suite of cases for which verification conditions were produced is incomplete. Using this method there is no set of verification conditions for the case where neither conditions 1 to 4 and either 5 and 6 or 8 and 9 hold (see Figure 8).

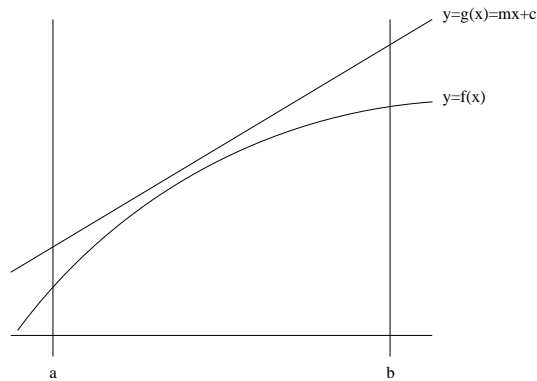


Fig. 8. Monotonic increasing curve.

If the curve has a point of inflection in the interval of interest then the *exact* point of inflection is needed for the proof to be complete. This may not be possible to find as there is no guarantee that the point will be a rational number.

5 Approach 2

This approach differs from the first in that instead of proving that a curve lies below a given line one *finds* a line the curve lies below. This approach uses the fact that it is easier to prove that a line lies below another than it is to prove a curve lies below a line. One finds a line that the curve lies below then proves this line lies below the line of the exclusion region. Based on the shape of the curve one can determine a line that the curve will definitely lie below, for instance, a convex curve lies entirely below its tangent at any point, and a concave curve

lies below the line which intersects the curve at the end points of an interval. These fact can be combined to produce methods for finding a line that a curve with one point of inflection lies below.

Suppose a curve is convex in the region $[a, v]$ and concave in the region $[v, b]$, then we know that the convex region of the curve lies below any tangent $nx + d$, from which it follows that the curve at v must also lie below $nx + d$. Since we know that v lies below $nx + d$ then all that remains for the concave part of the curve to lie below the line is for the curve at b to lie below $nx + d$. A similar line exists for curves which are concave then convex.

These methods can be adapted for proving that a curve lies above.

The next sections show an example of the theorems associated with a monotonic increasing convex curve lying below a line. The theorem is first given in a general form for the line $y = mx + c$ and the curve $y = f(x)$ in the interval $[a, b]$, and then in the specific parametric form.

5.1 An Example Theorem

In this section we consider the real-valued curve $y = f(x)$ and line $y = g(x) = mx + c$ in the interval $[a, b]$.

Theorem. Suppose we have $f(x)$ and $g(x)$ as above and the following hold:

1. $f(x)$ is continuous in $[a, b]$
2. $f(x)$ is twice differentiable; that is $\frac{df}{dx}$ and $\frac{d^2f}{dx^2}$ exist

We have the following four cases:

- Case 1 (see Figure 9): If the following hold then $f(x) < g(x)$ in $[a, b]$
 3. $f(x)$ is convex in $[a, b]$; that is $\frac{d^2f}{dx^2} < 0$
 4. $nx + d$ is tangent to the $f(x)$ in the interval $[a, b]$
 5. tangent lies below line of exclusion region in interval $[a, b]$; that is $nx + d < mx + c$
- Case 2 (see Figure 10): If the following hold then $f(x) < g(x)$ in $[a, b]$
 6. $f(x)$ is concave in $[a, b]$, that is $\frac{d^2f}{dx^2} \geq 0$
 7. $nx + d$ is line passing through points $(a, f(a))$ and $(b, f(b))$; that is $y = \frac{f(b)-f(a)}{b-a}x - \frac{f(b)-f(a)}{b-a}b + f(b)$ or $y = \frac{f(b)-f(a)}{b-a}x - \frac{f(b)-f(a)}{b-a}a + f(a)$
 8. line lies below line of exclusion region in interval $[a, b]$; that is $nx + d < mx + c$
- Case 3 (see Figure 11): If the following hold then $f(x) < g(x)$ in $[a, b]$
 9. $f(x)$ is concave in interval $[a, v]$ and convex in $(v, b]$
 10. $nx + d$ is tangent to the convex part of the curve which also passes through point $(a, f(a))$
 11. line of exclusion region lies above the tangent in interval $[a, b]$; that is $nx + d < mx + c$
- Case 4 (see Figure 12): If the following hold then $f(x) < g(x)$ in $[a, b]$
 12. $f(x)$ is convex in interval $[a, v)$ and concave in $[v, b]$

