LANCASTER UNIVERISTY A picture containing table

Description automatically generated

Faculty of Science & Technology

Engineering Department

|  |
| --- |
|  |
| THIRD YEAR MEng PROJECT |
| “Adaptive Medical Treatment Focused on the Control Application of Warfarin”  Supervisor: Dr James Taylor |
|  |

|  |
| --- |
| By George Caddick  15-05-2020 |

Lancaster University

Faculty of Science and Technology

Engineering Department

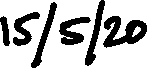
**Signed Declaration on the Submission of a Project**

I declare that this project is my own work and has not been submitted in substantially the same form towards the award of a degree or other qualificatory work and affirm that acknowledgement has been made to assistance given and that all major sources have been appropriately referenced.

Signed: ……………………………………………………



Date: ……………………………………………………



# Summary

The issue of the dosage of warfarin is a sensitive problem, with constant attention being needed to monitor the dosage and effects of this drug. Some patients being lifetime users must carefully check their INR (International Normalised Ratio), supervised by medical professionals who change the dosage accordingly. The INR should be in the range of 2 to 3. If a patient’s INR leaves this range, there can be difficulties such as blood clots or bleeding complications. Introducing a control system for the dosage of warfarin would reduce the risk of this happening.

The project’s aims were to design, model, control and simulate the systems to control the problem of warfarin dosage and be compared using previous data and error calculations. This was done using MATLAB and specialised functions included in the Captain Toolbox.

This project used MATLAB to create models using previous patient data and functions such as RIVID or trial and error. The models were simulated and compared against other models, taken from papers, using the previous patient data. Rt2 was calculated using the model response and data output along with error calculations for each model and these were used in the comparison of the models.

The function PIP was used to create the ideal controller variables using specifically chosen poles and the model, ensuring the system was stable, and the full system was then modelled in SIMULINK. For some models the PIP function did not work, as the model was not the right form, and therefore a trial and error method were used for these controller variables.

The results of the modelling show two distinct models that represent the previous patient data most accurately: the Gain model and DLR Eq1 model. The results from these can be seen in the table below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Percentage over 0.2** | **Percentage over 0.5** | **Positives** | **Negatives** |
| DLR Eq 1 | 74.6% | 37.0% | Simple gain model | Gain depends on patient and previous data |
| Gain | 55.8% | 17.5% | Simplest model and generalised for all patients | Not as effective as DLR Eq 1/3 |

The results for the control systems were similar to the results from the models: the Gain model system and DLR Eq1 systems were the best for speed of response, control of disturbances and robustness. Of these, the Gain model system was considered the best as it was a generalised system, that was effective on a large percentage of patients.

Table of Contents

[Adaptive Medical Treatment Focused on the Control Application of Warfarin 0](#_Toc39759774)

[Summary 1](#_Toc39759775)

[List of Figures: 5](#_Toc39759776)

[List of Tables: 7](#_Toc39759777)

[1.0.0 - Introduction 9](#_Toc39759778)

[1.1.0 - Project Summary 9](#_Toc39759779)

[1.2.0 - Literature Review 10](#_Toc39759780)

[1.2.1 - Introduction of Warfarin 10](#_Toc39759781)

[1.2.2 - Control Theory already used in other medication 10](#_Toc39759782)

[1.2.3 - Why control is needed in this problem and issues with dosing of warfarin 10](#_Toc39759783)

[1.2.4 - Problems with diet and how it affects the effectiveness of warfarin including how warfarin works 11](#_Toc39759784)

[1.2.5 - Previous work with control theory and warfarin 11](#_Toc39759785)

[1.2.6 - Speculation into how this project could be implemented in the medical field 12](#_Toc39759786)

[1.2.7 – Transfer Functions 12](#_Toc39759787)

[1.2.8 – PIP Control 12](#_Toc39759788)

[1.3.0 - Section Overview 13](#_Toc39759789)

[2.0.0 - Patients Data 14](#_Toc39759790)

[3.0.0 - Model Research and Investigation 15](#_Toc39759791)

[3.1.0 – Model Introduction 15](#_Toc39759792)

[3.1.2 – Method 15](#_Toc39759793)

[3.2.0 - First Order Model 15](#_Toc39759794)

[3.2.1 – Introduction 15](#_Toc39759795)

[3.2.2 – Comparing Model Output and Data Output for First Order Model 16](#_Toc39759796)

[3.3.0 - Second Order Model 19](#_Toc39759797)

[3.3.1 – Introduction 19](#_Toc39759798)

[3.3.2 – Comparing Model Output and Data Output for Second Order Model 19](#_Toc39759799)

[3.4.0 - Patient Specific Models 22](#_Toc39759800)

[3.4.1 - Brief Description of How RIVID Works 22](#_Toc39759801)

[3.4.2 - Individual Transfer Functions for a Couple Patients 22](#_Toc39759802)

[3.5.0 - Generalised Patient Models using RIVID 27](#_Toc39759803)

[3.5.1 – Introduction 27](#_Toc39759804)

[3.5.2 – Comparing Model Output and Data Output for PG73 Model 27](#_Toc39759805)

[3.5.3 – Comparing Model Output and Data Output for PG216 Model 29](#_Toc39759806)

[3.6.0 - Dynamic Linear Regression Models 32](#_Toc39759807)

[3.6.1 – Introduction 32](#_Toc39759808)

[3.6.2 – Equation 3 with varying NVR 32](#_Toc39759809)

[3.6.3 – Equation 3 Testing c1 and c2 variables 33](#_Toc39759810)

[3.6.4 – Comparing Equation 1 and 3 34](#_Toc39759811)

[3.6.5 – Comparing Model Output and Data Output using Eq1 35](#_Toc39759812)

[3.6.6 – Eq2 Testing 36](#_Toc39759813)

[3.7.0 - Gain Model Research 38](#_Toc39759814)

[3.7.1 – Introduction 38](#_Toc39759815)

[3.7.2 – Gain Research 38](#_Toc39759816)

[3.7.3 - Further Research into Gain Value 38](#_Toc39759817)

[3.7.4 – Comparing Model Output and Data Output 39](#_Toc39759818)

[3.8.0 - Model Conclusions 42](#_Toc39759819)

[3.8.1 – First Order Model Conclusions 42](#_Toc39759820)

[3.8.2 – Second Order Model Conclusions 42](#_Toc39759821)

[3.8.3 – Patient Specific Models Conclusions 42](#_Toc39759822)

[3.8.4 – Patient Generalised Models Conclusions 42](#_Toc39759823)

[3.8.5 – DLR Models Conclusions 42](#_Toc39759824)

[3.8.6 – Gain Model Conclusions 43](#_Toc39759825)

[3.8.7 – Overall Model Conclusions 43](#_Toc39759826)

[4.0.0 - PIP Controller 45](#_Toc39759827)

[4.1.0 - PIP Introduction 45](#_Toc39759828)

[4.2.0 - Second Order Model 47](#_Toc39759829)

[4.2.1 - PIP Controller Poles 47](#_Toc39759830)

[4.2.2 - PIP Controller Disturbance with Varying Poles 50](#_Toc39759831)

[4.2.3 - Monte Carlo Simulation on System 51](#_Toc39759832)

[4.3.0 - Patient Generalised – 73 53](#_Toc39759833)

[4.3.1 - PIP Controller Poles 53](#_Toc39759834)

[4.3.2 - PIP Controller Disturbance with Varying Poles 55](#_Toc39759835)

[4.3.3 - Monte Carlo Simulation on the PG73 system 57](#_Toc39759836)

[4.4.0 - Gain Model 60](#_Toc39759837)

[4.4.1 - Controller Variables 60](#_Toc39759838)

[4.4.2 - Controller Disturbances 62](#_Toc39759839)

[4.4.3 - Monte Carlo Simulation on the Gain System. 62](#_Toc39759840)

[4.5.0 - DLR Eq1 64](#_Toc39759841)

[4.5.1 - Controller Variables 64](#_Toc39759842)

[4.5.2 - Controller Disturbances 66](#_Toc39759843)

[4.5.3 - Monte Carlo Simulation on the DLR Eq1. 67](#_Toc39759844)

[4.6.0 - PIP Conclusions 69](#_Toc39759845)

[4.6.1 – Introduction 69](#_Toc39759846)

[4.6.2 - Speed of the responses 69](#_Toc39759847)

[4.6.3 - Disturbance Control 69](#_Toc39759848)

[4.6.4 - Robust Variables 70](#_Toc39759849)

[5.0.0 - Conclusions 72](#_Toc39759850)

[6.0.0 - Future Work 73](#_Toc39759851)

[6.1.0 – Introduction 73](#_Toc39759852)

[6.2.0 – Testing 73](#_Toc39759853)

[6.3.0 – Integration 73](#_Toc39759854)

[7.0.0 – Critical Reflection 74](#_Toc39759855)

[8.0.0 – Appendices 75](#_Toc39759856)

[8.1.0 – Appendix Contents 75](#_Toc39759857)

[8.2.0 – Patients Data 76](#_Toc39759858)

[8.3.0 – Patient Specific Model 78](#_Toc39759859)

[8.4.0 – Patient Generalised Models, First Order Model and Second Order Model 79](#_Toc39759860)

[8.5.0 – Patient Generalised Models 82](#_Toc39759861)

[8.6.0 – DLR Models 84](#_Toc39759862)

[8.7.0 – Gain Model 86](#_Toc39759863)

[8.7.1 – Gain Research 86](#_Toc39759864)

[8.7.2 – More in-depth Gain Model Research 87](#_Toc39759865)

[8.8.0 – Second Order Model Controller 89](#_Toc39759866)

[8.8.1 – Poles and Disturbances 89](#_Toc39759867)

[8.8.2 – Monte Carlo Simulation 90](#_Toc39759868)

[8.9.0 – PG73 Controller 93](#_Toc39759869)

[8.9.1 – Poles and Disturbances 93](#_Toc39759870)

[8.9.2 – Monte Carlo Simulation 94](#_Toc39759871)

[8.10.0 – Gain Model Controller 97](#_Toc39759872)

[8.10.1 – Controller Variable Testing 97](#_Toc39759873)

[8.10.2 – Monte Carlo Simulation 97](#_Toc39759874)

[8.11.0 – DLR Eq1 Controller 99](#_Toc39759875)

[8.11.1 – Controller Variable Testing 99](#_Toc39759876)

[8.11.2 – Monte Carlo Simulation 100](#_Toc39759877)

[8.12.0 - Proposed Timeline and Actual Timeline 101](#_Toc39759878)

[9.0.0 - References 103](#_Toc39759879)

# List of Figures:

Figure 1 - Shows a generalised PIP controller 12

Figure 2 - Shows all 303 patient data plotted along with desired and critical bounds plotted in red and blue 13

Figure 3 - Results from First Order Model 15

Figure 4 - Shows the model fit and data output for patient 10 (LEFT) and patient 52 (RIGHT) using the first order model 15

Figure 5 - Shows the model fit and data output for patient 84 (LEFT) and patient 119 (RIGHT) using the first order model 15

Figure 6 - Shows the model fit and data output for patient 199 (LEFT) and patient 214 (RIGHT) using the first order model 16

Figure 7 - Shows the best two responses in terms of rt2 from the first order model, patient 29 (LEFT) and patient 141 (RIGHT) 16

Figure 8 - Second Order Model Results 18

Figure 9 - Shows the model fit and data output for patient 30 (LEFT) and patient 39 (RIGHT) using the second order model 18

Figure 10 - Shows the model fit and data output for patient 192 (LEFT) and patient 247 (RIGHT) using the second order model 19

Figure 11 - Shows the model fit and data output for patient 275 (LEFT) and patient 277 (RIGHT) using the second order model 19

Figure 12 - Shows the model fit and data output for patient 46 (LEFT) and patient 141 (RIGHT) using the second order model 20

Figure 13 - Example of the output of script 1 using data from patient 1 22

Figure 14 - Shows patient 1 (LEFT) and 4 (RIGHT) separate model responses compared to their individual output data 22

Figure 15 - Shows patient 157 (LEFT) and 250 (RIGHT) separate model responses compared to their individual output data 23

Figure 16 – Shows patient 76 (LEFT) and 110 (RIGHT) separate model responses compared to their individual output data 23

Figure 17 - Shows patient 256 (LEFT) and 285 (RIGHT) separate model responses compared to their individual output data 24

Figure 18 - Shows patient 15 (LEFT) and 18 (RIGHT) separate model responses compared to their individual output data 25

Figure 19 – Shows patient 85 with a rt2 of 0.3811 (LEFT) and patient 192 with a rt2 of 0.5546 (RIGHT) 26

Figure 20 – Shows patient 247 with a rt2 of -2.1536 (LEFT) and patient 275 with a rt2 of 0.6107 (RIGHT) 27

Figure 21 - Shows patient 36 with a rt2 of 0.5731 (LEFT) and patient 37 with a rt2 of 0.5351 (RIGHT) 28

Figure 22 - Shows patient 61 with a rt2 of -2.6574 (LEFT) and patient 124 with a rt2 of 0.2801 (RIGHT) 28

Figure 23 - Shows patient 291 with a rt2 of 0.8103 (LEFT) and patient 209 with a rt2 of 0.9151 (RIGHT) 29

Figure 24 - Shows the full response of patient 195 (left) and the close up (right) 30

Figure 25 - Shows number and percentage of patients with rt2 over 0.2 and 0.5. NVR = 0, no delay 32

Figure 26 - Shows number and percentage of patients with rt2 over 0.2 and 0.5. NVR = 0.001, no delay 32

Figure 27 - Shows number and percentage of patients with rt2 over 0.2 and 0.5. NVR = 0.01, no delay 32

Figure 28 - Shows number and percentage of patients with rt2 over 0.2 and 0.5. NVR = 0.1, no delay 32

Figure 29 - Shows all c1 and c2 values, nvr equal to 0 32

Figure 30 - Results from equation 1, nvr = 0 33

Figure 31 - Equation 3, nvr = 0.01 33

Figure 32 - Shows Patient 44 variables with NVR equal to 0 and 0.1 34

Figure 33 - Shows patient 109 (LEFT) and 154 (RIGHT) using DLR Eq1 34

Figure 34 - Shows patient 272 (LEFT) and 273 (RIGHT) using DLR Eq1 35

Figure 35 - Shows number and percentage of patients with rt2 over 0.2 and 0.5 using equation 2. NVR = 0 35

Figure 36 - Shows number and percentage of patients with rt2 over 0.2 and 0.5 using equation 2. NVR = 0.01 36

Figure 37- Shows number and percentage of patients with rt2 over 0.2 and 0.5 using equation 2. NVR = 0.1 36

Figure 38 - Shows the results from a gain between -1 and 1 in steps of 0.01 37

Figure 39 - Patient 69 (LEFT) and patient 98 (RIGHT) responses using a gain model of -1.142 39

Figure 40 - Patient 195 (LEFT) and patient 302 (RIGHT) responses using a gain model of -1.142 39

Figure 41 - Output of all Patients using gain of -1.141 40

Figure 42 - The control system for the first order model using PIP 44

Figure 43 - Schematic of second order models control system 46

Figure 44 - Response of the second order system with poles of 0 47

Figure 45 - Response of the second order system with poles of 0.1 47

Figure 46 - Response of the second order system with poles of 0.5 48

Figure 47 - Response of the second order system with poles of -0.5 48

Figure 48 - Shows the response of the SOM system with poles of 0 and disturbance at day 10 49

Figure 49 - Shows the response of the SOM system with poles of 0.1 and disturbance at day 10 49

Figure 50 - Shows the response of the SOM system with poles of 0.5 and disturbance at day 10 50

Figure 51 - Shows the Monte Carlo simulation of a1 (LEFT) and a2 (RIGHT) 50

Figure 52 - Shows the Monte Carlo simulation of b1 and b2 51

Figure 53 - Shows the Monte Carlo simulation of b3 51

Figure 54 – Schematic of the PG73 control system 52

Figure 55 - Response of the PG73 system with poles of 0 52

Figure 56 - Response of the PG73 system with poles of 0.1 53

Figure 57 - Response of the PG73 system with poles of 0.5 53

Figure 58 - Response of the PG73 system with poles of -0.5 54

Figure 59 - Shows the response of the PG73 system with poles of 0 and disturbance at day 10. Step up of 1 (LEFT) and step up of 2 (RIGHT). 54

Figure 60 - Shows the response of the PG73 system with poles of 0.1 and disturbance at day 10. Step up of 1 (LEFT) and step up of 2 (RIGHT). 55

Figure 61 - Shows the response of PG73 system with poles of 0.1, ki = -5 and a step-up disturbance of 2 at day 10. 55

Figure 62 - Shows the response of the PG73 system with poles of -0.5 and disturbance at day 10. Step up of 1 (LEFT) and step up of 2 (RIGHT). 56

Figure 63 - Shows the Monte Carlo simulation of a1 (LEFT) and b1 (RIGHT). 56

Figure 64 - Shows the Monte Carlo simulation of b2 (LEFT) and b3 (RIGHT). 57

Figure 65 - Shows the Monte Carlo simulation of a1 (LEFT) and b1 (RIGHT). 57

Figure 66 - Shows the Monte Carlo simulation of b2 (LEFT) and b3 (RIGHT). 58

Figure 67 - Shows the response of the gain model with f0 and ki of 0.88 and -5 60

Figure 68 - Shows the response of the gain model with f0 and ki of 1 and -3 60

Figure 69 - Shows the response of the Gain Model system with f0 of 0.88 and ki of -5 and disturbance at day 10. Step up of 1 (LEFT) and step up of 2 (RIGHT). 61

Figure 70 - Shows the Monte Carlo simulation of the gain. 61

Figure 71 - Shows patient 30 and 85 response using their individual controller variables 63

Figure 72 - Shows patient 166 and 192 response using their individual controller variables 64

Figure 73 - Shows the response of all patients using common f0 and ki 64

Figure 74 - Shows the patient specific variables with a disturbance at time 10 65

Figure 75 - Shows the patient specific variables with a disturbance at time 10 65

Figure 76 - Shows the Monte Carlo simulation of patient 30 and 85 66

Figure 77 - Shows the Monte Carlo Simulation of patient 166 and 192 66

Figure 78 - Shows the proposed timeline (LEFT) and the Actual timeline (RIGHT) 100

# List of Tables:

[Table 1 - Shows 20 patients with their rt2, mean error and variance of the error using the first order model 16](#_Toc39759958)

[Table 2 - Shows 6 patients and their errors 19](#_Toc39759959)

[Table 3 - The best and worst patient responses with their errors 20](#_Toc39759960)

[Table 4 - Shows the first 20 patients with their rt2, mean error and variance of the error when using the Patient Specific models 24](#_Toc39759961)

[Table 5 – Shows 21 patients with their rt2, mean error and variance of the error using the TF from patient 73 27](#_Toc39759962)

[Table 6 - Shows 21 patients with their rt2, mean error and variance of the error using the TF from patient 216 29](#_Toc39759963)

[Table 7 - Shows the change in values depending on the days chosen 30](#_Toc39759964)

[Table 8 - Shows four patients with their rt2 values, errors and c1 35](#_Toc39759965)

[Table 9 - Shows the best gain models according to rt2 over 0.2 between -4 and 0 37](#_Toc39759966)

[Table 10 - Shows the best gain models according to rt2 over 0.2 between -1.1 and -1.2 in steps of 0.001 38](#_Toc39759967)

[Table 11 - Selection of patients with their rt2 and errors 38](#_Toc39759968)

[Table 12 - Shows the models researched, with percentages, positives and negatives 42](#_Toc39759969)

[Table 13 - Shows the error in order along with the corresponding f0 and ki values 59](#_Toc39759970)

[Table 14 - Shows the number of patients with the corresponding f0 value 63](#_Toc39759971)

[Table 15 - Shows the original, minimum and maximum c1 value for each patient 66](#_Toc39759972)

[Table 16 - Shows the best systems variables for each model system 69](#_Toc39759973)

# 1.0.0 - Introduction

## 1.1.0 - Project Summary

The project is to compare and evaluate several models, with the aim of developing better control of warfarin dosage, using previous data to minimize adverse effects such as thromboembolic events. Potential applications of this include use with individual patients for self-management of warfarin and use for medical professionals to help determine the dosage change.

This is a control project focused on warfarin dosing strategies for long term anticoagulation, in which the output is blood clotting speed (International normalised ratio, INR) and the input, the change of dosage of warfarin.

When deciding on the dosage that should be given to a patient, one approach is to use general estimates for what works best on average for a group of people of a similar age, size, gender, etc. By contrast, personalized medicine has a focus on each patient’s history, to help determine the time-varying ideal dosage e.g. based on their past response to the drug. More generally, a Dynamic Treatment Regime (DTR) is a way to improve the treatment for patients on an individual scale, by adapting the treatment according to a list of decision rules (Murphy, 2003). This is also designed to provide treatment when and if the patient needs it. In the statistical literature, methods exist for such optimal dynamic treatment.

Objectives of this project –

• To research the literature on the application of control engineering to the problems of warfarin dosing, and to use this review to identify common themes and challenges.

• To analyse selected biomedical data sets (provided by supervisor), focusing on estimation of suitable input-output models that could be used for control.

• Research the control problem, from model estimation, control algorithm design, through to systematic simulation-based evaluation, and quantitative comparison between different approaches.

From these objectives, the aims are to:

* To research the medical problem inherent in warfarin and the issues with dosage
* Find previous models and controllers already used for similar applications
* Use statistical models using the data and captain toolbox to create a model
* To develop of at least 1 controller for the problem
* Compare and evaluate the models
* Compare and evaluate the controlled models

## 1.2.0 - Literature Review

### 1.2.1 - Introduction of Warfarin

The aim of this literature review is to understand the reasons why a higher level of control is needed for the dosage of warfarin and to research any previous work in this area of control.

Warfarin was discovered in the 1920s when healthy cattle died from internal bleeding with no obvious cause. It was then marketed as a rodenticide in the late 1940s named as warfarin (Wisconsin Alumni Research Foundation, WARF) and the ‘arin’ from the end of the natural substance coumarin. In early 1950s studies were run to use warfarin as a therapeutic anticoagulant and transitioned into clinical use in humans in 1954 (Lim, 2019).

### 1.2.2 - Control Theory already used in other medication

Use of biomedical control is already being used and improved with other drugs such as insulin yet, like warfarin, further research is needed into ways to improve the controllers. A simple proportional derivative (PD) controller was used for the insulin control problem and was found to follow the trend of the patient data and was within 10% of the insulin dosage and was also able to reduce glucose excursions by 89%, which shows that control can be used for this problem too. The simple controller follows the fundamental dynamics for brief trials but does not work for longer time periods and would need to be improved with the use of versatile controllers such as PID, PIP or MPC (Carmen V, 2005).

### 1.2.3 - Why control is needed in this problem and issues with dosing of warfarin

Control methods are needed for help with dosage of warfarin due to the serious effects this drug can cause, such as deep vein thrombosis (DVT) and pulmonary embolism (PE), which are sometimes referred to as thromboembolic events (Milner and Bonaventura, 2019). They are more widely referred to as thromboembolism. “Thromboembolism: Formation in a blood vessel of a clot (thrombus) that breaks loose and is carried by the blood stream to plug another vessel. The clot may plug a vessel in the lungs (pulmonary embolism), brain (stroke), gastrointestinal tract, kidneys, or leg.” Thromboembolism is a significant cause of morbidity (disease) and mortality (death), especially in adults.” (MedicineNet, 2019).

On the other hand, there can be cases where the clotting factor is found not to be high enough, meaning bleeding doesn’t stop. Data from (Wysowski, Nourjah and Swartz, 2007) shows what a high risk this can be. Even though the data is old, it still demonstrates that without proper control of this drug the effects can be serious, sometimes fatal. In the US between 1993 and 2006, AERS (Adverse Event Reporting System) had a total of 9766 bleeding cases, 86% of which ended in a serious outcome, “serious outcome includes death, hospitalisation, life threatening, disability congenital malformation, and required intervention”, (Wysowski, Nourjah and Swartz, 2007), and 10% were fatal cases. For the same period, for 1.8 million cases reported to AERS for all other drugs, 30% had serious outcomes and only 7% were fatal illustrating how dangerous warfarin can be and how important proper control is with dosage of this drug.

### 1.2.4 - Problems with diet and how it affects the effectiveness of warfarin including how warfarin works

Warfarin stops the production of vitamin K in the body which is needed to make the clotting factors and therefore the ability to form blood clots. INR is used to measure the blood clotting speed, a higher INR means there is a greater risk of bleeding, and a lower INR means there is a greater risk of blood clots (www.heart.org, 2019). The range that INR should be kept in varies from patient to patient and depends on various factors including the purpose for which they are taking the drug. According to (Keeling, 2019) the most common target INR is 2.5 and to keep it with the ranges of +/- 0.5, i.e. 2.0 to 3.0. Milner and Bonaventura (2019)**,** Makohusová and Bátorová (2019) andBaglin, Keeling and Watson (2019) all agreewith this recommendation. However, adjustments are made depending on the medical condition of the patient such that for a patient with recurrent VTE (DVT or PE) the target INR is 3.5 (+/- 0.5) or for a patient undergoing cardioversion the target is 3.0 (+/- 0.5), pre operation and 2.5 (+/-0.5) post operation (Keeling, 2019).

While taking warfarin, the patient’s diet can affect the function of the drug. Alcohol intake should be reduced from approximately 14 units per week (2 units per day) (nhs.uk, 2019) to approximately 9.8 units per week (1.4 units per day) (www.heart.org, 2019). Alcohol increases the risk of bleeding even if the patient’s INR is in the target range.

Certain foods can also alter the effectiveness of warfarin; for example, green and leafy vegetables, such as kale, spinach and cabbage, are high in vitamin K which increases the clotting factors and therefore the risk of blood clots increases. Therefore, to reduce the risk of blood clots, the recommendation given is to keep the intake of vitamin K consistent (www.heart.org, 2019).

Other medications can also disturb the performance of warfarin either increasing or decreasing the patients INR, possibly to an unsafe level. For instance, some over the counter drugs, ibuprofen and naproxen, raise the probability of bleeding when taken alongside warfarin (www.heart.org, 2019).

### 1.2.5 - Previous work with control theory and warfarin

There has been previous work into the control problem of warfarin. However, these have been looking at either the issue of missing data or robust and adaptive control. This previous work can be adapted to be used in the current research project for comparison of controllers. Proportional Integral (PI) and Model Predictive Control (MPC) both worked as controllers for this problem, when the INR was in the target range, the controller would advise similar decisions to the medical professionals, and when the INR was outside the desired range, the controller would advise a dosage change to correct the INR to the target range (Wilson et al., 2019). The conclusions from this were that the MPC outperformed the PI and kept closer to the desired INR. This demonstrates that the MPC has a better response in this application.

For a more robust controller, three models were compared using proportional integral plus (PIP) as the controller for the second two. These three were called deadbeat (DB), linear quadratic optimal (LQ) (both of these assume that the parameters are known), and stochastic robustness analysis (SRA) (which assumes there is an uncertainty and specifies a probability distribution for their estimates). The conclusions were that DB, LQ and SRA followed the desired output, i.e. would keep INR in the target range and if INR went too high, the models would act to decrease INR until back into the target range or act to increase INR when it went too low. LQ and SRA were better models than DB for robustness and stability (Avery et al., 2019).

### 1.2.6 - Speculation into how this project could be implemented in the medical field

Recommendations for further research and implementation of this project could include self-management of warfarin (SMW). Of eligible patients included in a study on SMW, only 35% responded that they were willing to take part in self-testing of their INR. Reasons for this as stated in the paper are “not time for training” or “satisfied with current dosing by provider”(Milner and Bonaventura, 2019). However, patients may not want to risk exposing themselves to dangerous, potentially fatal, side effects by doing the calculations themselves. Therefore, further research could be completed into combining the self-checking INR tool, Coaguchek, and a control system to give a change of dosage directly to the patient. A similar approach could be used for medical professionals, reducing the time needed for them to do the calculations. According to the Coaguchek website (Coaguchek.co.uk, 2019) the device is comparable with the equipment that medical professionals use. However, this is outside the scope of this project.

### 1.2.7 – Transfer Functions

In this project, discrete time transfer functions are used to describe the patient’s response to a change in input. These are in the form (Taylor, Chotai and Young, 2001) where is the output or change in INR, is the input or change in dosage of warfarin and k indicates the time sample. The transfer function can also be referred to by the numerator, bt, and the denominator, at, where and .

The backwards shift operator is used for sample delays where (Taylor, Chotai and Young, 2001) for example indicating two samples previous to sample .

### 1.2.8 – PIP Control

Proportional Integral Plus, PIP, is an extension of Proportional Integral, PI, controllers, allowing for more control of the system. PIP uses pole placement to determine the value of the controller variables while the number of variables depends on the plant being controlled (Taylor, Chotai and Young, 2001).

Using the transfer function from 1.2.7 a PIP system would be:

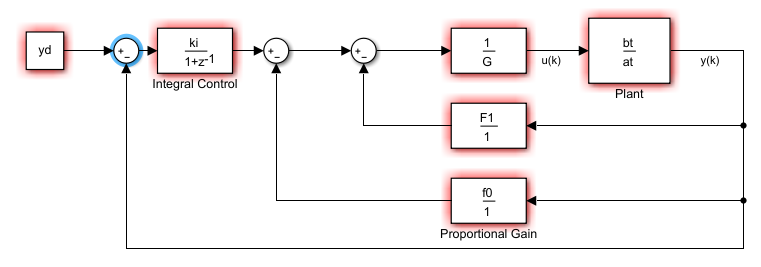


Figure - Shows a generalised PIP controller

Where and

## 1.3.0 - Section Overview

Section 2.0.0 – This introduces the patient data used in the testing of the models.

Section 3.0.0 – This section is the in-depth investigation and research of the models tested in this project, with the major sub-sections being the different models.

Section 4.0.0 – The PIP controller section, testing and evaluating four models chosen from section 3.

Section 5.0.0 – The conclusions from the whole report, deciding on the best system.

Section 6.0.0 – Ideas for future work that will further this project.

Section 7.0.0 – Critical reflection on issues faced in this project and how they were overcome.

Section 8.0.0 – Appendices, including scripts used for model creation and testing, scripts for controller creation and testing and proposed and actual timeline of the project.

Section 9.0.0 – References used throughout this report.

# 2.0.0 - Patients Data

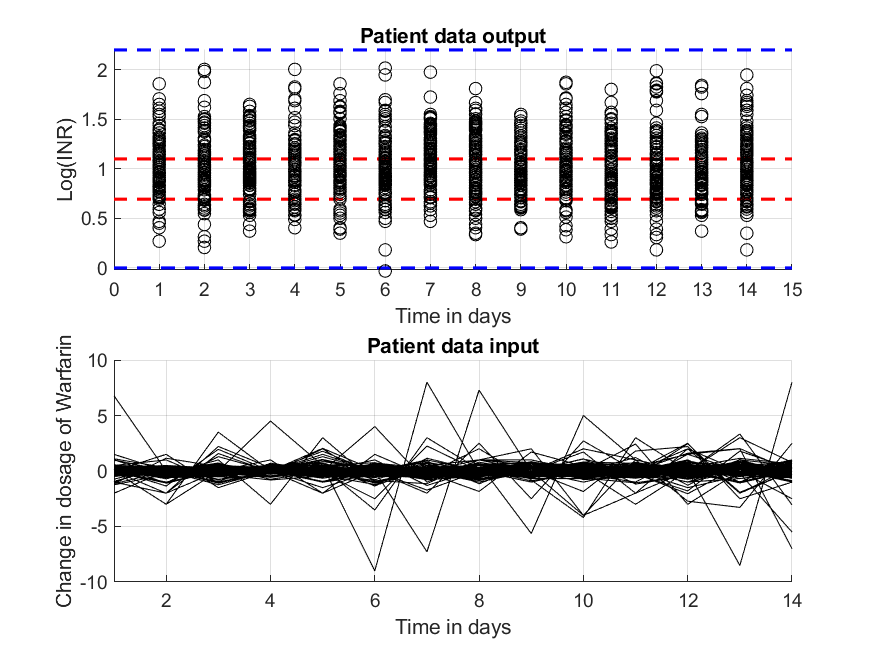


Figure - Shows all 303 patient data plotted along with desired and critical bounds plotted in red and blue

Figure 2 shows all the patients plotted in terms of the log of INR at the desired value of 2.5. This was done by creating a new file of the output data by adding 2.5 and taking the log of that, , where i is the patient number (increased for 1 to 303) and k is the position of the data point (1 to 14).

The red dotted lines show the desired range for the INR, log of 2 and 3. The concentration of the outputs are either within or close to these desired bounds and therefore are useful data points to create models, and test models against. The blue dotted lines show the maximum and minimum safe range and all data points are within this range. The maximum is set to an INR of 9, which is considered as a high risk of bleeding (Pagano and Chandler, 2012) and the minimum is set to an INR of 1.

The dosage for the patients is normally started at 5mg per day (Bragg and Amir, 2003), or between 2mg and 10mg (Cambridge NHS, 2017). However, some patients can have a resistance to warfarin and require higher dosages. This can be due to a genetic cause, or to noncompliance or diet (Cambridge NHS, 2017).

# 3.0.0 - Model Research and Investigation

In this chapter, six main models are investigated, tested, and compared to previous patient data, using several error calculations and graphic responses. Thus, the most useful model is determined.

## 3.1.0 – Model Introduction

### 3.1.1 – Method

The models are tested and compared using 303 patients with 14 corresponding data points which are the input and output data.

The method for evaluating the models had three main comparisons. The first was calculating the rt2 values from each of the responses using . The error used in this equation was calculated by the model response minus the data output, the closer the value of rt2 is to 1 the better the response is. This was calculated for each patient’s response to a model, and the number of patient rt2 values that were higher than 0.2 and 0.5 was found. This gave a general idea to how the model performed, a higher percentage would indicate that the model response followed the trend of the data output and that implied a better model.

The second method for evaluating the models was this: The model responses for a selection of patients was plotted and compared to the rt2 value calculated for that patient. This allowed for verification of the rt2 values calculated and that the responses were or were not following the trend of the data output.

The third main method used in the comparison of the models, was to calculate two errors, the mean absolute error and the variance of the error. The smaller these were, the closer the model response was to the data output. , and , where x is the error, and n is the sample size.

The models could be a general model, that could be used for all patients, or a model that is patient specific, meaning that each patient has an individual variation for their model.

## 3.2.0 - First Order Model

Section 3.2.0 evaluates the First Order Model using rt2 calculations and plotted model responses. The input data from the patients is passed through the model and the output is plotted, the rt2 and errors were calculated as discussed in section 3.1.1.

### 3.2.1 – Introduction

The first order model was obtained from paper 1, Wilson et al., 2019, given as , where . This can be rearranged as seen below.

In MATLAB the model was tested by using the filter command: filter(bt, at, x), with each patient’s input data x (change in dosage of warfarin in mg), the numerator of the model bt and the denominator of the model at. For the model below, bt = [0 0.25] and at = [1 -0.4].

The Model:

### 3.2.2 – Comparing Model Output and Data Output for First Order Model

A screenshot of a cell phone

Description automatically generated

Figure - Results from First Order Model using all patient data

Using the results from figure 3, the model does not capture any of the outputs, the average rt2 being -0.0423, which is far from the ideal value of 1. Only 2 patients from a set of 303 have a rt2 over 0.2, which is 0.66% and 0 have a rt2 over 0.5.

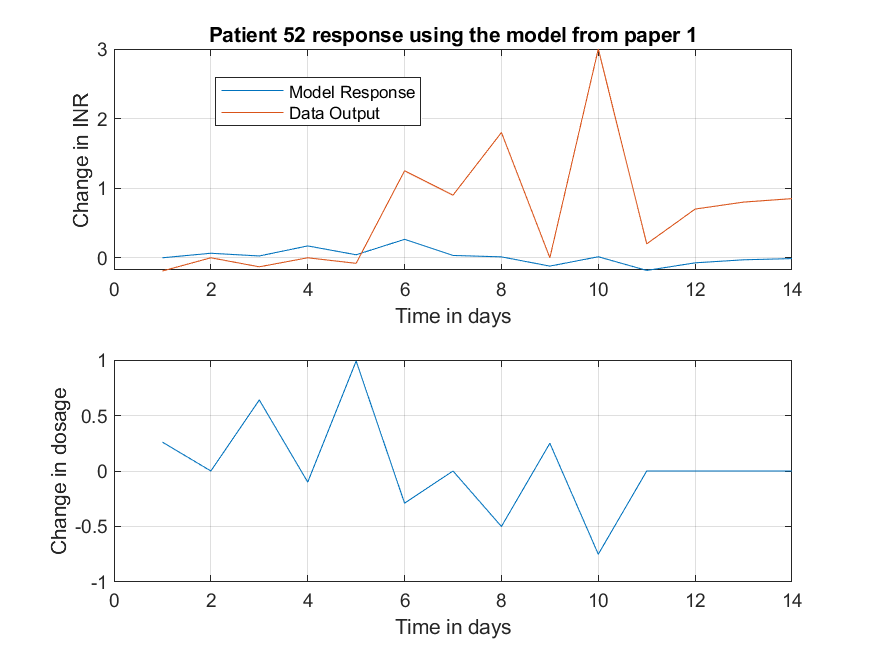
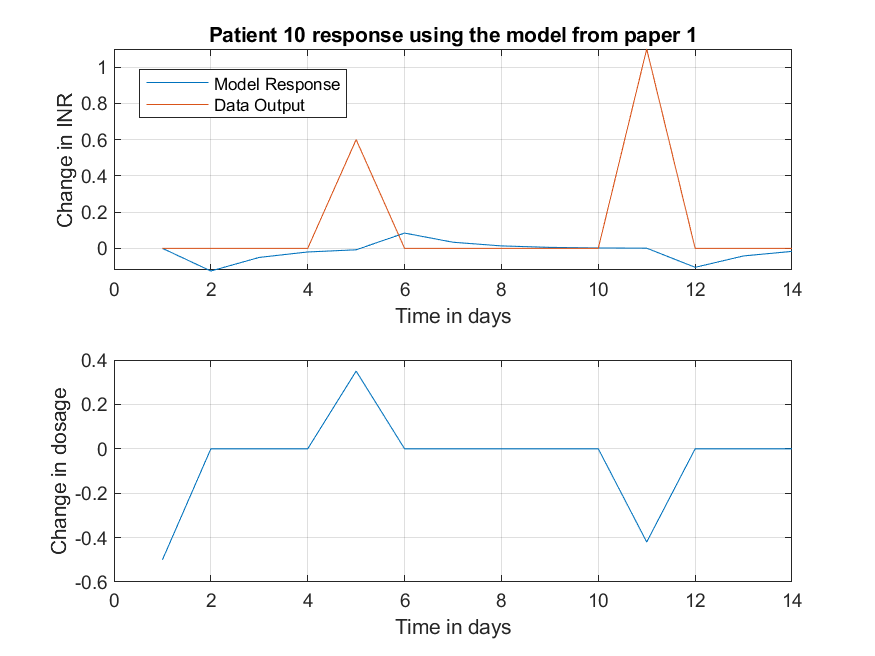


Figure - Shows the model fit and data output for patient 10 (LEFT) and patient 52 (RIGHT) using the first order model

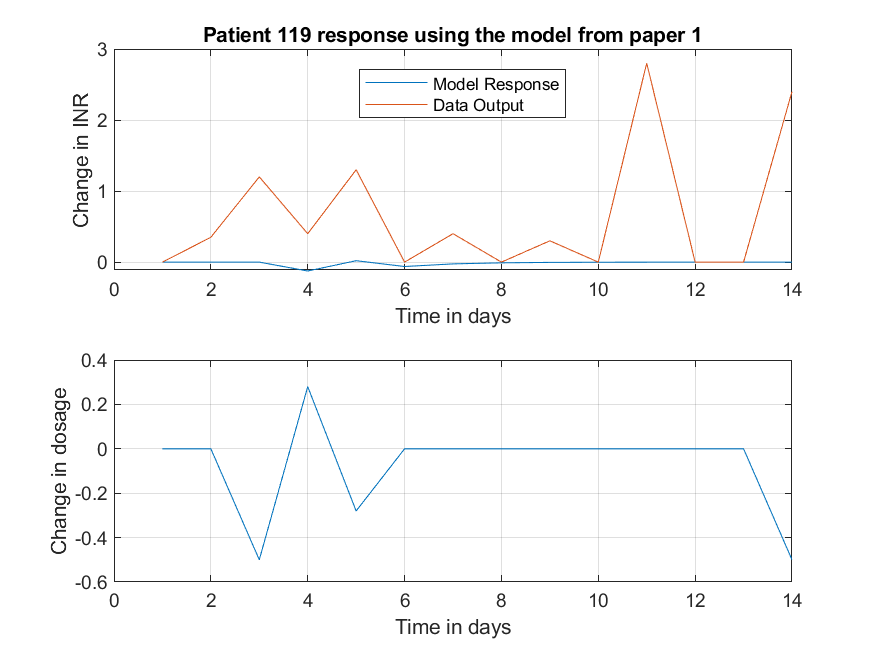
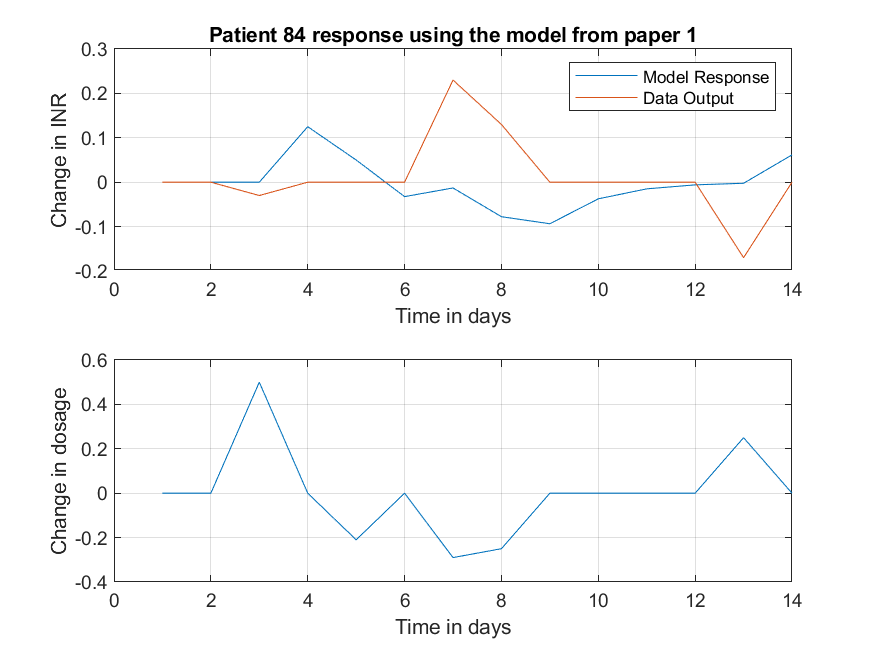


Figure - Shows the model fit and data output for patient 84 (LEFT) and patient 119 (RIGHT) using the first order model

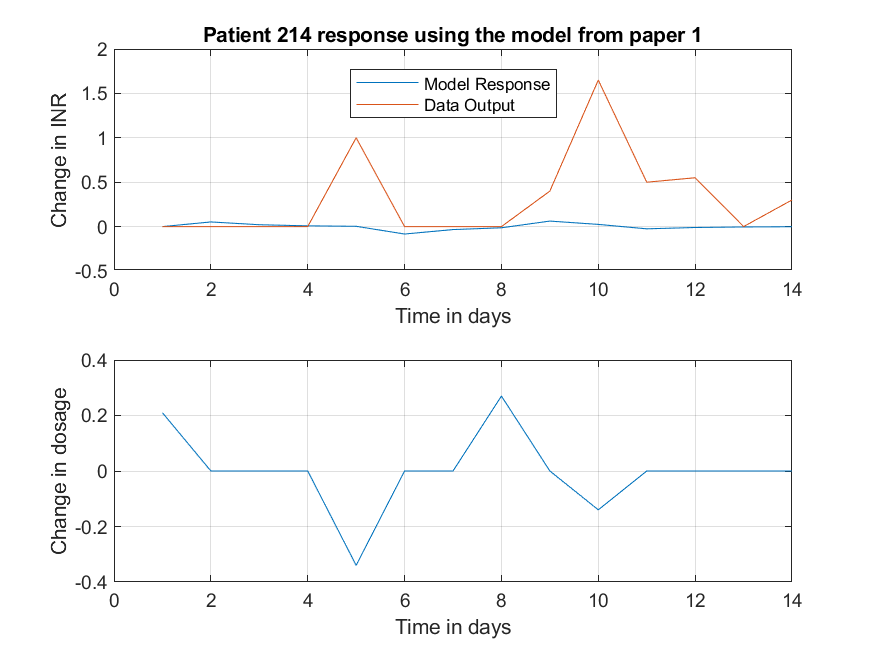
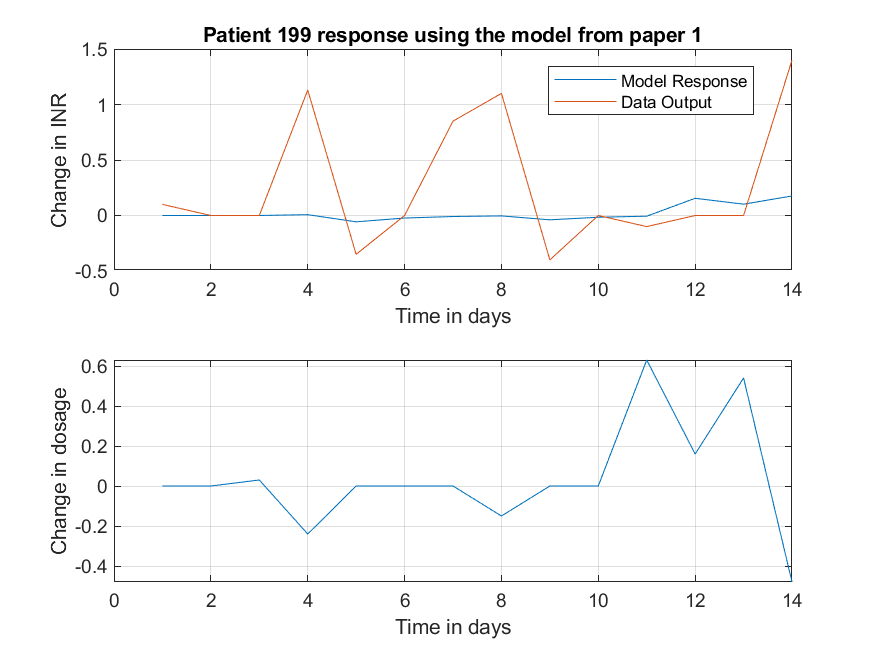


Figure - Shows the model fit and data output for patient 199 (LEFT) and patient 214 (RIGHT) using the first order model

Figures 4 to 6 show the disappointing responses using the first order model. The model doesn’t capture any of the patient’s data output.