13. $nx + d$ is tangent to the convex part of the curve which also passes through point $(b, f(b))$
14. line of exclusion region lies above the tangent in interval $[a, b]$; that is $nx + d < mx + c$

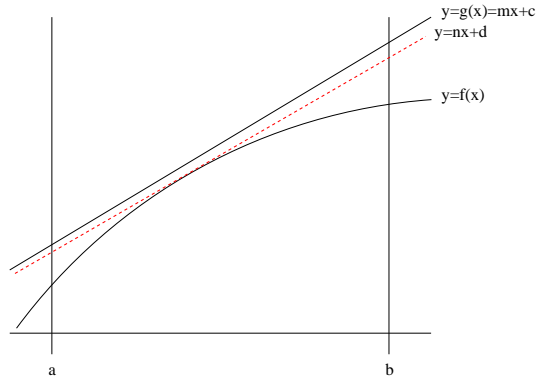


Fig. 9. Monotonic increasing convex curve.

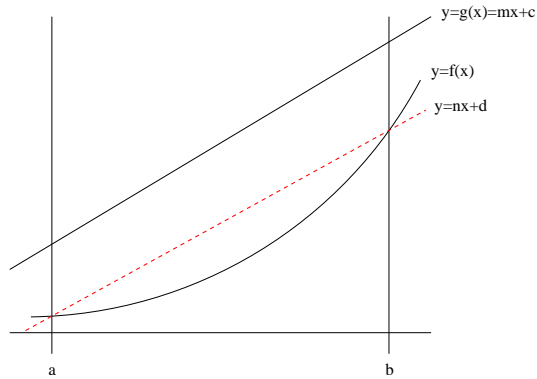


Fig. 10. Monotonic increasing concave curve.

In the case that a curve is convex in the interval (case 1) any tangent could be used as the line the curve lies below, since the curve would lie entirely below any of them. However, all of the tangents may not lie below the line of the exclusion region. A choice must be made as to which tangent to use – a tangent could be found at a particular point, for example at either of the end points; or a tangent could be found at a particular gradient of the curve, for example at the point where the curve has a gradient equal to the exclusion line, that is $\frac{df}{dx} = m$.

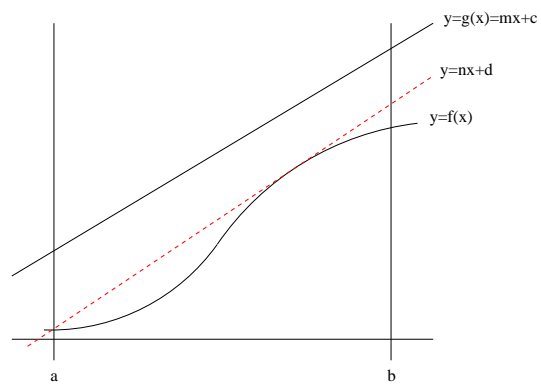


Fig. 11. Monotonic increasing concave then convex curve.

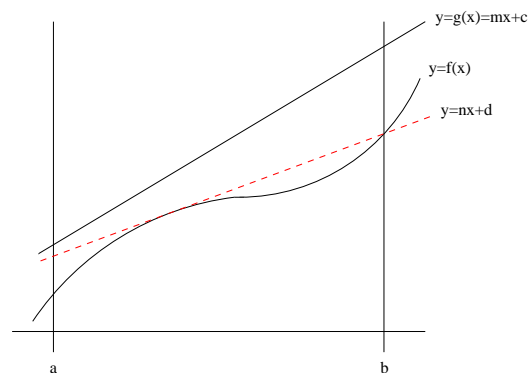


Fig. 12. Monotonic increasing convex then concave curve.

5.2 An Example Theorem for Parametric Curve

In this section we consider the real-valued parametric curve $y = Y(\omega)$, $x = X(\omega)$ and the line $y = g(x) = mx + c$ in the interval $[\omega_a, \omega_b]$, where $a = X(\omega_a)$ and $b = X(\omega_b)$.