The best two model responses, in terms of rt2 were patient 29 and 141 and their responses were less than desired as seen below.

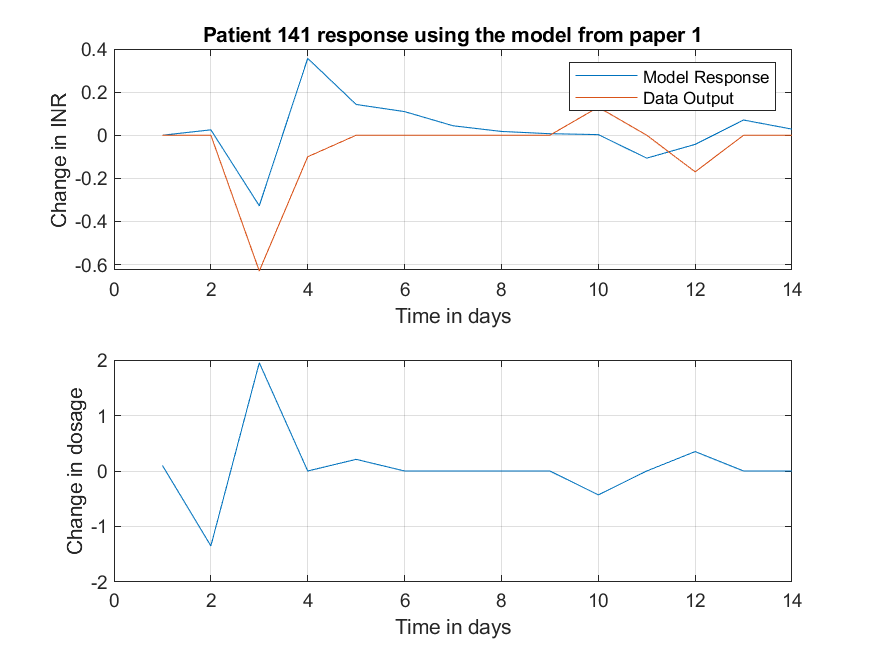
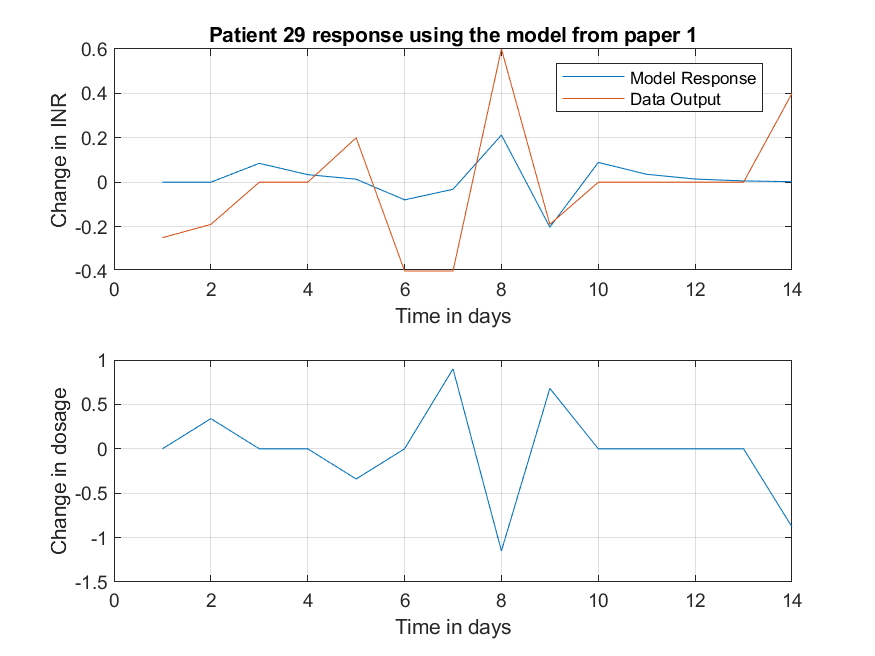


Figure - Shows the best two responses in terms of rt2 from the first order model, patient 29 (LEFT) and patient 141 (RIGHT)

As seen from these eight patients’ model output and data outputs, the model is not acceptable for this application.

Table - Shows 20 patients with their rt2, mean error and variance of the error using the first order model

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| PatientNumber | RT2 | MeanError | VarError | Over0x2 | Over0x5 |
| 1 | -0.071831242 | 0.824016775 | 0.656285803 | 2 | 0 |
| 2 | -0.158833503 | 0.638171591 | 0.580762145 | 0 | 0 |
| 3 | -0.064972158 | 0.243809461 | 0.100383573 | 0 | 0 |
| 4 | -0.138323094 | 0.439616028 | 0.435350416 | 0 | 0 |
| 5 | -0.144353778 | 0.381386556 | 0.424528215 | 0 | 0 |
| 6 | -0.061128183 | 0.379777494 | 0.281092273 | 0 | 0 |
| 7 | -0.257040381 | 0.423588209 | 0.261768299 | 0 | 0 |
| 8 | -0.153916088 | 0.815152783 | 1.405833089 | 0 | 0 |
| 9 | -0.050245603 | 0.365102629 | 0.248387124 | 0 | 0 |
| 10 | -0.079707231 | 0.146020268 | 0.11325061 | 0 | 0 |
| 11 | -0.100601973 | 0.401055414 | 0.721104737 | 0 | 0 |
| 12 | -0.005216201 | 0.585762123 | 1.476397488 | 0 | 0 |
| 13 | -1.688321005 | 0.154547023 | 0.069259716 | 0 | 0 |
| 14 | -0.301083176 | 0.319088992 | 0.484037971 | 0 | 0 |
| 15 | -0.333169126 | 0.136249282 | 0.046198705 | 0 | 0 |
| 16 | -0.027305028 | 0.569970018 | 0.478952746 | 0 | 0 |
| 17 | -0.497853005 | 0.741560887 | 0.990146676 | 0 | 0 |
| 18 | -0.09640078 | 0.576936662 | 1.190265338 | 0 | 0 |
| 19 | -0.044973287 | 0.671218596 | 0.738658315 | 0 | 0 |
| 20 | -0.13973174 | 0.131650731 | 0.039127241 | 0 | 0 |

None of these first 20 patients have a rt2 greater than 0. With all 303 patients, 295 or 97.4% of the patients have a rt2 less than 0, only 8 patients have a rt2 greater 0.

## 3.3.0 - Second Order Model

Section 3.3.0 evaluates the Second Order Model using rt2 calculations and plotted model responses. The input data from the patients is passed through the model and the output is plotted. The rt2 and errors were calculated as discussed in section 3.1.1.

### 3.3.1 – Introduction

The second order model was obtained from paper 2, Avery et al., 2019, given as , where . This can be rearranged as seen below.

The model was filtered using MATLAB the same way as the first order model. For this second order model bt = [0 0.2025 0.2056] and at = [1 -0.2324 -0.1491].

The Model:

### 3.3.2 – Comparing Model Output and Data Output for Second Order Model

A picture containing screenshot

Description automatically generated

Figure - Second Order Model Results

Figure 8 shows the average, maximum and minimum rt2 values from all the 303 patients. Only 14 patients had a rt2 value greater than 0.2 and 0 patients have a rt2 value over 0.5.

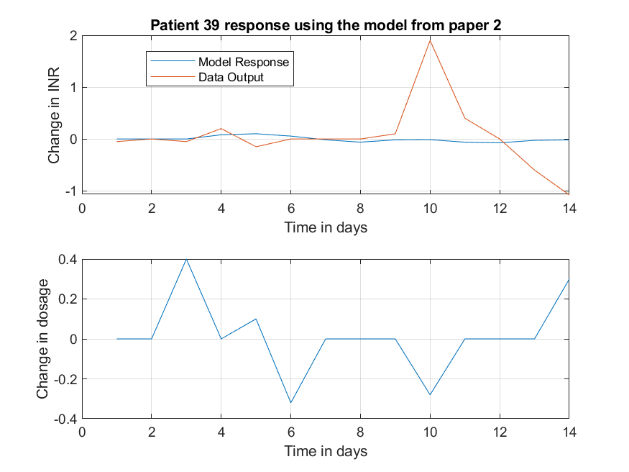
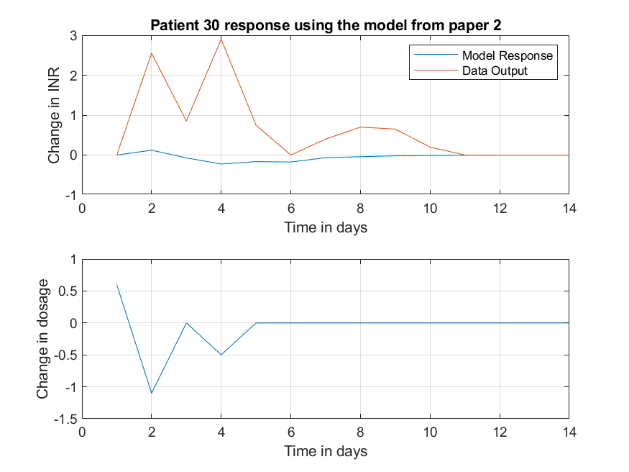


Figure - Shows the model fit and data output for patient 30 (LEFT) and patient 39 (RIGHT) using the second order model

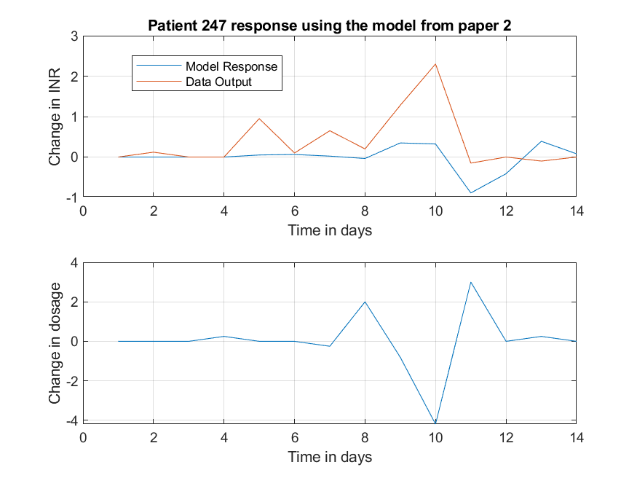
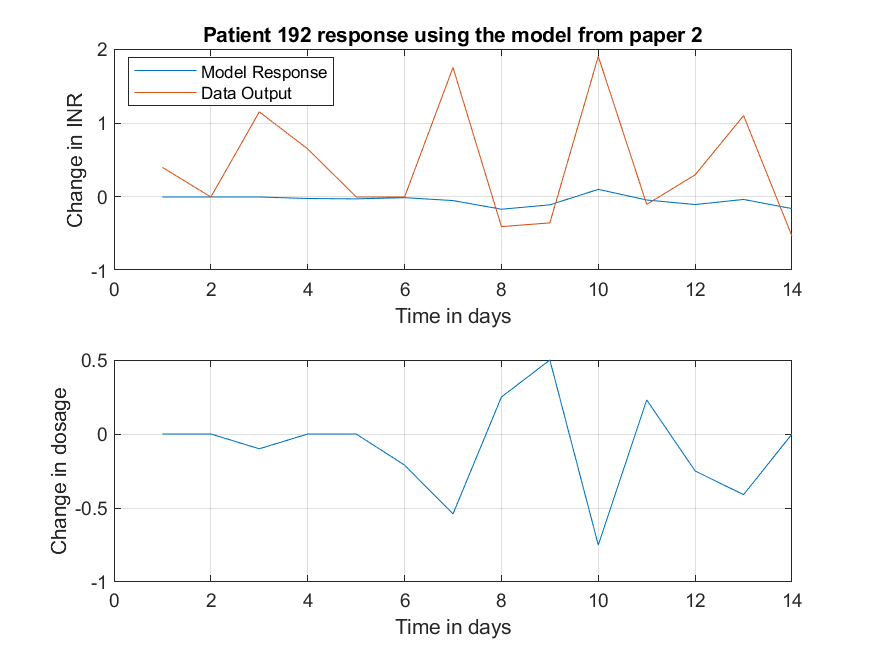


Figure - Shows the model fit and data output for patient 192 (LEFT) and patient 247 (RIGHT) using the second order model

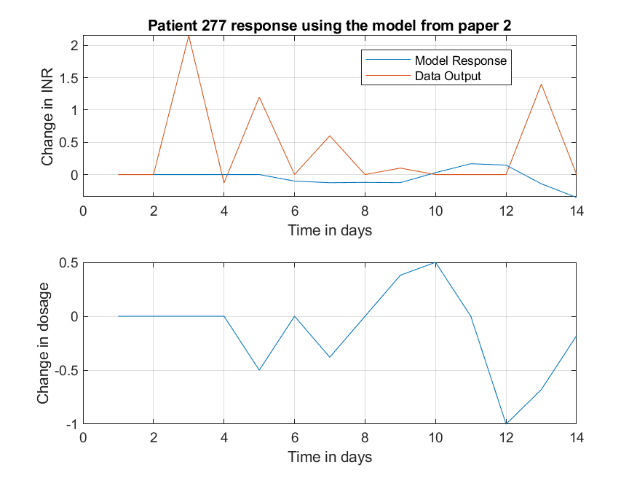


Figure - Shows the model fit and data output for patient 275 (LEFT) and patient 277 (RIGHT) using the second order model

These figures show that the second order model does not capture the data output. From figure 9 to 11 the rt2 values are -0.0386, -0.0115, 0.1109, 0.2198, 0.0406 and -0.0510, patient 247 having the best rt2 value.

Table - Shows 6 patients and their errors

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| PatientNumber | RT2 | MeanError | VarError | Over0x2 | Over0x5 |
| 30 | -0.03857 | 0.689929 | 0.920674 | 14 | 0 |
| 39 | -0.01149 | 0.343249 | 0.415631 | 0 | 0 |
| 192 | 0.110914 | 0.592496 | 0.546425 | 0 | 0 |
| 247 | 0.219804 | 0.467642 | 0.378445 | 0 | 0 |
| 275 | 0.040622 | 0.101057 | 0.022671 | 0 | 0 |
| 277 | -0.05099 | 0.492581 | 0.516426 | 0 | 0 |

From this table, patient 247 has a large error, but not the greatest in this selection. Patient 275 has the smallest error for both mean of the error and the variance of the error and this can be verified by figure 11. From this selection the best response would be patient 275 as it has the smallest error.

Table - The best and worst patient responses with their errors

|  |  |  |  |
| --- | --- | --- | --- |
| PatientNumber | RT2 | MeanError | VarError |
| 141 | 0.340524 | 0.118364 | 0.020817 |
| 46 | -1.37394 | 0.056318 | 0.006946 |

Table 3 shows the best and worst patient responses according to their rt2 value. Both patients have low errors, however, patient 46 (the worst response according to rt2) has errors 10 times less than patient 141.

A close up of a map

Description automatically generatedA close up of a map

Description automatically generated

Figure - Shows the model fit and data output for patient 46 (LEFT) and patient 141 (RIGHT) using the second order model

These show that the errors can be deceptive in deciding which is the best response. Patient 141 clearly has a response that is similar to the data output, whereas patient 46 seems to have no relevance to the data output.

## 3.4.0 - Patient Specific Models

Section 3.4.0 evaluates the Patient Specific Models using rt2 calculations and plotted model responses. The input and output data of a patient are used to create a specific model for that patient. This is done for all patients, and a selection are plotted to be evaluated. The creation of the models starts with one denominator and numerator value with no delay, up to 3 numerators and denominators with a 5-sample delay. The rt2 and errors were calculated as discussed in section 3.1.1.

### 3.4.1 - Brief Description of How RIVID Works

Using CAPTAIN toolbox, MATLAB can calculate the best fitting transfer function (TF) for a specific set of data. The function RIVID is set out as such: [th, stats, e] = rivid (z, [1 1 0 0; 3 3 5 0], sc). Th is the theta matrix which hold numerous values such as ‘at’ and ‘bt’ which are the denominators and numerators respectively. The function stats() also contains various values, most importantly rt2. E is the model error, where y = fit + e. Z is the data inputted to the function, in this example z = [patient output, patient input]. [1 1 0 0; 3 3 5 0] is the range for the section to calculate in, the form this is written in is [at, bt, time delay, noise]. So, this searches from one denominator and numerator, no time delay or noise to three denominators and numerators, 5 step time delay and no noise. SC prints the table in order of what value this is, 2 is normally chosen as this is in order of rt2.

### 3.4.2 - Individual Transfer Functions

Script 8.3.0 (found in the appendix), first made by James Taylor, chooses a patient, finds the best TF for this set of data, plots this alongside the actual output and input data, prints a table of the stats calculated in order of rt2 and gives the value for ‘at’ and ‘bt’ and finally prints the value of rt2 (my\_rt2) calculated in the script, not the function. An example of the output of this script is seen below in figure 13.

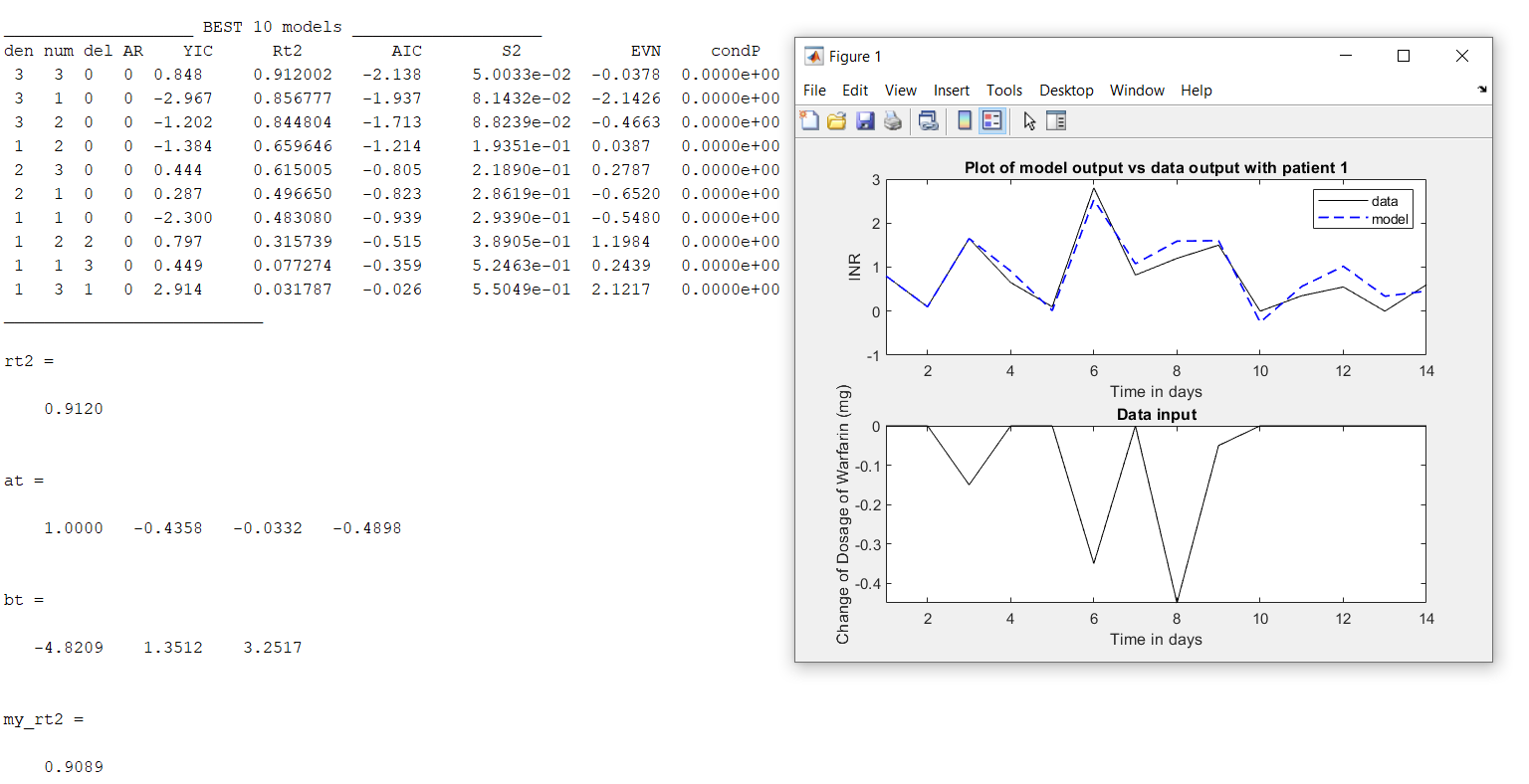


Figure - Example of the output of script 8.3.0 using data from patient 1

As shown, the value for rt2 and my\_rt2 are slightly different. This is because my\_rt2 is calculated in the same way as discussed in section 3.1.1, whereas rt2 is calculated by RT2+RT2AR\*(1-RT2), therefore there are some slight differences between these values. YIC, AIC, EVN and condP are extra calculations made by the function but are not used in this project.

Script 8.3.0 was used to find a few patients with high values of rt2 and a few with lower values of rt2. A high value of rt2 was considered as greater than 0.5, which clearly showed a response of the model that was like the recorded output from the data. This can be seen from patients 1, 4, 157 and 250 in figures 14 and 15. The rt2 for each are 0.9120, 0.8214, 1 and 0.9754 respectively. These four patients have separate TF with different variables and coefficients.

Figure 14 - Shows patient 1 (LEFT) and 4 (RIGHT) separate model responses compared to their individual output data

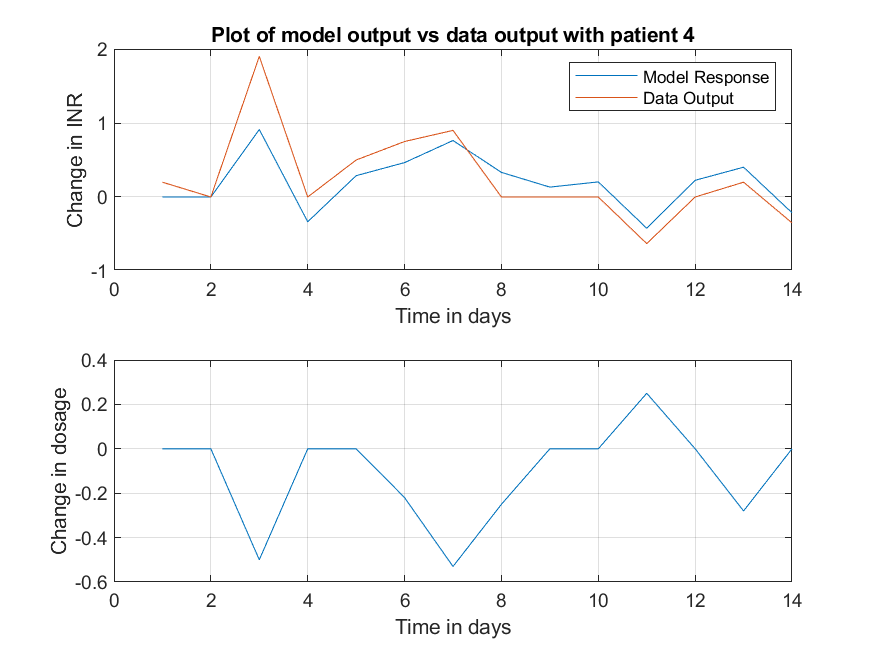
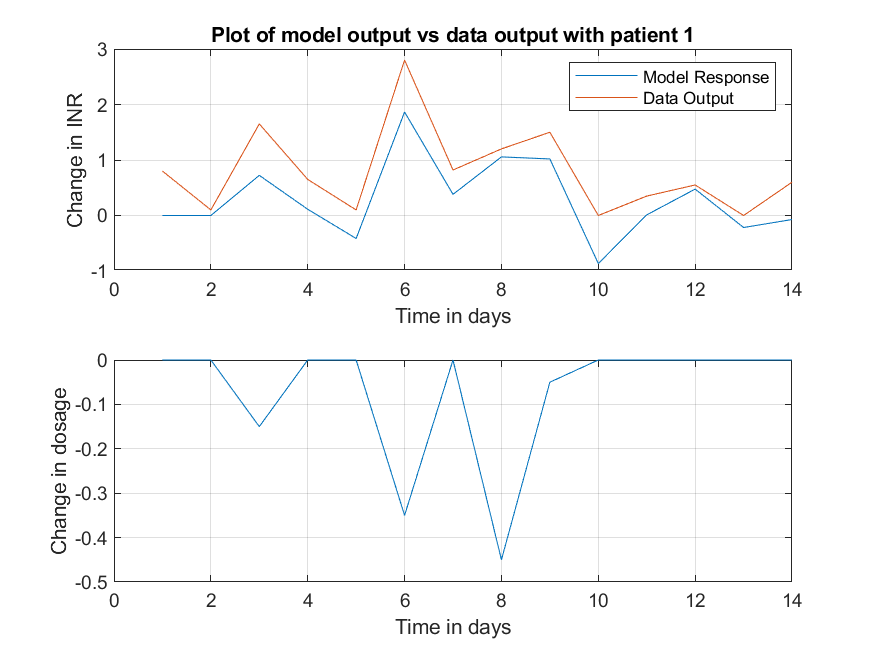
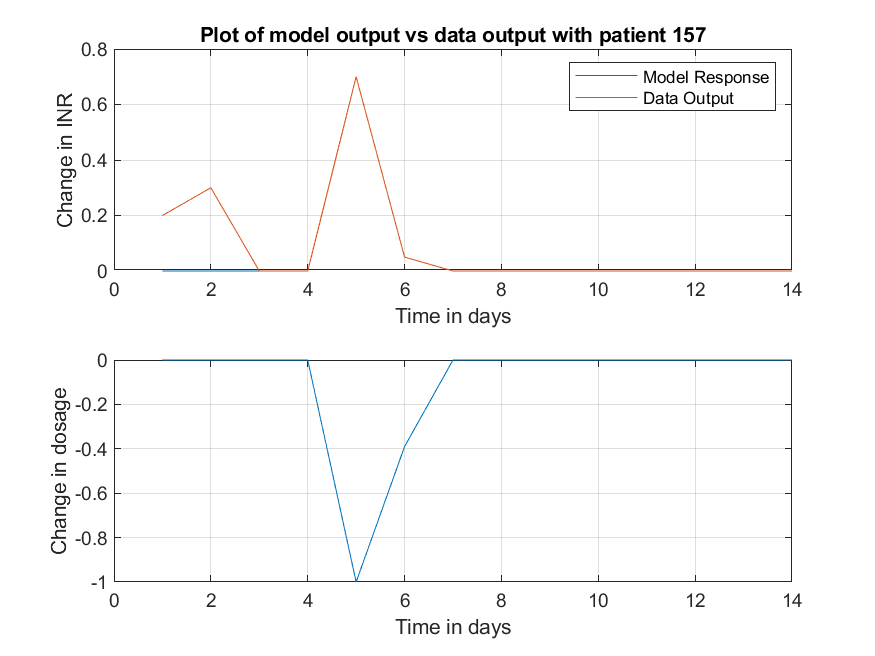


Figure 15 - Shows patient 157 (LEFT) and 250 (RIGHT) separate model responses compared to their individual output data



For patient 4, the TF is , this gives a good response and could have been a viable model for this patient.

For patient 250 the TF is , this is a significantly different transfer function to the TF of patient 4. Patient 4’s data would not have such good approximation of the output. This is addressed further on.

A lower rt2 would be a value less than 0.2, which shows the output of the model not keeping on track with the actual output, such as patients 76, 110, 256 and 285 shown in figure 16 and 17. As shown the model output for all four patients barely captures the same output as the data which means it is a ‘bad’ best model for these patients. Again, these four patients have separate TF and values.

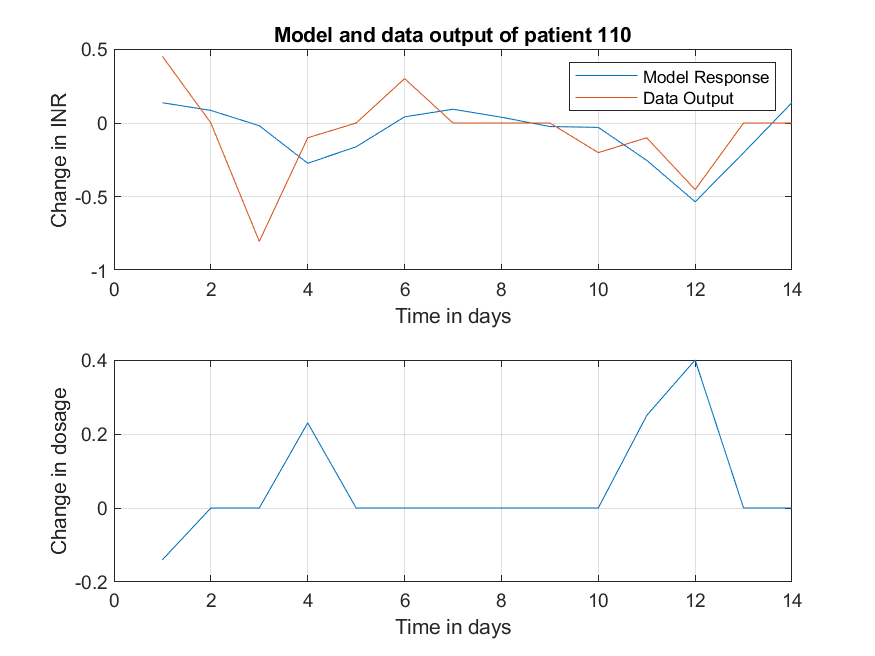
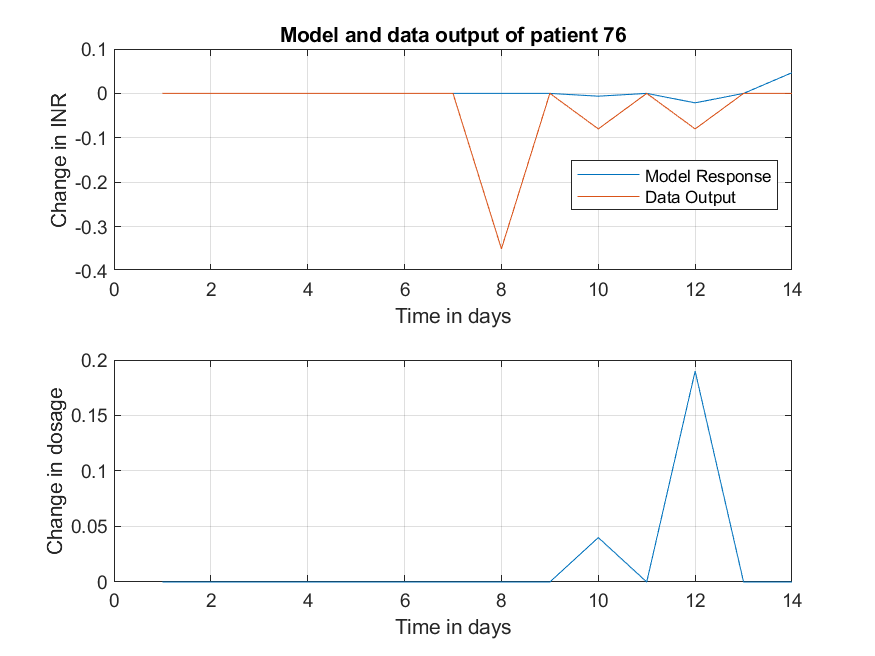


Figure – Shows patient 76 (LEFT) and 110 (RIGHT) separate model responses compared to their individual output data

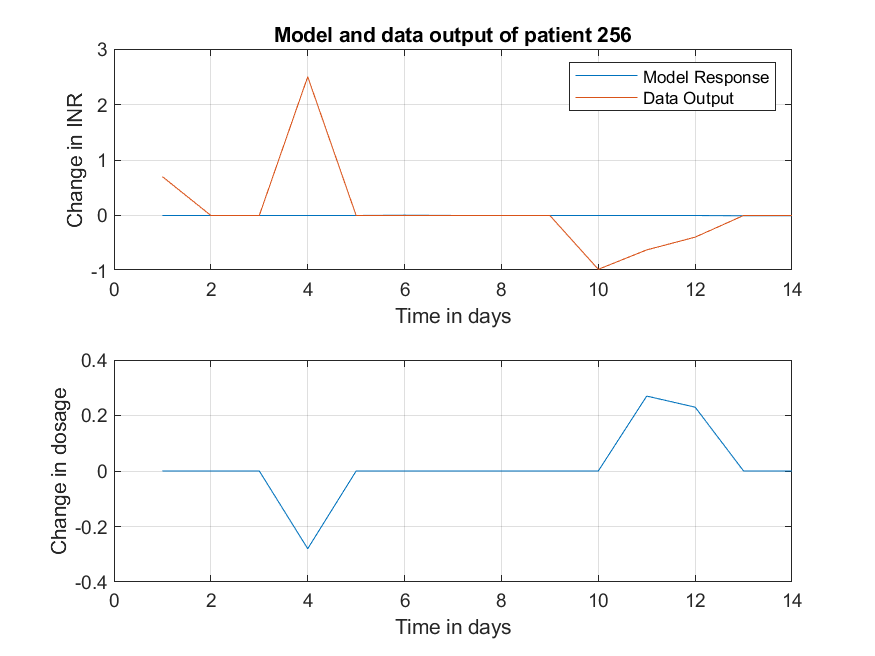


Figure - Shows patient 256 (LEFT) and 285 (RIGHT) separate model responses compared to their individual output data

The rt2 value for each of patients 76, 110, 256 and 285 are 0.0270, 0.1494, 2.218e-5 and 0.1808 respectively. These are well below the threshold of 0.2, apart from 285, which means these are not viable models for this application.

For patient 76, the TF is . This is the best TF that RIVID could find, with a rt2 of only 0.0270. This model would not be suitable for modelling this patient as it does not follow the data trends.

All the rt2 values were calculated and set out in a table along with the mean of the absolute error and the variance of each error. The first 20 are seen in table 4.

Table - Shows the first 20 patients with their rt2, mean error and variance of the error when using the Patient Specific models

|  |  |  |  |
| --- | --- | --- | --- |
| PatientNumber | Rt2 | MeanError | VarienceError |
| 1 | 0.8486 | 0.5033 | 0.0927 |
| 2 | 0.4704 | 0.4642 | 0.2654 |
| 3 | 0.6385 | 0.1623 | 0.0341 |
| 4 | 0.6860 | 0.2571 | 0.1201 |
| 5 | 0.3770 | 0.4141 | 0.2311 |
| 6 | 0.1471 | 0.4035 | 0.2259 |
| 7 | 0.6730 | 0.2259 | 0.0681 |
| 8 | 0.5627 | 0.5524 | 0.5328 |
| 9 | -2.7142 | 1.2051 | 0.8784 |
| 10 | -0.0480 | 0.1473 | 0.1099 |
| 11 | 0.2497 | 0.3411 | 0.4916 |
| 12 | -1199459.1158 | 573.2899 | 1761690.5691 |
| 13 | 0.4654 | 0.0964 | 0.0138 |
| 14 | 0.7748 | 0.2221 | 0.0838 |
| 15 | -0.1806 | 0.1677 | 0.0409 |
| 16 | 0.5924 | 0.3427 | 0.1901 |
| 17 | -0.8205 | 0.8318 | 1.2034 |
| 18 | 0.8509 | 0.2984 | 0.1618 |
| 19 | 0.3469 | 0.4073 | 0.4616 |
| 20 | 0.7354 | 0.0803 | 0.0091 |

The varience of the error, along with the rt2 value, indicates how similar the model response and the data output are; the smaller the value of the varience of the error, the better the response. However, that does not hold true for all the patients, such as patient 18 who has a high value for rt2, 0.8509, but not as low a value for the varience, 0.1618, compared to patient 15, 0.0409.

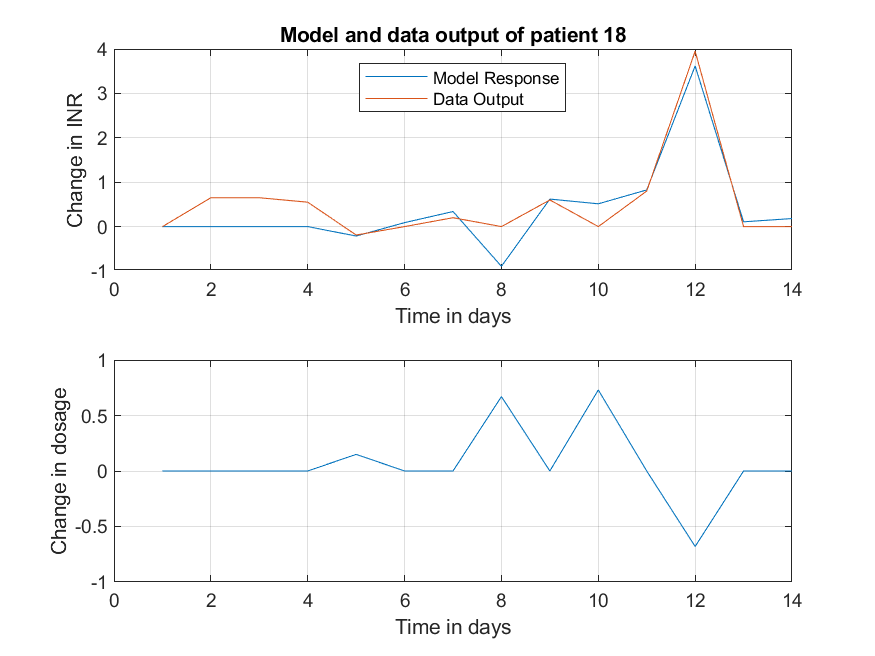
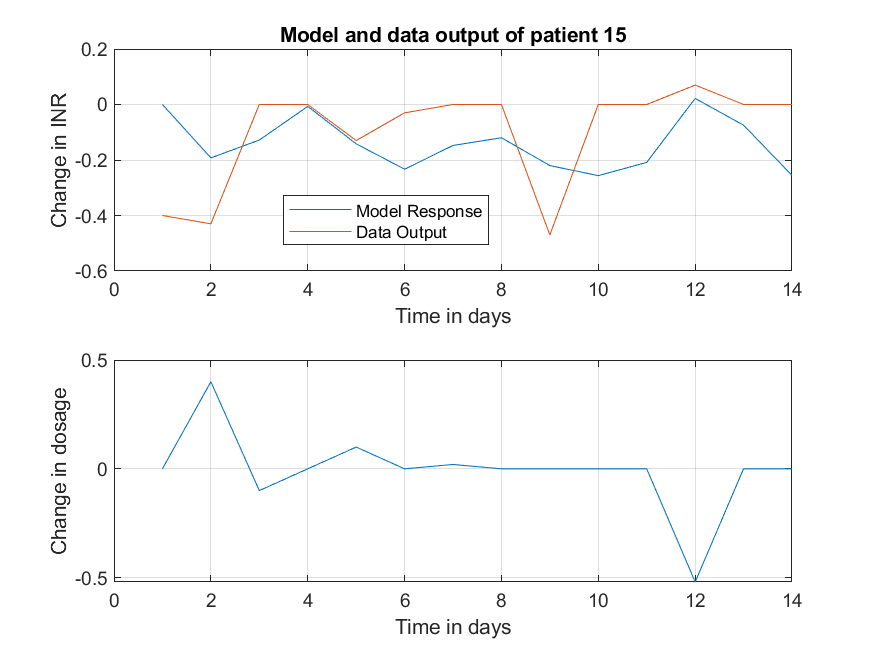


Figure - Shows patient 15 (LEFT) and 18 (RIGHT) separate model responses compared to their individual output data

Patient 18’s reponse follows the data trend more than patient 15, as shown by the rt2 value. However, the error value is larger for patient 18 compared to patient 15. The mean error (calculated by taking the mean of the absolute values of the error) is more sporadic. For example, patients 1 and 18 have similar rt2 values but drastically different mean error values; patient 1 is 0.5033 and patient 18 is 0.2984. This error term is less helpful in determining how successful the model response is.

Using this approach to model each patient has a fairly successful outcome, with 68.3% of patient models having a rt2 value over 0.2 and 42.2% over 0.5. However, the models are vastly different and previous data is needed to create each of the models. Therefore this is not the ideal approach for finding the best model for this application.

## 3.5.0 - Generalised Patient Models using RIVID

Section 3.5.0 evaluates the Patient Specific Models as Patient Generalised models using rt2 calculations and plotted model responses. The input data from all 303 patients are passed through the Patient Specific model for one patient and the rt2 calculated. The Patient Specific model is changed, and the process repeated, and these models are ordered in terms of percentage of rt2 over 0.2 and 0.5. The rt2 and errors were calculated as discussed in section 3.1.1.

### 3.5.1 – Introduction

Script 8.3.0 was edited to input one patients input data into another patients TF, this is script 8.5.0. This is done by using the filter function. First the script finds the best TF for patient A in the same way shown in script 8.3.0. Then, making a new variable, ya = filter (bt, at, z(:, 2)); where bt and at are from patient A’s transfer function. The input from patient B is z(:, 2). Next, ya is used to find the error and then the rt squared value for patient B. Patient B is cycled through all patients to find all rt squared values and calculates the percentage over a value of 0.2 and 0.5. Patient A is then changed and the process starts again until all patients have been tested.