Theorem. Suppose we have $Y(\omega)$, $X(\omega)$ and $g(x)$ as above, and the following hold:

1. $Y(\omega)$ and $X(\omega)$ are continuous in $[\omega_a, \omega_b]$
2. $Y(\omega)$ and $X(\omega)$ are twice differentiable; that is $\frac{dy}{d\omega}$, $\frac{d^2y}{d\omega^2}$, $\frac{dx}{d\omega}$ and $\frac{d^2x}{d\omega^2}$ exist

We have the following four cases:

- Case 1: If the following hold then $Y(\omega) < g(x)$ in $[\omega_a, \omega_b]$
 3. curve is convex in interval $[\omega_a, \omega_b]$; that is $\frac{d^2y}{dx^2} < 0$
 4. $nx + d$ is tangent to curve
 5. tangent lies below line of exclusion region in interval $[\omega_a, \omega_b]$; that is $nx + d < mx + c$
- Case 2: If the following hold then $Y(\omega) < g(x)$ in $[\omega_a, \omega_b]$
 6. curve is concave in interval $[\omega_a, \omega_b]$; that is $\frac{d^2y}{dx^2} \geq 0$
 7. $nx + d$ is line passing through the points $(X(\omega_a), Y(\omega_a))$, $(X(\omega_b), Y(\omega_b))$
 8. line lies below line of exclusion region in interval $[\omega_a, \omega_b]$; that is $nx + d < mx + c$
- Case 3: If the following hold then $Y(\omega) < g(x)$ in $[\omega_a, \omega_b]$
 9. curve is concave in interval $[\omega_a, \omega_v]$ and convex in $[\omega_v, \omega_b]$
 10. $nx + d$ is tangent to the convex part of the curve which also passes through point $(X(\omega_a), Y(\omega_a))$
 11. line lies below line of the exclusion region; that is $nx + d < mx + c$
- Case 4: If the following hold then $Y(\omega) < g(x)$ in $[\omega_a, \omega_b]$
 12. curve is convex in interval $[\omega_a, \omega_v]$ and concave in $[\omega_v, \omega_b]$
 13. $nx + d$ is tangent to the convex part of the curve which also passes through point $(X(\omega_b), Y(\omega_b))$
 14. line lies below line of the exclusion region; that is $nx + d < mx + c$

This parametric version of the theorem is given assuming the tangent to the convex parts of curves is found using the non-parametric method, however, the theorem would still be perfectly valid if the tangents were found in parametric form.

5.3 Benefits and Issues

This method provides coverage for all the cases identified in the first approach and would also work for both increasing and decreasing curves below any line with a real gradient. Cases with more than one point of inflection could either be split, or specific cases could be identified for convex, concave then convex curves and concave, convex then concave curves.

The approach using a gradient to find the tangent to the convex curve is the most effective when it comes to finding a tangent which also lies below the exclusion line. However, this method requires that $\frac{dy}{dx} = m$ not only has a solution in $[\omega_a, \omega_b]$ but also that it has an *exact* solution for ω in $[\omega_a, \omega_b]$. This problem is similar to the problem of finding the exact point of inflection.

Depending on which method for finding the tangent is chosen, and whether the curve has a point of inflection within the interval of interest, the complexity of the calculation in Maple is increased.

This method does not decrease the complexity of the proof in PVS as it is still necessary to prove that the curve is convex or concave and there is still the need to extend the support in PVS for the proof of the continuity of $Y(\omega)$ and $X(\omega)$

6 Approach 3

This method attempts to deal directly with the equations of the curve and line, rather than examining the properties of the curve.

If you subtract one equation $f(x)$ from another $g(x)$ then you can prove that $f(x)$ lies below $g(x)$ if the result of the subtraction is positive for all x . This could be simplified to proving that the result is positive at a single point and that the result is at no point zero (or in other words that $\frac{1}{g(x)-f(x)}$ is continuous).

This method needs no case splitting of the curve, so the complete set of theorems is shown in the next section with the parametric version of the theorems following.

6.1 Example Theorem

In this section we consider the real-valued curve $y = f(x)$ and the line $y = g(x) = mx + c$ in the interval $[a, b]$.

Theorem. Suppose we have $f(x)$ and $g(x)$ as above, then we have the following two cases:

- Case 1: If the following hold then $f(x) < g(x)$ in $[a, b]$
 1. $\frac{1}{g(x)-f(x)}$ is continuous in the interval $[a, b]$
 2. $\frac{1}{g(a)-f(a)} > 0$ or $\frac{1}{g(b)-f(b)} > 0$
- Case 2: If the following hold then $f(x) > g(x)$ in $[a, b]$
 3. $\frac{1}{g(x)-f(x)}$ is continuous in the interval $[a, b]$
 4. $\frac{1}{g(a)-f(a)} < 0$ or $\frac{1}{g(b)-f(b)} < 0$

6.2 Example Theorem for Parametric Curve

In this section we consider the real-valued parametric curve $y = Y(\omega)$, $X(\omega)$ and the line $y = g(x) = mx + c$ in the interval $[\omega_a, \omega_b]$, where $a = X(\omega_a)$ and $b = X(\omega_b)$.