The best generalised transfer functions found were those of patients 73 and 216. Patient 73 had 56.1% rt2 over 0.2 and 16.5% over 0.5, patient 216 had 48.8% over 0.2 and 17.8% over 0.5. These are much better results compared to the first and second order models. A selection of the responses have been plotted below. Patient 73 transfer function is and paitent 216 is

### 3.5.2 – Comparing Model Output and Data Output for PG73 Model

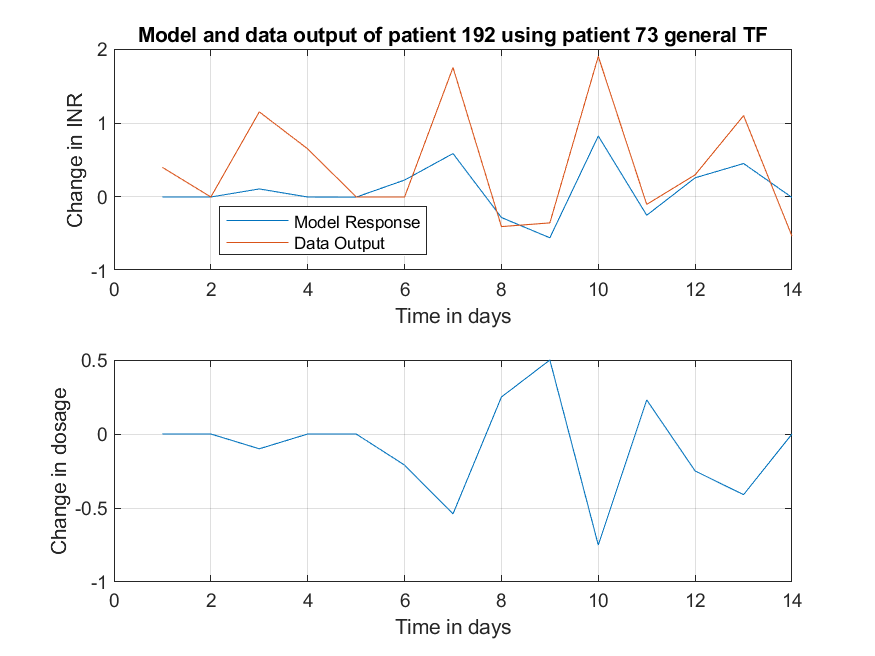
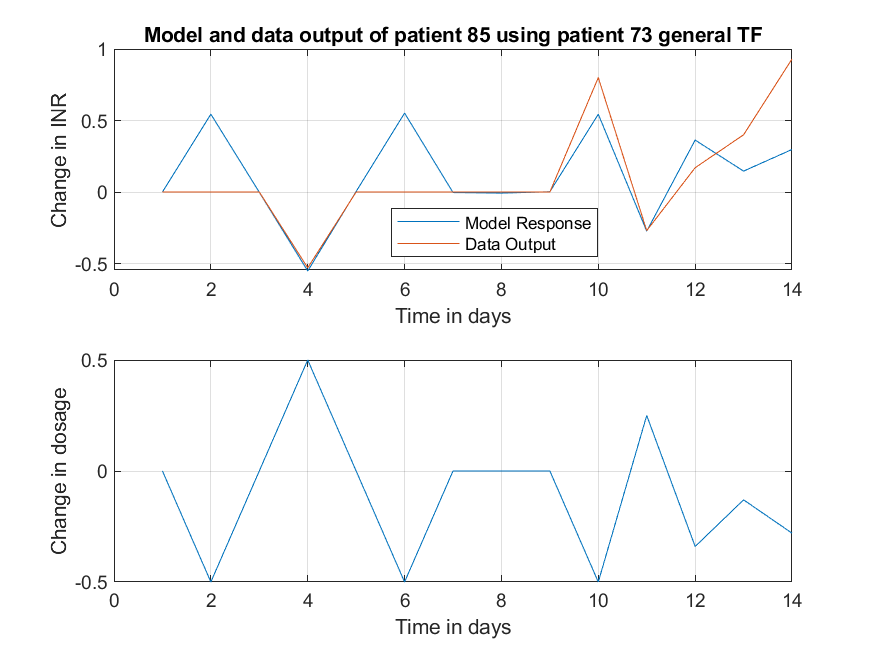


Figure – Shows patient 85 with a rt2 of 0.3811 (LEFT) and patient 192 with a rt2 of 0.5546 (RIGHT)

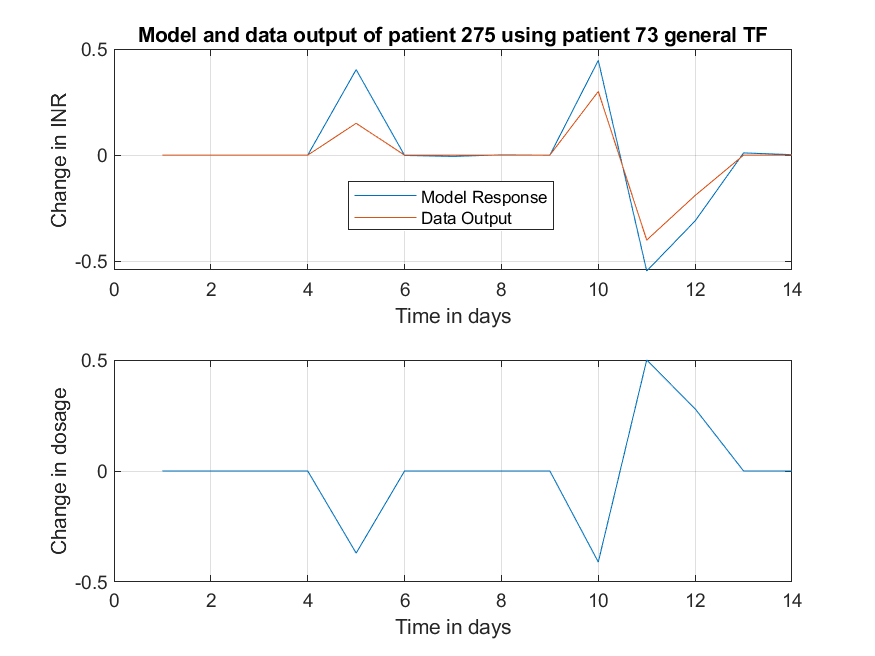
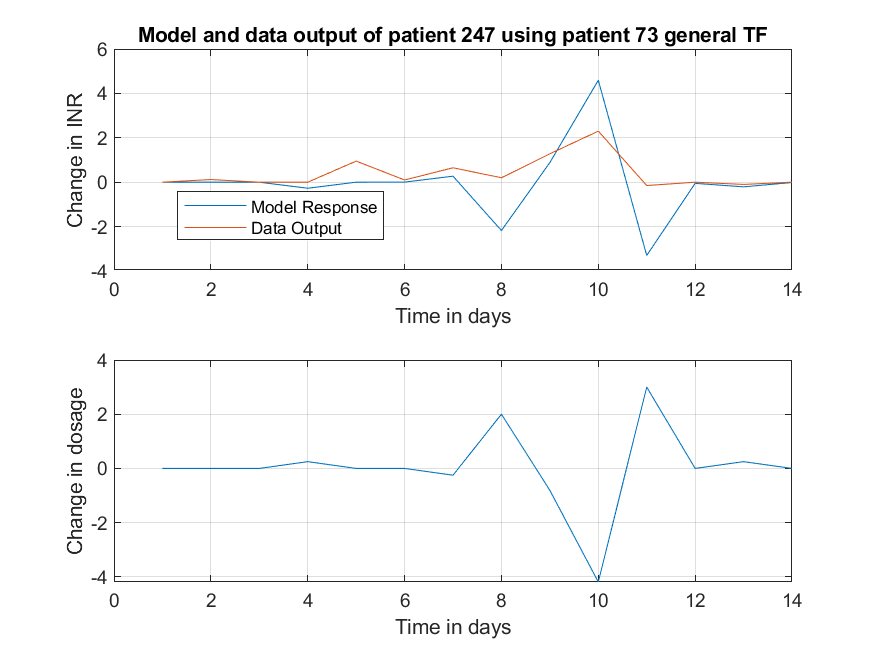


Figure – Shows patient 247 with a rt2 of -2.1536 (LEFT) and patient 275 with a rt2 of 0.6107 (RIGHT)

Figure 19 (RIGHT) and 20 (RIGHT) show good responses, rt2 values greater than 0.5, figure 19 (LEFT) shows an acceptable response and finally, figure 20 (LEFT) shows an unacceptable response.

Table – Shows 21 patients with their rt2, mean error and variance of the error using the TF from patient 73

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| PatientNumber | RT2 | MeanError | VarError | Over0x2 | Over0x5 |
| 33 | 0.469113 | 0.571081 | 0.300658 | 170 | 50 |
| 34 | 0.657185 | 0.1224 | 0.041099 | 0 | 0 |
| 35 | -1.72615 | 0.223238 | 0.096745 | 0 | 0 |
| 36 | 0.573074 | 0.418827 | 0.497891 | 0 | 0 |
| 37 | 0.535085 | 0.182401 | 0.118367 | 0 | 0 |
| 38 | -0.13558 | 0.176845 | 0.07096 | 0 | 0 |
| 39 | 0.269625 | 0.320646 | 0.300119 | 0 | 0 |
| 40 | 0.482009 | 0.121376 | 0.034141 | 0 | 0 |
| 41 | 0.810093 | 0.097384 | 0.024409 | 0 | 0 |
| 42 | 0.277265 | 0.791968 | 1.302249 | 0 | 0 |
| 43 | -0.23655 | 0.225351 | 0.095594 | 0 | 0 |
| 44 | -0.01638 | 0.63237 | 0.527917 | 0 | 0 |
| 45 | 0.46955 | 0.355037 | 0.215736 | 0 | 0 |
| 46 | -2.78195 | 0.055068 | 0.011065 | 0 | 0 |
| 47 | -7.94252 | 1.157853 | 6.293841 | 0 | 0 |
| 48 | -3.14473 | 0.44847 | 0.592559 | 0 | 0 |
| 49 | -0.76461 | 0.9928 | 3.085083 | 0 | 0 |
| 50 | -0.16312 | 0.087162 | 0.027177 | 0 | 0 |
| 51 | -0.14393 | 0.168672 | 0.117237 | 0 | 0 |
| 52 | 0.557157 | 0.704157 | 0.36277 | 0 | 0 |
| 53 | 0.200037 | 0.388043 | 0.265264 | 0 | 0 |

The same issues with comparing mean error and the varience of the error are present here as disscussed with reference to table 4. Table 5 shows the patient number and the rt2 values for each patient using patient 73’s model. Even with a higher percentage of the patient’s rt2 being better, there are still some patients with low values such as patient 47.

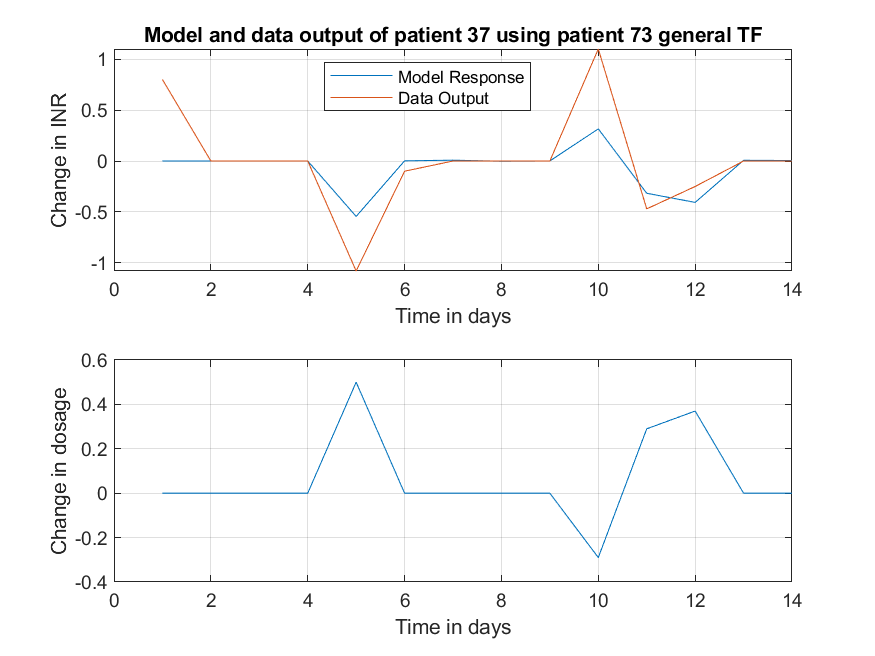
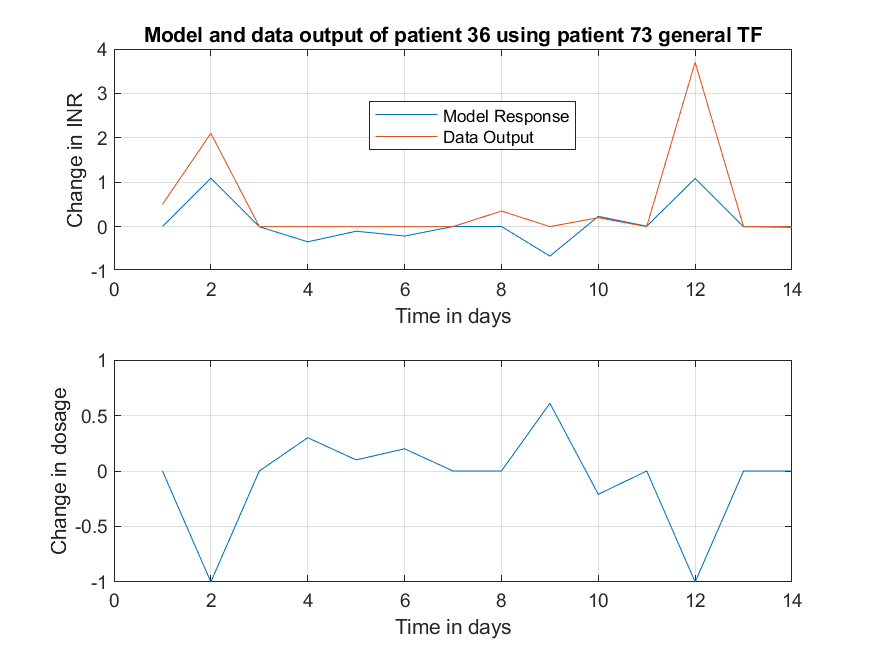


Figure - Shows patient 36 with a rt2 of 0.5731 (LEFT) and patient 37 with a rt2 of 0.5351 (RIGHT)

Patient 36 (left) has a larger rt2 value, mean and varience of error compared to patient 37 (right). Patient 37’s model response would be considered better as the model response and data points are closer together and therefore has a smaller error.

### 3.5.3 – Comparing Model Output and Data Output for PG216 Model

A close up of a map

Description automatically generated

Figure - Shows patient 61 with a rt2 of -2.6574 (LEFT) and patient 124 with a rt2 of 0.2801 (RIGHT)

A close up of a map

Description automatically generatedA close up of a map

Description automatically generated

Figure - Shows patient 291 with a rt2 of 0.8103 (LEFT) and patient 209 with a rt2 of 0.9151 (RIGHT)

Figures 22 and 23 show responses from four patients using patient 216 as the general model. Figure 23 shows very good responses with rt2s much greater than 0.5, whereas figure 22 (LEFT) shows patient 61 with a less than adequate response and a rt2 value less than 0. Patient 124 has a reasonable response, the rt2 is 0.2801 which is acceptable.

Table - Shows 21 patients with their rt2, mean error and variance of the error using the TF from patient 216

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| PatientNumber | RT2 | MeanError | VarError | Over0x2 | Over0x5 |
| 180 | -0.13843 | 0.139592 | 0.04585 | 148 | 54 |
| 181 | -0.33848 | 0.109763 | 0.036602 | 0 | 0 |
| 182 | 0.680378 | 0.167012 | 0.073481 | 0 | 0 |
| 183 | 0.556156 | 0.870713 | 0.260575 | 0 | 0 |
| 184 | 0.595322 | 0.13792 | 0.056216 | 0 | 0 |
| 185 | 0.016554 | 0.283807 | 0.161938 | 0 | 0 |
| 186 | 0.658064 | 0.382685 | 0.080862 | 0 | 0 |
| 187 | -2.44123 | 0.283983 | 0.138496 | 0 | 0 |
| 188 | -4.71901 | 0.954426 | 3.815161 | 0 | 0 |
| 189 | 0.741874 | 0.132861 | 0.068389 | 0 | 0 |
| 190 | -0.28274 | 0.202876 | 0.081261 | 0 | 0 |
| 191 | 0.362996 | 0.201886 | 0.115684 | 0 | 0 |
| 192 | 0.674096 | 0.416311 | 0.200298 | 0 | 0 |
| 193 | 0.101957 | 0.229718 | 0.105841 | 0 | 0 |
| 194 | 0.075985 | 0.257822 | 0.234824 | 0 | 0 |
| 195 | -146.283 | 1.231228 | 10.12594 | 0 | 0 |
| 196 | 0.539263 | 0.606085 | 0.197787 | 0 | 0 |
| 197 | 0.452506 | 0.822945 | 0.528715 | 0 | 0 |
| 198 | 0.209771 | 0.31323 | 0.168336 | 0 | 0 |
| 199 | 0.380099 | 0.418528 | 0.213538 | 0 | 0 |
| 200 | 0.388842 | 0.592351 | 0.572138 | 0 | 0 |

There are still some patients with very low rt2 values such as patient 195 shown below.

A close up of a map

Description automatically generatedA close up of a map

Description automatically generated

Figure - Shows the full response of patient 195 (left) and the close up (right)

From this you can see that the response for patient 195 is very close up until day 11, calculating the rt2 and errors up until and including day 11 shows this is a better response than first indicated by the full data set.

Table - Shows the change in values depending on the days chosen

|  |  |  |  |
| --- | --- | --- | --- |
| PatientNumber | RT2 | MeanError | VarError |
| 195 – Days 1 to 14 | -146.283 | 1.231228 | 10.12594 |
| 195 – Days 1 to 11 | 0.2838 | 0.1161 | 0.0241 |

Table 7 shows a big change when only the first eleven days are used to calculate the error terms and makes this an acceptable response for the first eleven days.

## 3.6.0 - Dynamic Linear Regression Models

The method for this section differs slightly due to the models. Section 3.6.0 evaluates the four DLR models using rt2 calculations and comparing these to four different nvr values, 0, 0.1, 0.01 and 0.001. This shows how much the C variables change with time when allowed to. Equation 1 and 3 are compared to assess the improvement of the response by having two variables and how much c1 and c2 change over time in comparison to each other. A selection of patient responses is plotted for evaluation of the ability of the model response to follow the data trend. The rt2 and errors were calculated as discussed in section 3.1.1.

### 3.6.1 – Introduction

Dynamic Linear Regression models used in this section are very simple, with equation 1 and 2 having only 1 variable and equation 3 and 4 having 2 variables. NVR stands for noise variance ratio. **T**his variable determines whether the C variables can change with time or not and if they can, how much they can change. The larger the NVR, the more the C variables can change with time.

The parameters are determined by the dlr function in MATLAB, in section 8.6.0. [fit, fitse, par, parse] = dlr(y, z, TVP, nvr); this is used to get the C parameters, with “fit” being the model fit, “fitse” is the model errors, “par” and “parse” are the parameters and the parameter errors respectively. The input data is “x”, the output data is “y” and “nvr” is decided by the user for how much the parameters can change over time. The nvr was equal to either 0, 0.1, 0.01 or 0.001 when testing these models.

Four models tested using the DLR system:

(1)

(2)

(3)

(4)

### 3.6.2 – Equation 3 with varying NVR

Having a larger nvr meant a higher percentage of rt2 values were closer to 1, with equation 3 having the highest percentage. Figures 25 to 28 are all equation 3 with varying nvr.

A screenshot of a social media post

Description automatically generated

Figure - Shows number and percentage of patients with rt2 over 0.2 and 0.5. NVR = 0, no delay

A screenshot of a social media post

Description automatically generated

Figure - Shows number and percentage of patients with rt2 over 0.2 and 0.5. NVR = 0.001, no delay

A screenshot of a cell phone

Description automatically generated

Figure - Shows number and percentage of patients with rt2 over 0.2 and 0.5. NVR = 0.01, no delay

A screenshot of a cell phone

Description automatically generated

Figure - Shows number and percentage of patients with rt2 over 0.2 and 0.5. NVR = 0.1, no delay

### 3.6.3 – Equation 3 Testing c1 and c2 variables

For all of these, C1 starting point varies considerably depending on the patient, concentrated within the range of 0 to -2.5 as seen in figure 29.

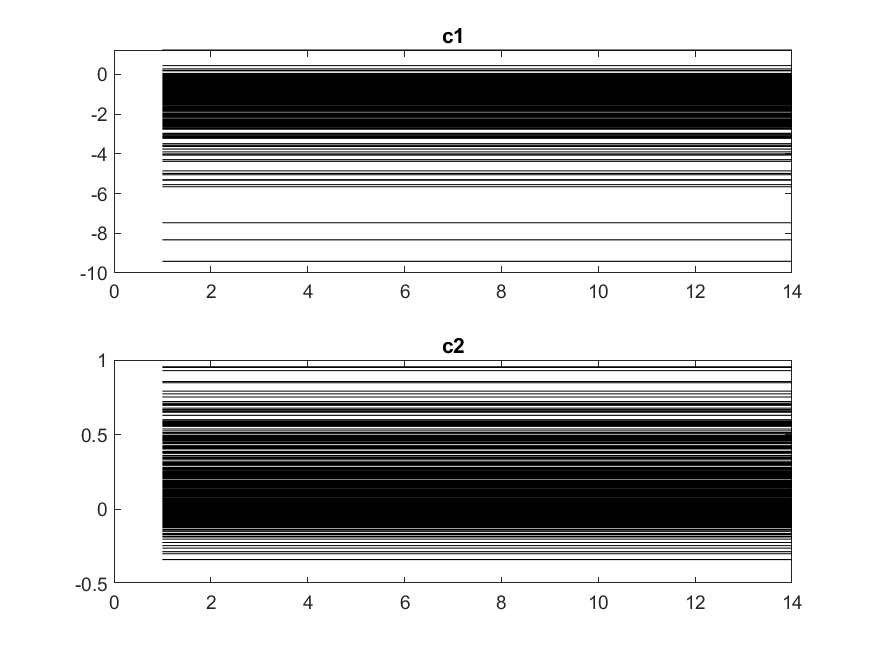


Figure - Shows all c1 and c2 values, nvr equal to 0

The c2 variable does not have a significant change in starting point, concentrated in the range -0.2 to 0.4. The c2 variable does not affect the response as significantly as c1.

### 3.6.4 – Comparing Equation 1 and 3

A screenshot of a cell phone

Description automatically generated

Figure - Results from equation 1, nvr = 0

As seen by comparing figure 25 and figure 30, the change in including c2 gives a slight improvement but not significantly. C1 has a varying start point across all the patients when nvr = 0.01, however, remains constant throughout the 14 samples for all the patients. C2 has a similar start point for all patients but varies for each patient throughout the 14 samples to differing degrees.

A screenshot of a computer

Description automatically generated

Figure - Equation 3, nvr = 0.01

Figure 31 illustrates the point of c1 staying constant, while c2 varies over time significantly in comparison.

A close up of a map

Description automatically generated

Figure - Shows Patient 44 variables with NVR equal to 0 and 0.1

Figure 32 shows that the C2 variable changes more when given the opportunity, compared to C1. The maximum change for C1 is approximately 0.15 whereas the maximum change for C2 is 0.7, which is a much bigger difference.

### 3.6.5 – Comparing Model Output and Data Output using Eq1

Focusing on equation 1 with nvr = 0, patients 109 and 273 were chosen as the best and worst response according to rt2 and patient 154 and 272 were chosen at random.

A close up of a map

Description automatically generatedA close up of a map

Description automatically generated

Figure - Shows patient 109 (LEFT) and 154 (RIGHT) using DLR Eq1

A close up of a map

Description automatically generatedA close up of a map

Description automatically generated

Figure - Shows patient 272 (LEFT) and 273 (RIGHT) using DLR Eq1

Table - Shows four patients with their rt2 values, errors and c1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PatientNumber | RT2 | MeanError | VarError | c1 |
| 109 | 0.978062 | 0.00592 | 0.00024 | -0.83659 |
| 154 | 0.253471 | 0.066432 | 0.024987 | -1.42839 |
| 272 | 0.017598 | 0.038803 | 0.009623 | -0.16215 |
| 273 | -0.1049 | 0.253074 | 0.105573 | -0.42067 |

Patient 109 clearly has the best response out of these four patients, as the rt2 is very close to one and the errors are very small, indicating the good response. Patient 273 has a good response from day 6 onwards as the dosage is changed even though it is the worst response using equation 1.

The variable c1 has a large range from -10.7340 to 1.1666 with a mean of -1.4269. The massive range makes this model hard to generalise for all the patients as c1 is bespoke for each patient’s data, therefore a generalised gain model is investigated further.

### 3.6.6 – Eq2 Testing

Equation 2 was researched by changing the input data to be shifted 1 day forward in time, therefore input data for day 3 was now the input data for day 4, with day 1 being set to 0. This was tested in the same way as equation 3.

A screenshot of a cell phone

Description automatically generated

Figure - Shows number and percentage of patients with rt2 over 0.2 and 0.5 using equation 2. NVR = 0

A screenshot of a cell phone

Description automatically generated

Figure - Shows number and percentage of patients with rt2 over 0.2 and 0.5 using equation 2. NVR = 0.01

A screenshot of a cell phone

Description automatically generated

Figure - Shows number and percentage of patients with rt2 over 0.2 and 0.5 using equation 2. NVR = 0.1

As seen from figures 35 to 37, the model is much less able to describe the patient’s data in the same way as equation 1 or 3, even the changing of NVR doesn’t affect the percentages a great deal. Equation 3’s base percentages were 74.9% and 38.0% when NVR was 0, compared with equation 2’s percentages which were 4.6% and 0%. This is 16 times difference and makes equation 2 one of the worst models in terms of rt2. With equation 3, increasing the NVR from 0 to 0.1, the over 0.2 percentage increased from approximately 75% to 97%, whereas for equation 2, it increased from 4.6% to 7.3%.

Using the DLR model, equation 1 and 3 have a large increase of the number of patients over 0.2 and 0.5 for rt2 values when compared to the first and second order models. Even the generalised patient specific model is less effective than this model, and the DLRs are much less complicated when compared to the models considered so far.

## 3.7.0 - Gain Model Research

Section 3.7.0 evaluates the Gain Model using rt2 calculations and plotted model responses. The input data from the patients is passed through the model and the output is plotted. The rt2 and errors were calculated as discussed in section 3.1.1.

### 3.7.1 – Introduction

Through trying various models, it was found that a high percentage of rt2 values over both 0.2 and 0.5 was just a simple gain model. This was researched further; the gain should be negative as the change in INR and change of dosage are negatively proportional.

### 3.7.2 – Gain Research

First a simple attempt at trialling a gain between -1 and 1 in steps of 0.01 for all patients was shown to be quite effective.

A screenshot of a cell phone

Description automatically generated

Figure - Shows the results from a gain between -1 and 1 in steps of 0.01

With a massive increase to 55.78% of Rt2 values over 0.2 and 16.50% over 0.5 Rt2, this shows a substantial difference between this model and either the first or second order model. Here it shows that the best gain for over 0.2 rt2 is -0.83 and for over 0.5 rt2 it is -1. This brief investigation gives an indication that a more detailed testing would be useful.

### 3.7.3 - Further Research into Gain Value

Next, a more thorough search into values between -4 to 0 in steps of 0.01 was completed. These values were chosen as this range covers the mean c1 variable from the DLR research.

Table - Shows the best gain models according to rt2 over 0.2 between -4 and 0

|  |  |  |
| --- | --- | --- |
| Gain | Over0x2 | Over0x5 |
| -0.86 | 169 | 44 |
| -0.84 | 169 | 43 |
| -0.83 | 169 | 42 |
| -1.14 | 169 | 53 |
| -1.08 | 169 | 51 |
| -1.07 | 169 | 49 |

Table 9 shows the best values of the gain model, all of which have 169 responses over 0.2 for rt2, which is 55.78% of the patients. The model gain of -1.14 seems to be the best as this model has the most patients with a rt2 over 0.5, therefore further testing was conducted. Testing between a multiplier of -1.2 and -1.1 in steps of 0.001 was carried out.

Table - Shows the best gain models according to rt2 over 0.2 between -1.1 and -1.2 in steps of 0.001

|  |  |  |  |
| --- | --- | --- | --- |
| Number | Gain | Over0x2 | Over0x5 |
| 1 | -1.143 | 169 | 53 |
| 2 | -1.142 | 169 | 53 |
| 3 | -1.141 | 169 | 53 |
| 4 | -1.140 | 169 | 53 |

Table 10 shows the best results found. There is no difference between these results so therefore the gain can be chosen as one of these four. The gain has been chosen as -1.141, as number 3 and 4 have one less patient rt2 under 0.

### 3.7.4 – Comparing Model Output and Data Output

Table - Selection of patients with their rt2 and errors

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| PatientNumber | rt2 | MeanError | VarienceError | Over0x2 | Over0x5 |
| 4 | 0.493474 | 0.280231 | 0.19372 | 169 | 53 |
| 69 | 0.508006 | 0.307786 | 0.258141 | 0 | 0 |
| 98 | 0.835596 | 0.057004 | 0.008756 | 0 | 0 |
| 139 | 0.132213 | 0.371429 | 0.246938 | 0 | 0 |
| 195 | -72.184 | 0.860237 | 5.031524 | 0 | 0 |
| 217 | 0.152405 | 0.091416 | 0.022296 | 0 | 0 |
| 265 | -0.08178 | 0.253634 | 0.142281 | 0 | 0 |
| 291 | 0.903587 | 0.095441 | 0.027022 | 0 | 0 |
| 302 | 0.334187 | 0.248344 | 0.24497 | 0 | 0 |

A close up of a map

Description automatically generatedA close up of a map

Description automatically generated

Figure - Patient 69 (LEFT) and patient 98 (RIGHT) responses using a gain model of -1.142

A close up of a map

Description automatically generatedA close up of a map

Description automatically generated

Figure - Patient 195 (LEFT) and patient 302 (RIGHT) responses using a gain model of -1.142

These graphs show the responses of four patients using the gain model, all of which show the model response follows the trend of the data output, including patient 195 until day 13. However, upon further inspection of the data, the change in dosage decreases suddenly from 2 to -7 and there is very little reaction in the data output from this input change. Looking at the patient data, there are no other extreme cases where this occurs.

Due to the simplicity of this model and how effective it is, the research indicates that this has the best overall response for all the patients.

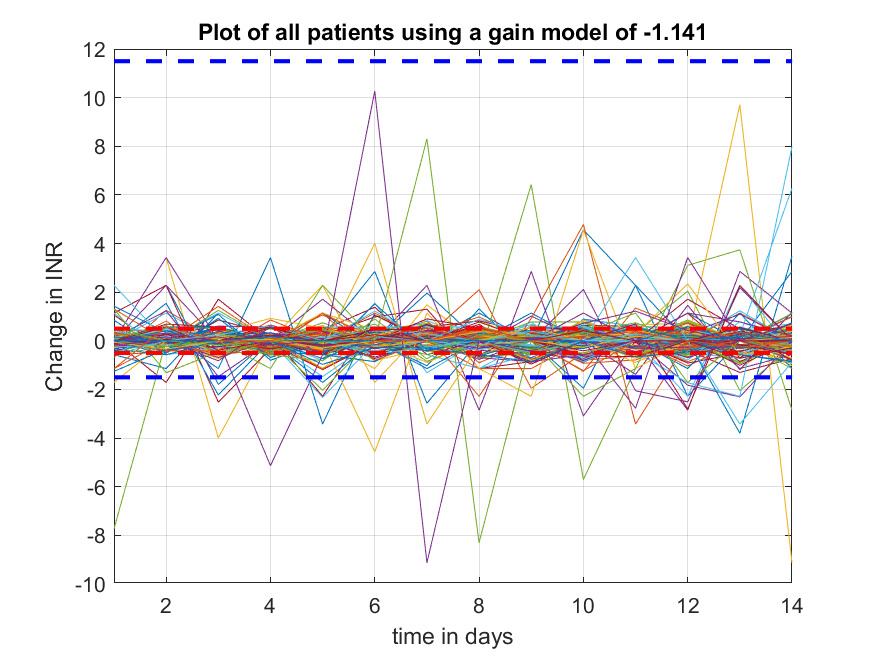


Figure - Output of all Patients using gain of -1.141

The red dotted lines show the upper and lower bounds of the desired INR (2 and 3). This shows that the majority of the outputs are within the bounds aimed for. The blue dotted lines show the upper and lower critical bounds, an INR equal to 9 or 1. The larger purple line, which significantly goes beyond the bounds to a very low INR, which could be dangerous for the patient.

## 3.8.0 - Model Conclusions

### 3.8.1 – First Order Model Conclusions

The model with the lowest percentage of rt2 values is the first order model, with only 0.66% over 0.2 and 0% over 0.5. This indicates the lack of ability to follow the trends of the patient’s data. This is verified by the figures 4 to 7. The best two responses of this model, according to rt2, follows the data trend more than the other plots of the first order model, but is still not acceptable. Also, the high mean errors of the responses, shown in table 1, indicates that this model is not a good representation of these patients’ data. However, due to this model being one of the simpler models, it would be easier to control, only needing two controller variables in PIP.

### 3.8.2 – Second Order Model Conclusions

The second order model has the joint second lowest percentage of rt2 values, 4.62% of responses over 0.2 and 0% over 0.5. This is an improvement on the first order model but is still not acceptable as figures 9 to 12 show that the responses do not model the data well. The model also has high mean errors as shown in table 2.

### 3.8.3 – Patient Specific Models Conclusions

These models have much higher percentages when compared to the first and second order models, with percentages of 68.3% and 42.2% over 0.2 and 0.5 respectively. This model set has the highest percentage over 0.5 compared to any model, apart from DLR Eq3 when the NVR is 0.1. However, looking at table 4, patient 12 has an extremely low rt2 of -1199459 and extremely high errors. Because the models are patient specific it means that the control systems would also have to be patient specific, adding to the complexity of the systems.

### 3.8.4 – Patient Generalised Models Conclusions

The two models are similar in performance, with PG73 edging out PG216 in rt2 performance over 0.2 whereas PG216 has a slightly better performance for over 0.5. PG73 is 56.1% and 16.5% and PG216 is 48.8% and 17.8%. PG73 is better than PG216 due to the larger percentage difference between the improvement of rt2 values over 0.2. The two models have similar ability to follow the patient data and mean errors. Controlling this model is much easier compared to the patient specific models as it is one model for all patients.

### 3.8.5 – DLR Models Conclusions

The DLR models Eq1 and Eq3 have the highest overall percentages of 74.6% and 74.9% for rt2 values over 0.2 respectively. These DLRs model the data very well as shown by figures 33 and 34. Eq1 and Eq3 are patient specific which creates the same problem as discussed in section 3.8.3, however Eq3 is more complicated than Eq1 as Eq3 has two variables whereas Eq1 has only 1 variable. The slight improvement of having two variables is not worth the extra complexity it creates and therefore Eq1 is better.

### 3.8.6 – Gain Model Conclusions

The gain model is a good balance between simplicity and ability to model the data. This model is the best performing in terms of rt2 for the generalised models. It is also the simplest model being just a gain.

### 3.8.7 – Overall Model Conclusions

Table - Shows the models researched, with percentages, positives and negatives

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Percentage over 0.2** | **Percentage over 0.5** | **Positives** | **Negatives** |
| Patient Specific | 68.3% | 42.2% | Good percentages | Bespoke models |
| Patient Generalised – 73 | 56.1% | 16.5% | Generalised model | Less effective model than patient specific |
| Patient Generalised – 216 | 48.8% | 17.8% | Generalised model | 7.3% less for over 0.2 than PG 73 |
| First Order | 0.66% | 0% | Simple first order model | The lowest performing model in terms of rt2 |
| Second Order | 4.62% | 0% | Improvement on FOM | Joint second lowest performing in terms of rt2 |
| DLR Eq 1 | 74.6% | 37.0% | Simple gain model | Gain depends on patient and previous data |
| DLR Eq 2 | 4.62% | 0% | - | Joint second lowest performing in terms of rt2 |
| DLR Eq 3 | 74.9% | 38.0% | Simple gain model and separate adder | More bespoke than DLR Eq1 |
| Gain | 55.8% | 17.5% | Simplest model and generalised for all patients | Not as effective as DLR Eq 1/3 |

Table 12 gives an overview of the models researched and tested. It can be concluded that the gain model is the best overall model for this data set. It is a generalised model so can be applied to all the patients quickly and is also the simplest model. The patient generalised 73 model is close, however, is not as simple as the gain model.

DLR Eq 3 is the best model in terms of percentages, however the model is not generalised and therefore could not be applied to new patients easily. The worst model is the first order model as it is the lowest performing model in terms of the rt2 values calculated.

The models tested in the controller section will be the gain model, DLR Eq 1, PG73 and the second order model. The top 3 models have been taken along with the second order model, due to the gain model and DLR Eq1 not being able to be calculated for the PIP, using pole placement and therefore another model is needed to be compared against PG73 that is calculated using pole placement, while gain and DLR Eq1 will only be a PI controller made using trial and error.

Gain Model: DLR Eq1:

(ii indicates the variable changes with each patient)

PG73: Second Order:

# 4.0.0 - PIP Controller

In chapter 4, PIP controllers will be tested and evaluated using four models, the gain model, the DLR Eq1 model, the PG73 model and the second order model. Three areas are investigated for the controllers: the poles of the system, the disturbance control of the system and the robustness of the system. Using PIP, the poles of the system can be easily chosen, and this affects the controller variables and therefore the speed of the response and the disturbance control. The robustness is tested by a Monte Carlo simulation of each variable in the model.

The method used for this investigation had three steps. For the Second Order Model and PG73 systems the PIP function was used and poles of 0, 0.1, 0.5 and -0.5 were chosen and plotted to evaluate the speed of the response using the variables for the controller found by the function PIP. These systems then had a disturbance of a step up of 1 (and a step up of 2 for PG73 systems) at sample time 10, to assess the ability of the systems at the different poles to control a disturbance. Finally a Monte Carlo Simulation was applied for a robustness check of the systems.

For the Gain Model and DLR Eq1 systems, a trial and error method were used to calculate the f0 and ki variables for a PI control system, and a disturbance and Monte Carlo simulation was conducted using the same method described above.

## 4.1.0 - PIP Introduction

PIP refers to Proportional Integral Plus, using pole placement to create a stable control system. The number and value of controller variables depend on the model being controlled, for example the first order system, using PIP would have a control system below.

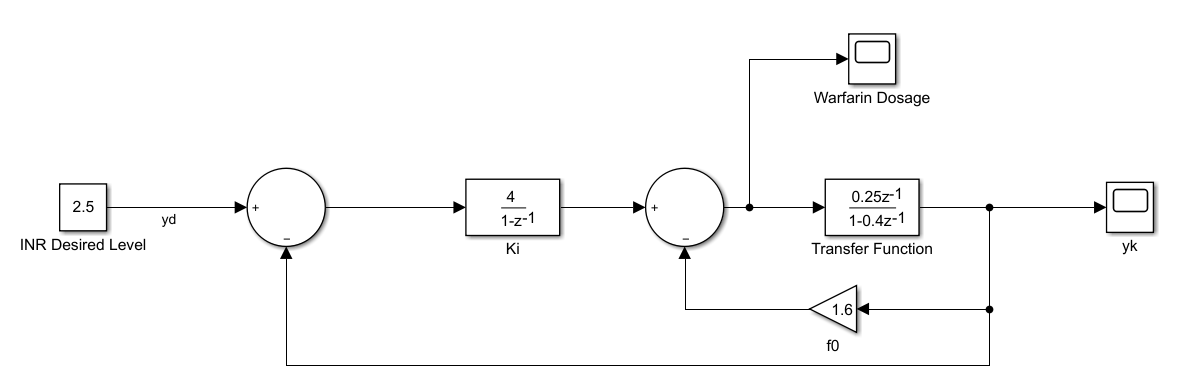


Figure - The control system for the first order model using PIP

As seen above, the first order system has a Ki variable, the integral variable and the f0 variable, the proportional variable. In more complicated systems, such as for the second order model, a second integral and proportional variable are used.

The structure of the function PIP is v = pip(at, bt, [poles]), at depends on the denominator of the transfer function and bt depends on the numerator of the transfer function. For the first order model, at = [-0.4] and bt = [0.25]. Notice the difference compared with the filter command, for the filter at = [1 -0.4] and bt = [0 0.25]. The poles are chosen to create a stable system, poles within the unit circle are stable, also the closer the poles are to the origin the faster the response, therefore the supposed ideal poles are 0.

For this system the system equation would be , where and . The denominator can be rewritten as For the system above, the poles have been chosen as 0, Z = 0, therefore , solving these two equations gives and . Changing the poles affects these variables, for instance poles of 0.1 gives and and can be found to be 1.56 and 3.24 respectively.

## 4.2.0 - Second Order Model

Second Order Model:

### 4.2.1 - PIP Controller Poles

The second order model is more complicated than the first order model. When using PIP four variables are produced, f0, f1, g1 and ki, these are arranged as below.

A screenshot of a cell phone

Description automatically generated

Figure - Schematic of second order models control system

There are now the extra g1 and f1 terms given by the extra numerator and denominator terms for the model compared to the first order model system.

The values for the controller terms when the poles are chosen as 0 are: f0 = 0.9514, f1 = 0.3942, g1 = 0.5435, ki = 2.4504 and these get passed through to the Simulink diagram above and the response is plotted.



Figure - Response of the second order system with poles of 0

The response hits the desired output at sample time 3 using these values, changing them does not cause the desired output to be reached a sample earlier. Therefore, this can be considered the best response in terms of speed.

Increasing the poles to 0.1 causes the controller variables to change to: f0 = 0.8813, f1 = 0.2377, g1 = 0.3284, ki = 1.6077.

A close up of text on a white background

Description automatically generated

Figure - Response of the second order system with poles of 0.1

The response becomes slower and the dosage change has a spike in day 2 and reduces to a similar starting dosage.

Taking the poles as 0.5 or -0.5 shows the over or underdamped responses of the system.

A close up of a map

Description automatically generated

Figure - Response of the second order system with poles of 0.5

This system does not reach the desired point until day 14, 11 days after the system with poles of 0.

A close up of a piece of paper

Description automatically generated

Figure - Response of the second order system with poles of -0.5

This system is very undesirable, it oscillates at the beginning for 14 samples which would be harmful to a patient, the dosage is also way too high and low at certain points.

### 4.2.2 - PIP Controller Disturbance with Varying Poles

Disturbances was added by including a step function block to the output before the feedback loop. This modelled a change in the INR that could be caused by various situations such as a change in diet. This allowed the testing of the time taken of the response to reach the desired output. The disturbance was set to step up to one at sample time 10. The speed of the response again depended on the value of the poles, the larger the ki variable, the quicker the response was to change the input to the desired output.

A close up of text on a white background

Description automatically generated

Figure - Shows the response of the SOM system with poles of 0 and disturbance at day 10

A close up of text on a white background

Description automatically generated

Figure - Shows the response of the SOM system with poles of 0.1 and disturbance at day 10

For poles of 0 and 0.1, the response is brought back to the desired limits (INR between 2 and 3) by day 12 and is kept within these boundaries. The system takes until day 16 to settle on the desired output of 2.5, 6 days after the step is introduced.

A close up of a map

Description automatically generated

Figure - Shows the response of the SOM system with poles of 0.5 and disturbance at day 10

By contrast, for poles of 0.5, the response has not yet reached the desired output of 2.5 when the disturbance occurs. Also the output is affected too greatly and is reduced to below an INR of 2 and does not get within the boundaries until day 17. There is a larger difference between the responses of poles 0 and 0.1 compared to poles of 0.5, and therefore the poles should be kept as small as possible.

### 4.2.3 - Monte Carlo Simulation on System

Checking the robustness of the system was done by a Monte Carlo simulation. This was done by testing the robustness of each coefficient of the model, therefore the second order model had five tests. These were plotted individually to show the response of the system due to these changed variables over 20 days. For the second order model, , and using the randn(1000,1) function gave a 1000 by 1 matrix of a standard normal distribution with a mean of 0, then by adding a coefficient from either at or bt to the matrix changed the mean to that coefficient, therefore giving a large testing base for each variable. After testing each coefficient using this method they are set back to the original values.

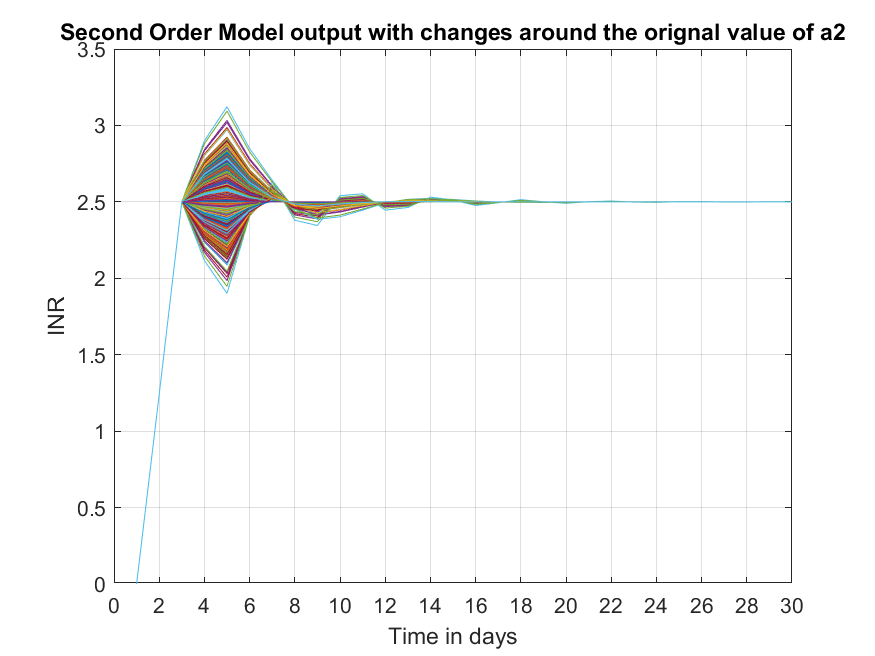
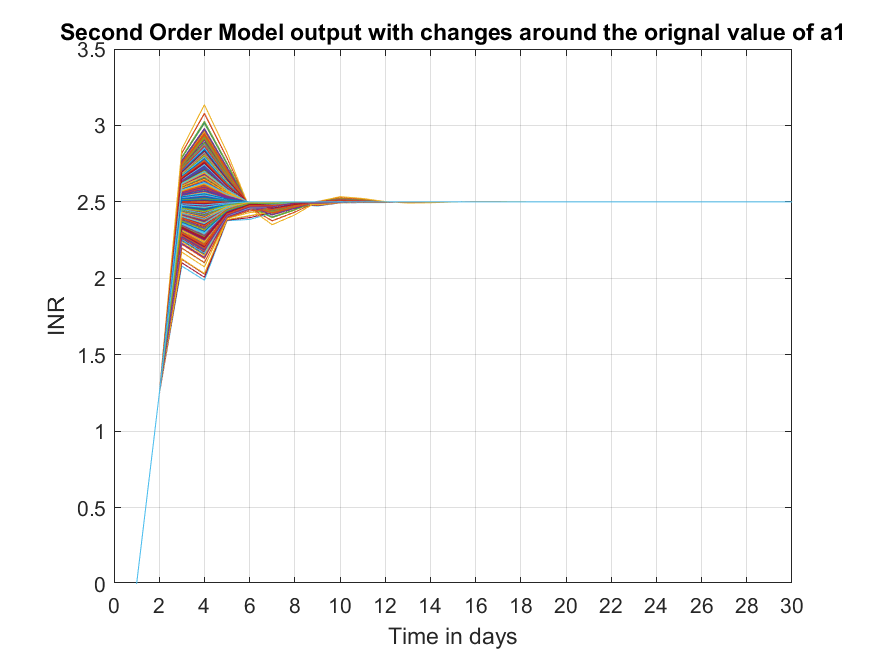


Figure - Shows the Monte Carlo simulation of a1 (LEFT) and a2 (RIGHT)

Figure 51 show the responses when the variables a1 and a2 are changed by small amounts individually. The responses of figure 51 show that a1 is robust, the responses become unfavourable for the extremes as a peak amplitude is outside of the desired range, but they do not oscillate out of control and all become stable and at the desired output by day 13. The extremes only just leave the boundaries of 2 and 3 and therefore is a good response. A2 is less robust as the extremes oscillate for a longer time period, again these only just leave the boundaries. All the outputs reach the desired output and are stable by day 22. These are both acceptably robust.