Theorem. Suppose we have $Y(\omega)$, $X(\omega)$ and $g(x)$ as above, then we have the following two cases:

- Case 1: If the following hold then $Y(\omega) < g(x)$ in $[\omega_a, \omega_b]$
 1. $\frac{1}{mX(\omega)+c-Y(\omega)}$ is continuous in the interval $[\omega_a, \omega_b]$
 2. $\frac{1}{mX(\omega_a)+c-Y(\omega_a)} > 0$ or $\frac{1}{mX(\omega_b)+c-Y(\omega_b)} > 0$
- Case 2: If the following hold then $Y(\omega) > g(x)$ in $[\omega_a, \omega_b]$
 3. $\frac{1}{mX(\omega)+c-Y(\omega)}$ is continuous in the interval $[\omega_a, \omega_b]$
 4. $\frac{1}{mX(\omega_a)+c-Y(\omega_a)} < 0$ or $\frac{1}{mX(\omega_b)+c-Y(\omega_b)} < 0$

In the parametric version of the theorem the two equations that we need to subtract are in terms of different variables. In order to be able to work with these equations one needs to be converted into the same variable as the other. This can either be done by converting the parametric curve into an equation which gives y in terms of x , or by converting the line to be y in terms of ω . In the theorem shown above the latter was done by substituting $X(\omega)$ for x .

6.3 Issues

This approach seems very simple on the surface – the work required in Maple for this approach is very simple – it only requires the simple setting up of variables and equations and then the subtraction of one equation from the other; and the theory in PVS consists of only two lemma, however, with the current support in PVS for the inverse trigonometric functions and logarithms, neither lemma could be proved. Also, due to principle values of the inverse trigonometric functions used in Maple the exclusion line is limited to ranging over $(-\pi, \pi]$

7 Conclusions

In each of the approaches that were investigated more support for the inverse trigonometric functions and logarithms would be needed in PVS.

The first approach needed many different cases depending on properties of the curve, and these cases were not complete as they did not cover every possibility for the curve. The lemmas dealing with the derivative and double derivative of the curve were fairly easy to prove for simple cases, and could possibly be automated. However, as the transfer function increased in complexity so did the proofs.

The second approach provides coverage for all cases of curve as well as requiring fewer different cases to do this than the first approach. In the second approach some similar work to that of approach 1 needed to be done in Maple and PVS but also more needed to be done. As well as determining which case the curve fell into, Maple also needed to find the line which the curve lay below. In PVS for the proof to be complete for convex cases one would need to prove that the line you were dealing with was indeed a tangent to the curve. Approach 2 suffers from the same problem as approach 1 when it comes to increased complexity of transfer function, however, there is the possibility that

one could decompose the transfer function into simpler components, find lines that each of these components lie below, do the proofs for these simpler cases, and then compose the two lines to show that if the separate components of the transfer function lie below these two lines then the composed transfer function must lie below the composition of the lines.

The third approach seems less complex than the first two as all cases of curve are covered in far fewer cases in the theorem. There is less work to be done in Maple and fewer lemmas in PVS theories. However, the support needed in PVS is not there for any of the lemmas. Since the support in PVS does not exist for the lemmas in this approach it is very difficult to make judgements about whether this approach will ultimately be easier to automate than the others, or about whether the proofs would become much more complex with more complex examples. This approach does not have the same potential for composability that approach 2 does.

8 Future Work

The first and perhaps most important item of further work for this project is to extend the support in PVS for the inverse trigonometric functions, in particular arctan, for the natural logarithm, and possibly for logarithm to the base ten.

Once the support is built into PVS the proof process could be automated with the development of purpose built strategies for the proof of the various lemmas. The strategies developed will depend on which approach to the problem is ultimately chosen.

The links between Maple and PVS can be exploited to automate the process further.