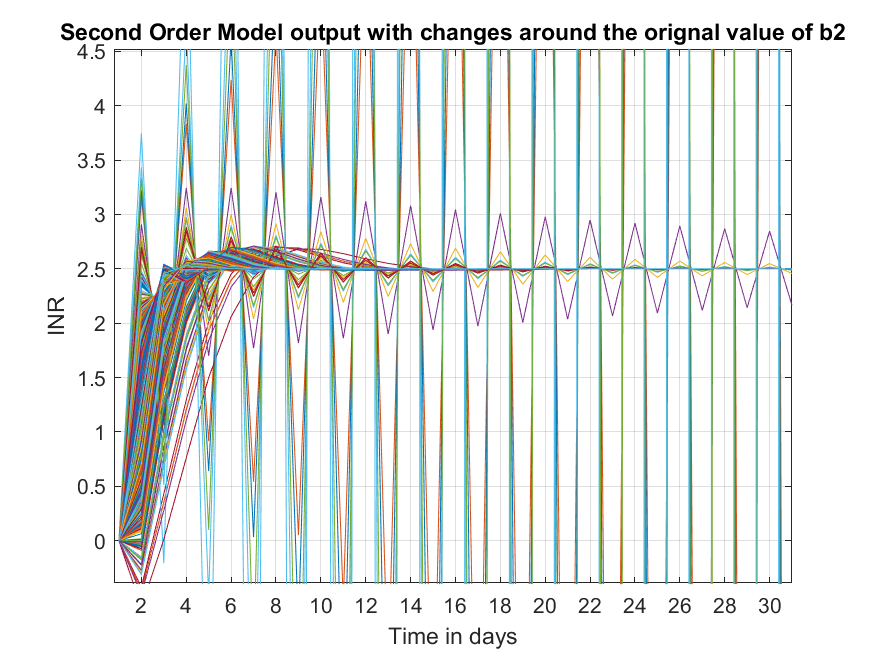
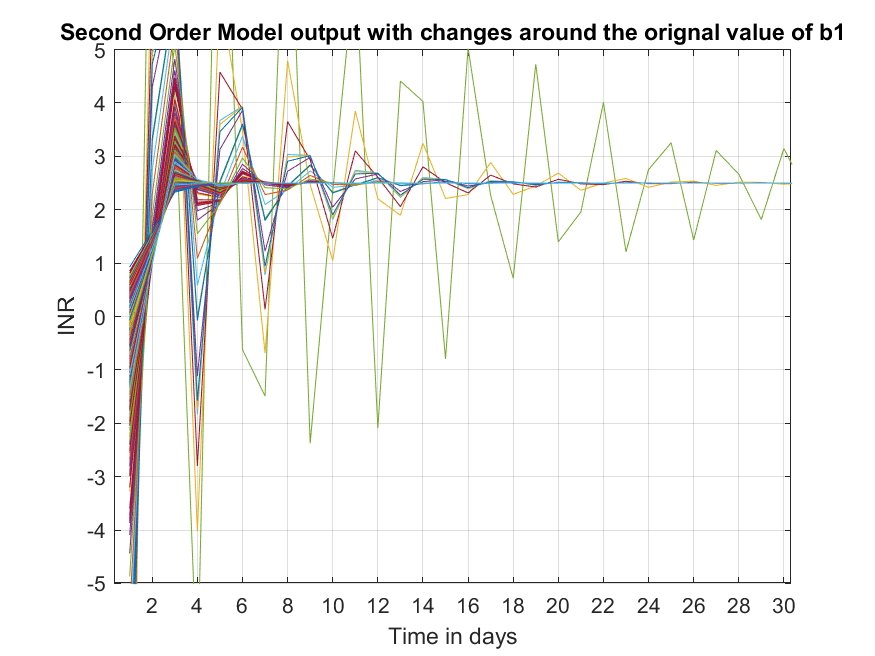


Figure - Shows the Monte Carlo simulation of b1 and b2

A screenshot of a cell phone

Description automatically generated

Figure - Shows the Monte Carlo simulation of b3

Figures 52 and 53 show the lack of robustness of the bt coefficients, these quickly oscillate out of control and are clearly not stable, therefore these values need to stay close to the original values.

## 4.3.0 - Patient Generalised – 73

PG73:

### 4.3.1 - PIP Controller Poles

This model has one less term in the denominator, therefore will not have the g1 variable, only f0, f1 and ki as seen in figure 54.

A picture containing clock

Description automatically generated

Figure – Schematic of the PG73 control system

Setting the poles to 0 gave the values for the control variables as f0 = 0.94566, f1 = -0.067233 and ki = -4.2079 and the response is shown below.

A close up of text on a white background

Description automatically generated

Figure - Response of the PG73 system with poles of 0

As seen from figure 55, the output is well within the desired range of 2 to 3 and specifically is within the bounds of 2.46 to 2.52. This is very close to the desired output. The speed of the response is excellent as it is already within the desired range.

Changing the poles to 0.1, 0.5 and -0.5 gives various different controller variable values and responses.

A close up of text on a white background

Description automatically generated

Figure - Response of the PG73 system with poles of 0.1

Poles of 0.1 gives the controller variables as f0 = 0.98867, f1 = -0.066072 and ki = -3.0675, and again has a very good response. It is very similar to figure 56 but slightly worse, with a maximum and minimum of 2.55 and 2.44 respectively. Figures 55 and 56 are both stable on the desired output by day 7, however figure 61 reaches this point by day 6.

A close up of text on a white background

Description automatically generated

Figure - Response of the PG73 system with poles of 0.5

The response given by poles of 0.5 is a highly undesirable response, by day 30 the INR has decreased to -300, with the dosage being nearly 300mg. Clearly in a real world set up, this would have been noticed and this kind of dosage would have never been given to the patient.

A close up of text on a white background

Description automatically generated

Figure - Response of the PG73 system with poles of -0.5

The output given by poles of -0.5 give a better response than poles of 0.5 but slightly worse than 0 and 0.1. The response stays within the desired bounds and is faster to the desired output than poles of 0.1, reaching this point by day 6. The controller values are f0 = 2.5303, f1 = -0.70733 and ki = -14.202.

### 4.3.2 - PIP Controller Disturbance with Varying Poles

For these disturbances the step up was set to 1 and a second test for a step up of 2, this showed how well the system controlled the disturbances. First poles of 0.

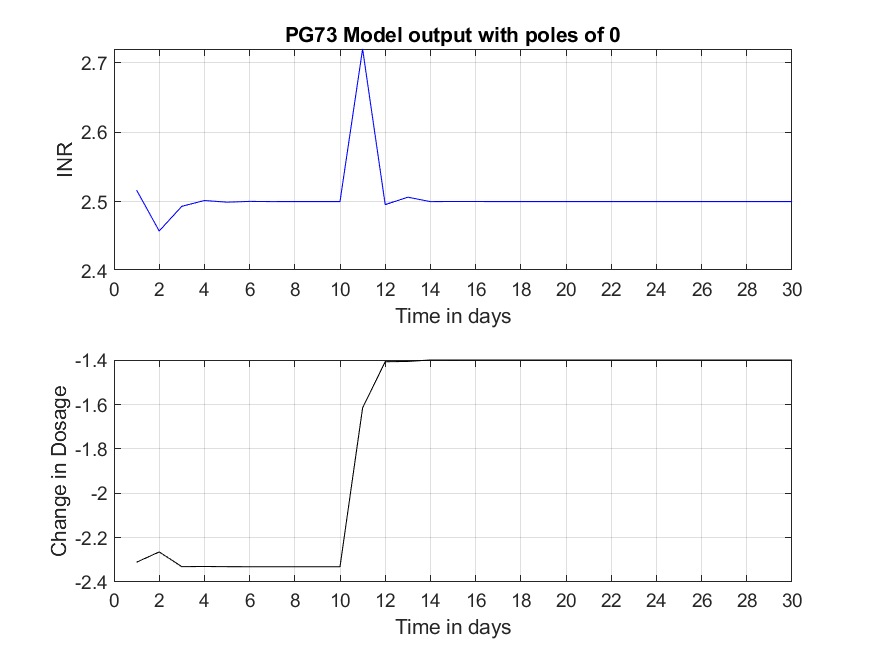
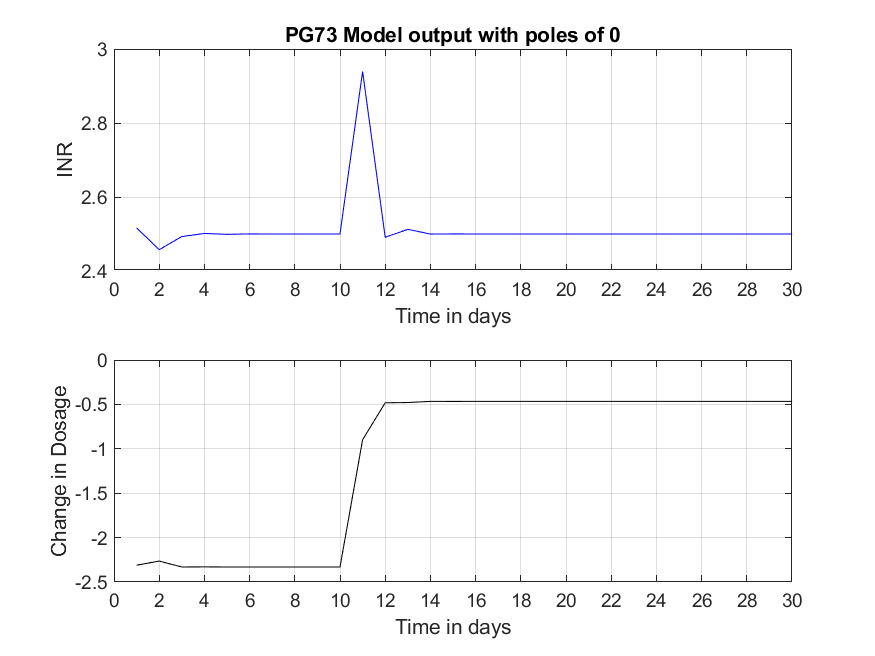
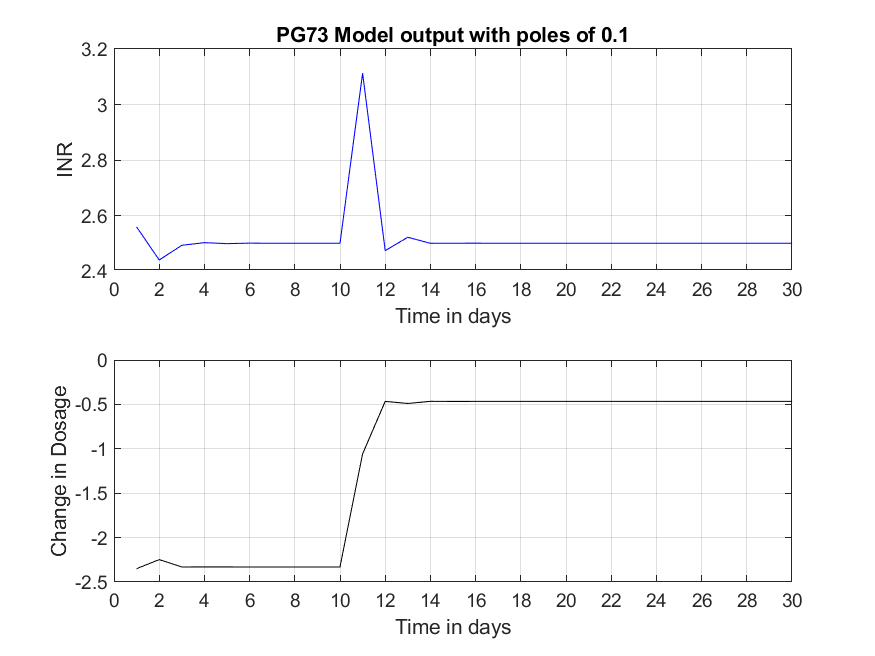
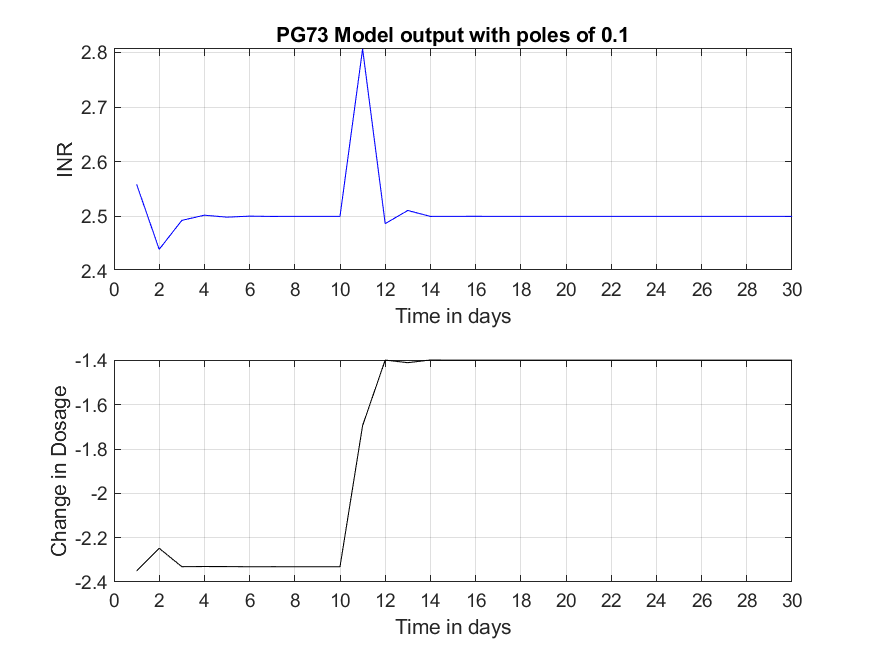


Figure 59 - Shows the response of the PG73 system with poles of 0 and disturbance at day 10. Step up of 1 (LEFT) and step up of 2 (RIGHT).

Both responses keep the INR within the desired limits, the large ki drives the response to the desired output more quickly than the smaller ki variables found in the second order system. For figure 59 both outputs are driven to the desired output by day 12, this is a very fast response.

Figure 60 is very similar to figure 59 but the poles are 0.1, the response for the step of 2 also goes slightly out of the desired limits to 3.1 but is still brought back to the desired output by day 12. For poles of 0.1, ki is -3.0675, this can be decreased to -5 and reduces the maximum amplitude of the peak at day 11 when the step is 2 as seen in figure 61.

Figure 60 - Shows the response of the PG73 system with poles of 0.1 and disturbance at day 10. Step up of 1 (LEFT) and step up of 2 (RIGHT).



A close up of a map

Description automatically generated

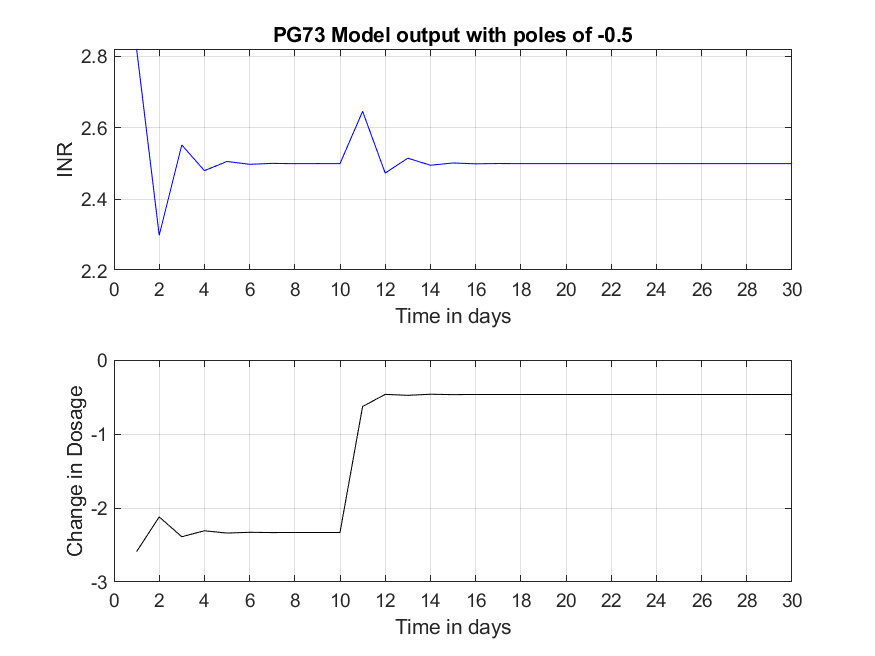
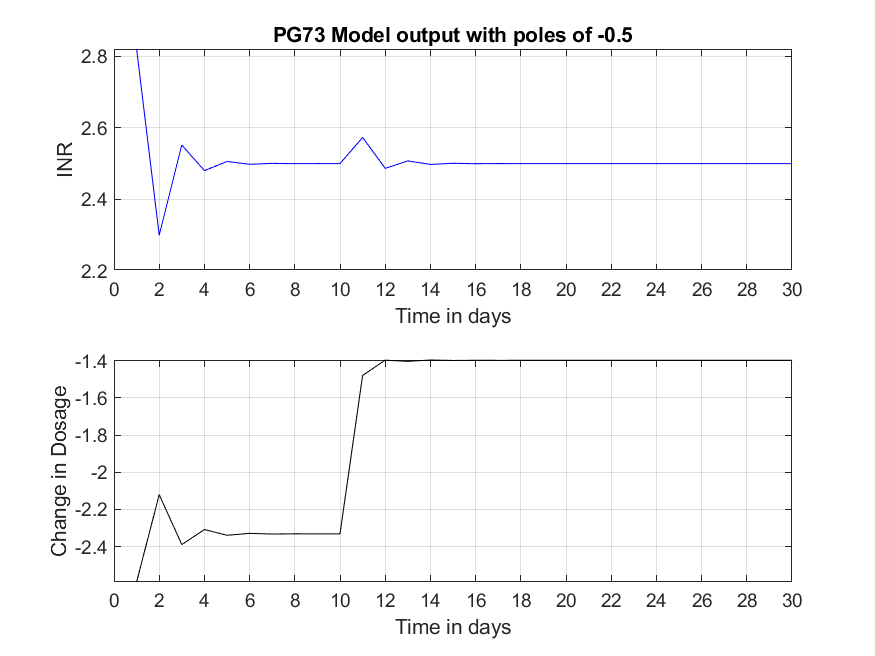
Figure - Shows the response of PG73 system with poles of 0.1, ki = -5 and a step-up disturbance of 2 at day 10.

The peak has now been reduced to approximately 2.87 and not changed the speed at which the response stabilizes at the desired output.

The next poles considered is -0.5 as this had a much better response compared to the system with poles of 0.5.

Here the ki variable is -14.202. This increases the speed at which the disturbance is reduced as seen in figure 62, as both responses are well within the desired bounds. This indicates that for this system the ki variable should be large and negative.

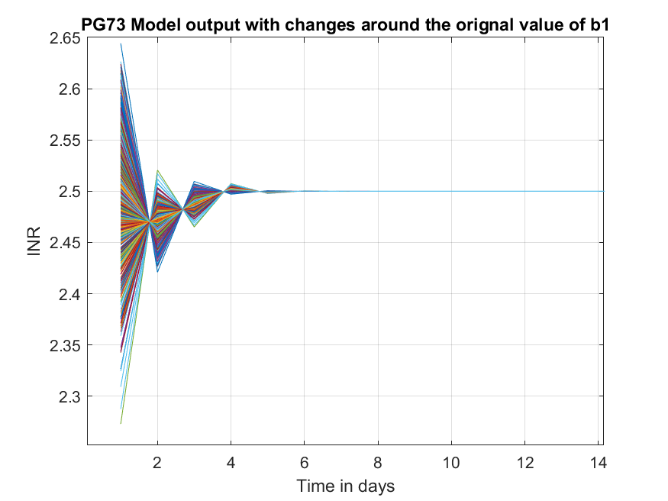
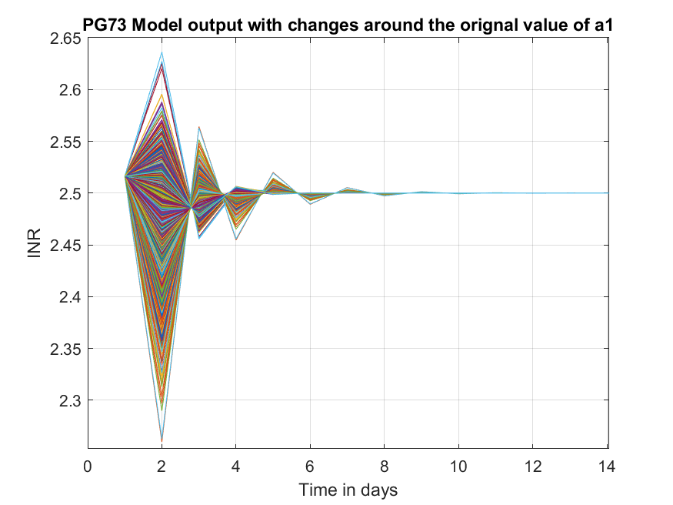
Figure 62 - Shows the response of the PG73 system with poles of -0.5 and disturbance at day 10. Step up of 1 (LEFT) and step up of 2 (RIGHT).