References

- [Act97] Action Group FM(AG08). Robust flight control design challenge problem formulation and manual: the high incidence research model (HIRM). Technical Report TP-088-4, version 3, Group for Aeronautical Research and Technology in Europe (GARTEUR), garteursecretary@inta.es, April 1997.
- [Ada95] R. A. Adams. *Calculus: a complete course*. Addison-Wesley, third edition, 1995.
- [ADG⁺01] A. Adams, M. Dunstan, H. Gottliebsen, T. Kelsey, U. Martin, and S. Owre. Computer algebra meets automated theorem proving: Integrating Maple and PVS. In R. J. Boulton and P. B. Jackson, editors, *Proceedings of the 14th International Conference on Theorem Proving in Higher Order Logics (TPHOLs 2001)*, volume 2152 of *Lecture Notes in Computer Science*, pages 27–42. Springer-Verlag, 2001.
- [AH00] Mark Aagaard and John Harrison, editors. *Proceedings of the 13th International Conference on Theorem Proving in Higher Order Logics (TPHOLs 2000)*, volume 1869 of *Lecture Notes in Computer Science*. Springer, 2000.

- [CM00] Victor Carreno and Cesar Munzo. Aircraft trajectory modelling and alerting algorithm verification. In Aagaard and Harrison [AH00].
- [DB01] R. C. Dorf and R. H. Bishop. *Modern Control Systems*. Prentice-Hall, ninth edition, 2001.
- [FPW98] G. F. Franklin, J. D. Powell, and M. Workman. *Digital Control of Dynamic Systems*. Addison Wesley Longman, third edition, 1998.
- [GM93] M. J. C. Gordon and T. F. Melham. *Introduction to HOL: A theorem proving environment for higher order logic*. Cambridge University Press, 1993.
- [Got00] H. Gottlieb. Transcendental functions and continuity checking in pvs. In Aagaard and Harrison [AH00], pages 387–439.
- [Har00] John Harrison. Formal verification of ia-64 division algorithms. In Aagaard and Harrison [AH00], pages 387–439.
- [Kal00] Erich Kaltofen. Challenges of symbolic computation: My favorite open problems. *Journal of Symbolic Computation*, 29(6):891–919, jun 2000.
- [Mar00] Ursula Martin. Towards formal methods for mathematical modeling. *Proceedings 5th NASA Langley Workshop on Formal Methods*, 2000.
- [Oga95] K. Ogata. *Discrete-Time Control Systems*. Prentice-Hall, second edition, 1995.
- [Oga97] K. Ogata. *Modern Control Engineering*. Prentice-Hall, third edition, 1997.
- [ORSvH95] Sam Owre, John Rushby, Natarajan Shankar, and Friedrich von Henke. Formal verification for fault-tolerant architectures: Prolegomena to the design of PVS. *IEEE Transactions on Software Engineering*, 21(2):107–125, feb 1995.


```

      ("2" (ASSERT) NIL NIL))
    NIL))
  NIL))
  NIL))
  NIL))
  NIL))
  NIL)
(|lemmaC2| "" (SKOSIMP*)
  (" (USE "lemmaC") ((" (ASSERT) NIL NIL)) NIL)) NIL)
(|lemmaD_TCC1| "" (SUBTYPE-TCC) NIL NIL)
(|lemmaD| "" (GRIND) NIL NIL)
(|lemmaE| "" (EXPAND "Y") ((" (POSTPONE) NIL NIL)) NIL))

```

A.2

PVS proof summary for example from Section 4.4

Proof summary for theory mapleCubic

X_TCC1.....	proved - complete	[0](0.05 s)
X_TCC2.....	proved - complete	[0](0.03 s)
X_TCC3.....	unfinished	[0](0.13 s)
Y_TCC1.....	proved - complete	[0](0.04 s)
Y_TCC2.....	proved - complete	[0](0.04 s)
Y_TCC3.....	unfinished	[0](0.11 s)
Y_TCC4.....	proved - complete	[0](0.16 s)
Y_TCC5.....	unfinished	[0](0.17 s)
d_TCC1.....	proved - complete	[0](0.22 s)
d2_TCC1.....	proved - complete	[0](0.05 s)
d2_TCC2.....	proved - complete	[0](0.04 s)
d2_TCC3.....	proved - complete	[0](0.17 s)
lemmaA.....	unfinished	[0](3.37 s)
lemmaA2.....	unfinished	[0](2.21 s)
lemmaB.....	proved - complete	[0](17.19 s)
lemmaB2.....	proved - complete	[0](0.11 s)
lemmaC.....	proved - complete	[0](76.53 s)
lemmaC2.....	proved - complete	[0](0.11 s)
lemmaD_TCC1.....	proved - complete	[0](0.03 s)
lemmaD.....	proved - complete	[0](0.35 s)
lemmaE.....	unfinished	[0](0.03 s)
Theory totals: 21 formulas, 21 attempted, 15 succeeded		(101.14 s)

Grand Totals: 21 proofs, 21 attempted, 15 succeeded (101.14 s)