### 4.3.3 - Monte Carlo Simulation on the PG73 system

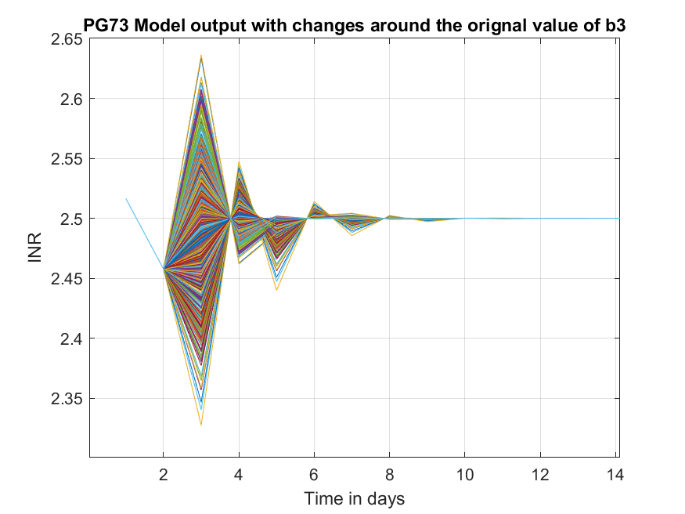
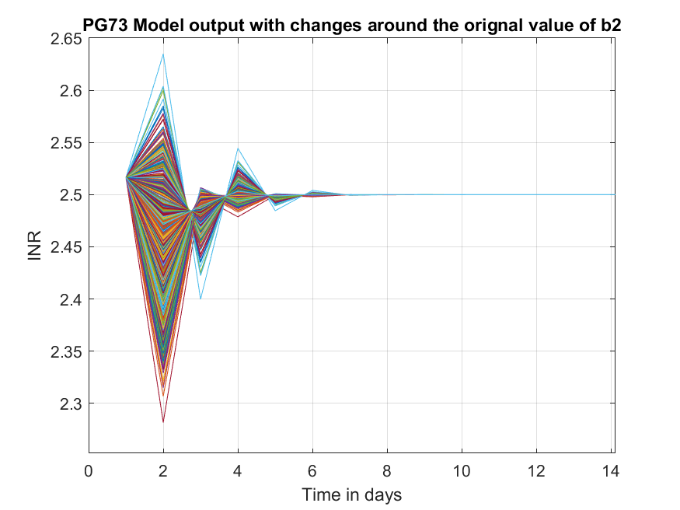
The Monte Carlo simulation was done on the two systems, poles of 0 and -0.5. This was done to see how robust the two control systems are as they are quite different from each other. The simulation was done in the same way as completed for the second order model. The maximum and minimum was found for each variable, for a1 the original value was 0.237 and the minimum and maximum was -0.862 and 0.5948, for b1 the original value was -1.089 and the minimum and maximum was -1.3962 and -0.7320. For b2 the original value was -0.2545 and the minimum and maximum was -0.6037 and 0.0921 and finally for b3 the original value was 0.01685 and the minimum and maximum was -0.2651 and 0.3435. Starting with poles of 0, the responses are below.

Figure 63 - Shows the Monte Carlo simulation of a1 (LEFT) and b1 (RIGHT).



Figures 63 and 64 show the responses from the Monte Carlo simulation. They show how robust this system is as in none of the simulations do the responses go above 2.65 or below 2.2. This is within the desired range. Neither do the responses oscillate out of control and become unstable.

Figure 64 - Shows the Monte Carlo simulation of b2 (LEFT) and b3 (RIGHT).



Moving onto the system with poles of -0.5. The maximum and minimum was found for each variable, for a1 the original value was 0.237 and the minimum and maximum was -0.1352 and 0.5696, for b1 the original value was -1.089 and the minimum and maximum was -1.3969 and -0.7980. For b2 the original value was -0.2545 and the minimum and maximum was -0.5648 and 0.0786 and finally for b3 the original value was 0.01685 and the minimum and maximum was -0.2960 and 0.3106.

Figure 65 - Shows the Monte Carlo simulation of a1 (LEFT) and b1 (RIGHT).

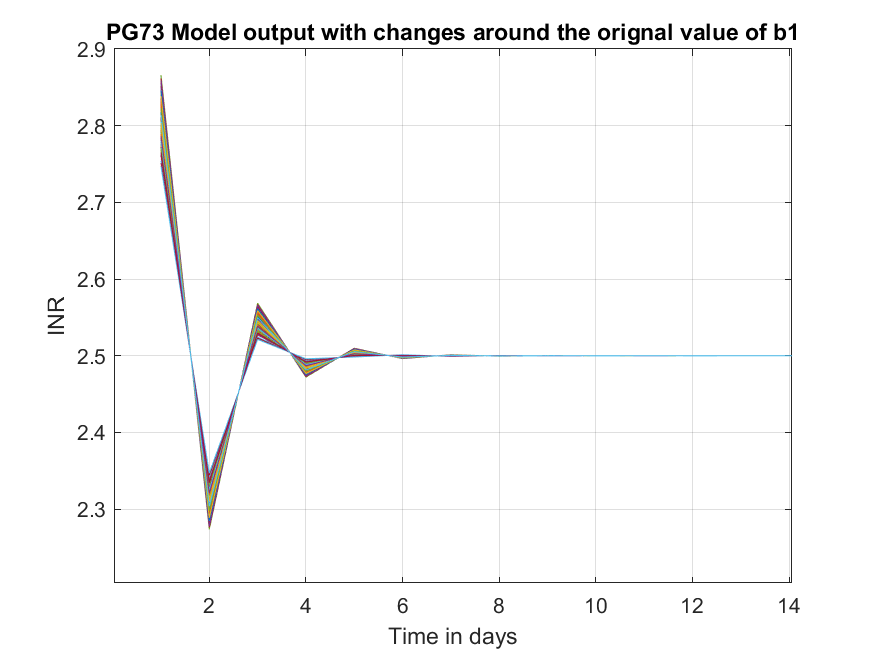
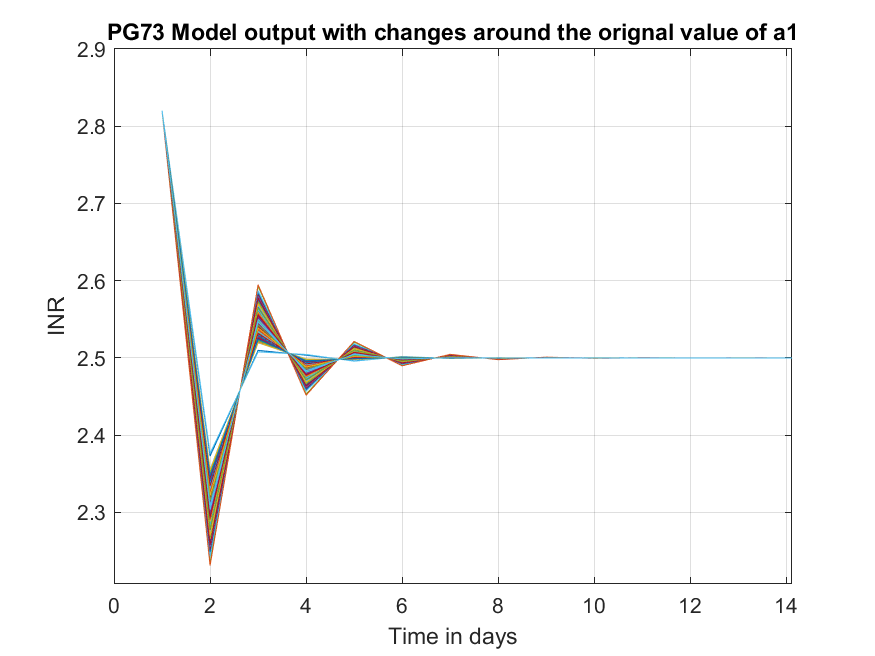
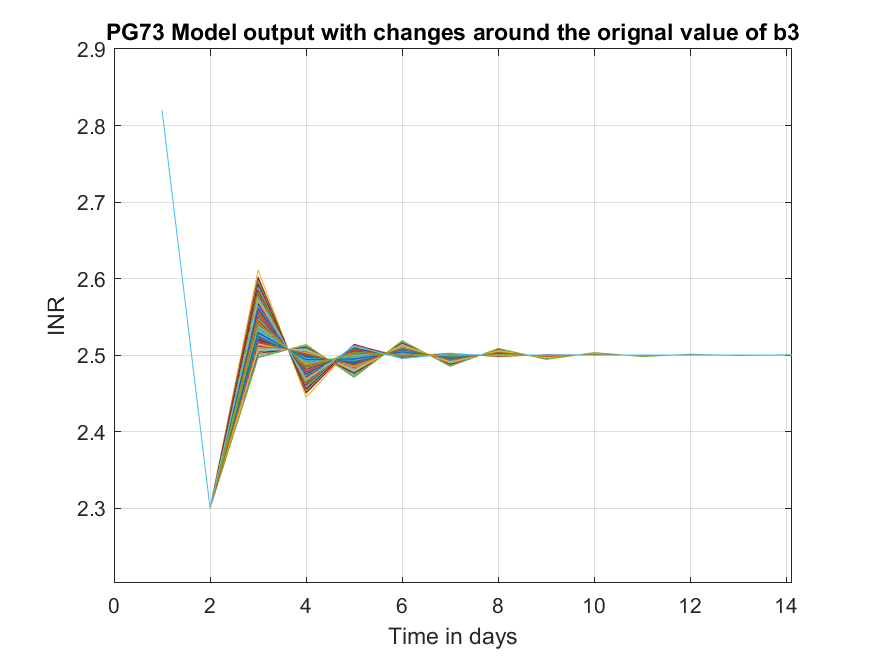
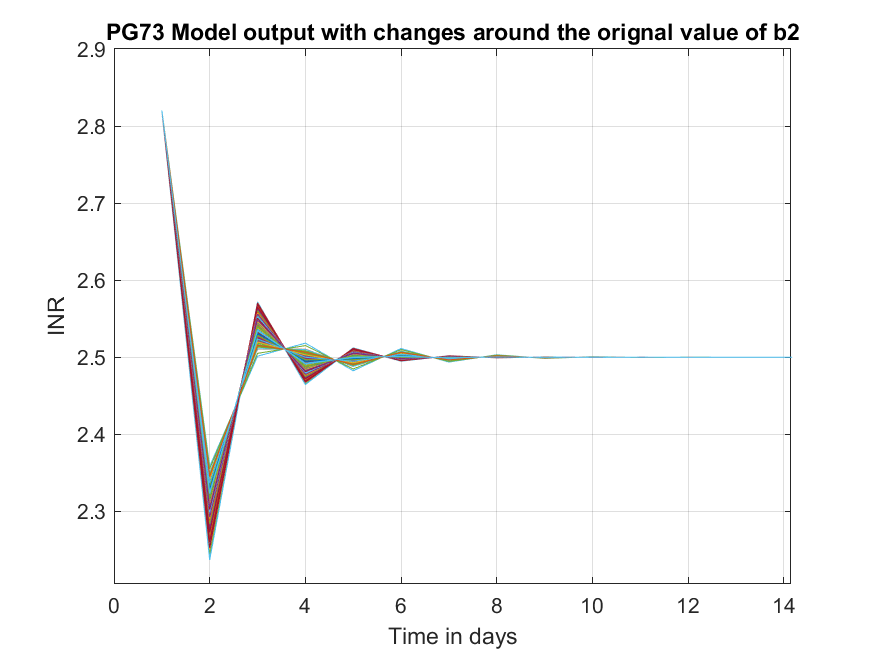


Figure 66 - Shows the Monte Carlo simulation of b2 (LEFT) and b3 (RIGHT).



As seen from figures 65 and 66 these are also very robust, as apart from day 1, the responses never go higher than 2.6 or lower than 2.2. The responses also are stable and do not oscillate out of control and all settle to 2.5 by day 12. This is slightly slower than the system with poles of 0 as these are all stable by day 10.

Both of these two systems are very robust, the coefficients don’t have to be constrained to the same extent as the second order system. However the systems are faster and have better responses when the coefficients are closer to the original values.

## 4.4.0 - Gain Model

Due to the way PIP function works, it only accepts transfer functions which neither the DLR Eq3 or gain model are, therefore, a trial and error method will be used to find the suitable f0 and ki values.

### 4.4.1 - Controller Variables

Looking at the starting values for the second order model and PG73, the f0s are close to 1 while the ki are vastly different, one being around -4 and the other being around 2.5. Therefore starting testing values for f0 were 0.8 to 1.0 in steps of 0.01 and ki between -5 and -3 in steps of 0.1.

Table - Shows the error in order along with the corresponding f0 and ki values

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Mean ABS error | f0 | ki | Mean ABS error | f0 | ki |
| 5.78E-05 | 0.88 | -5 | 8.5E-05 | 0.88 | -3.4 |
| 5.89E-05 | 0.88 | -4.9 | 8.76E-05 | 0.88 | -3.3 |
| 6.02E-05 | 0.88 | -4.8 | 9.03E-05 | 0.88 | -3.2 |
| 6.14E-05 | 0.88 | -4.7 | 9.32E-05 | 0.88 | -3.1 |
| 6.28E-05 | 0.88 | -4.6 | 9.64E-05 | 0.88 | -3 |
| 6.42E-05 | 0.88 | -4.5 | 0.000104 | 0.87 | -5 |
| 6.56E-05 | 0.88 | -4.4 | 0.000106 | 0.87 | -4.9 |
| 6.72E-05 | 0.88 | -4.3 | 0.000108 | 0.87 | -4.8 |
| 6.88E-05 | 0.88 | -4.2 | 0.00011 | 0.87 | -4.7 |
| 7.05E-05 | 0.88 | -4.1 | 0.000113 | 0.87 | -4.6 |
| 7.22E-05 | 0.88 | -4 | 0.000115 | 0.87 | -4.5 |
| 7.41E-05 | 0.88 | -3.9 | 0.000118 | 0.87 | -4.4 |
| 7.6E-05 | 0.88 | -3.8 | 0.00012 | 0.87 | -4.3 |
| 7.81E-05 | 0.88 | -3.7 | 0.000123 | 0.87 | -4.2 |
| 8.03E-05 | 0.88 | -3.6 | 0.000126 | 0.87 | -4.1 |
| 8.26E-05 | 0.88 | -3.5 | 0.00013 | 0.87 | -4 |

The error was calculated by taking 2.5 from the response and finding the mean of the absolute value and this was sorted in size order, the smallest error being when f0 is 0.88 and ki = -5.

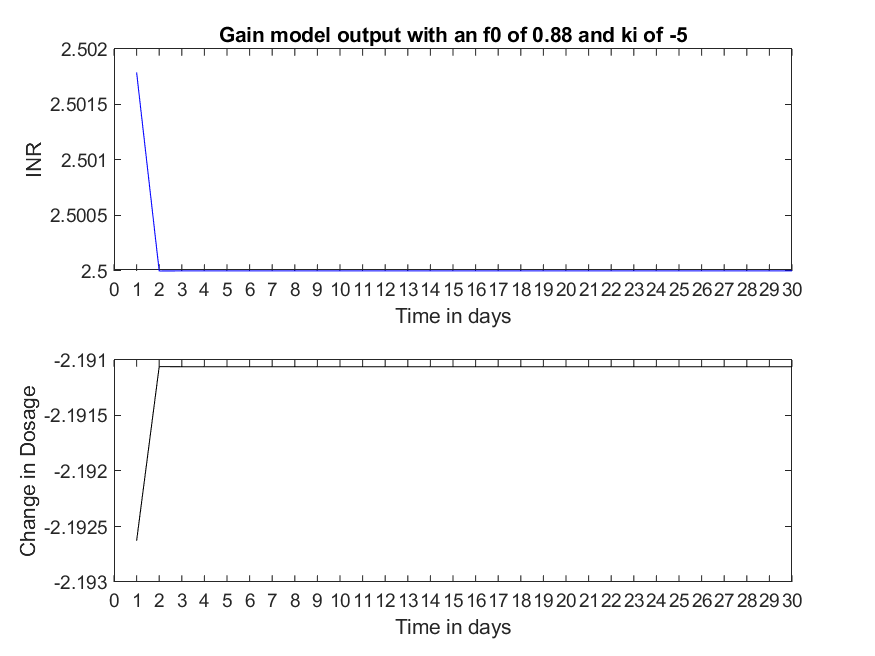


Figure - Shows the response of the gain model with f0 and ki of 0.88 and -5

This controlled response is very much within the desired range and these values will be taken for disturbances and the Monte Carlo simulation. There is no need for further testing for better variables as the change in dosage is already too precise for a real dosage as the minimum change in 1mg.

The worst response in terms of error calculated was when f0 was equal to 1 and ki equal to -3, this gave a mean error of 0.00362.

A picture containing screenshot

Description automatically generated

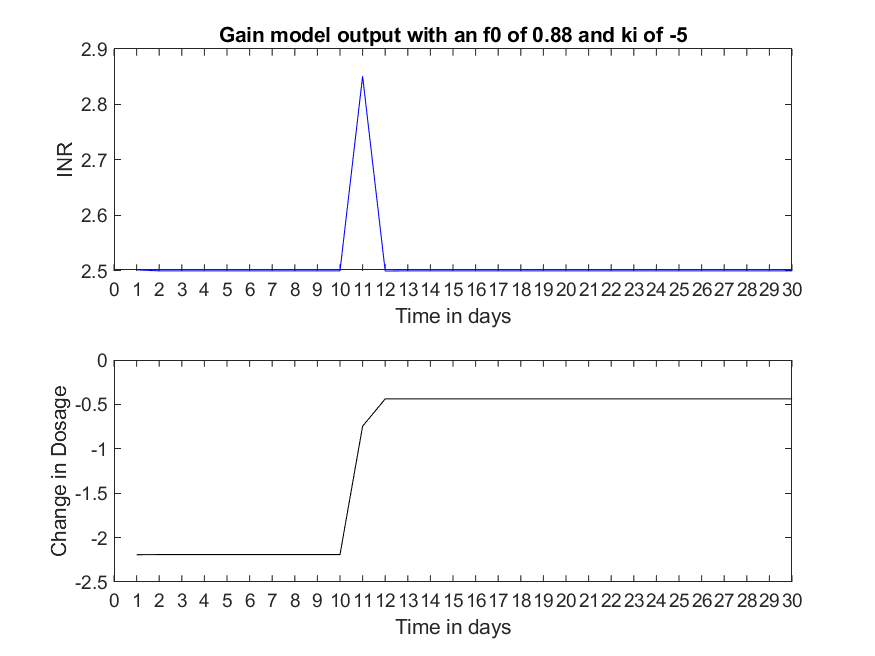
Figure - Shows the response of the gain model with f0 and ki of 1 and -3

The response is still within the desired range of 2 to 3, however it takes 3 days to settle to the desired output rather than 2 days as shown in figure 74.

### 4.4.2 - Controller Disturbances

This system controlled the disturbances better than the PG73 system with poles of 0.1 and had a similar response to the PG73 system with poles of -0.5 as seen below.

Figure 69 - Shows the response of the Gain Model system with f0 of 0.88 and ki of -5 and disturbance at day 10. Step up of 1 (LEFT) and step up of 2 (RIGHT).



The system corrects the step by day 12, 2 days after the step is introduced and the response does not leave the desired limits. The maximum amplitude of the responses are approximately 2.65 and 2.85, therefore showing how well this controller can counteract disturbances.

### 4.4.3 - Monte Carlo Simulation on the Gain System.

The same test was done on the gain system to find how robust it is.

A close up of a map

Description automatically generated

Figure - Shows the Monte Carlo simulation of the gain.

Figure 70 shows the response of the system when the gain is changed. The minimum gain is -1.7615 and the maximum gain is -0.4749. All the responses reach the desired output by day 4, all being within the desired limits. This shows the system is robust, and one of the faster systems.

## 4.5.0 - DLR Eq1

DLR Eq1: , (ii indicates the variable changes with each patient)

For ease of simulation, the nvr was set to 0 to find c1 for each patient.

### 4.5.1 - Controller Variables

For this model the c1 variable changes depending on the patient, therefore the f0 and ki variables could change with the patient increasing the complexity of controlling the system and thus having more or less robust systems and faster or slower responses for the same model depending on the patient.

Each patient model was tested for f0 and ki variables by sweeping f0 from 0.1 to 1 in steps of 0.1 and ki from -8 to -4 in steps of 0.25. The error for each combination of variable was found and the mean of this error was saved and sorted to find the smallest error and the f0 and ki values to create this response were saved for that specific patient. The mean value for the ki variable for all the patients is -7.78 indicating that most patients favoured a ki value of -8. Looking into the values saved, this holds true, with 274 patients ki equal to -8, 24 patients ki being -6 and even fewer as -7.75. The f0 variable was more erratic.

Table - Shows the number of patients with the corresponding f0 value

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| F0 | 0.0 | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1.0 |
| Number | 3 | 6 | 12 | 23 | 19 | 20 | 23 | 24 | 10 | 139 | 24 |

Here, the highest number of patients have an f0 0f 0.9 with 139 patients or 46% of patients, the next largest percentage is either an f0 of 1.0 or 0.7, 8% of patients. This indicates that a value for f0 in the region of 0.6 to 1.0 is best for a general controller.

A screenshot of a map

Description automatically generatedA picture containing screenshot

Description automatically generated

Figure - Shows patient 30 and 85 response using their individual controller variables

A picture containing screenshot

Description automatically generatedA picture containing screenshot

Description automatically generated

Figure - Shows patient 166 and 192 response using their individual controller variables

Four patients were chosen randomly using the randi function in MATLAB and their responses were plotted using their individual f0 and ki variables. Figures 71 and 72 show patients 30, 85, 166 and 192 which all have good responses. They are within the bounds of 2 and 3 as desired and reach the desired output by day 2, apart from patient 85’s response which takes an extra day to reach an INR of 2.5.

Using a global f0 and ki of 0.9 and -8 respectively gives very undesired responses from some of the patients.

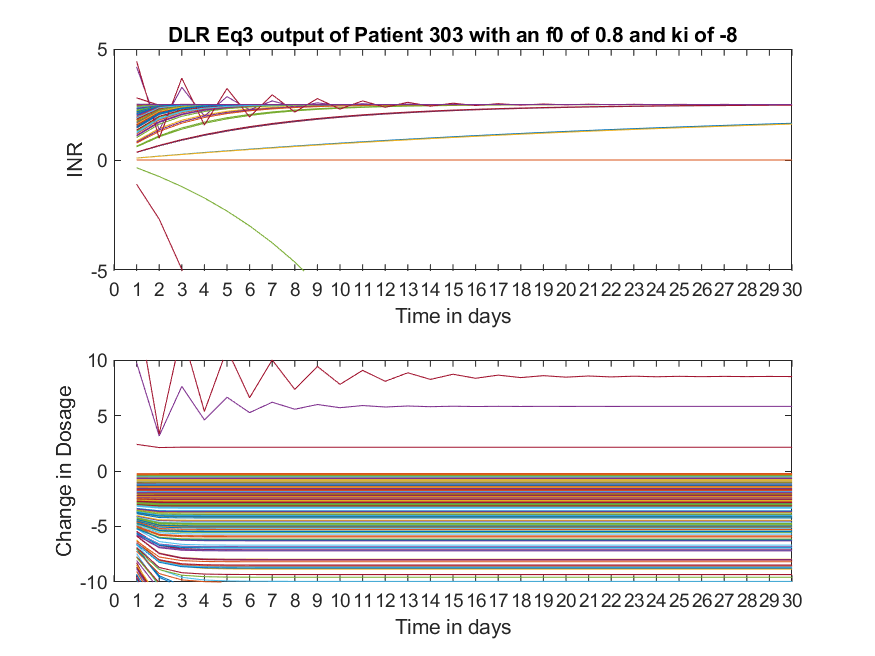


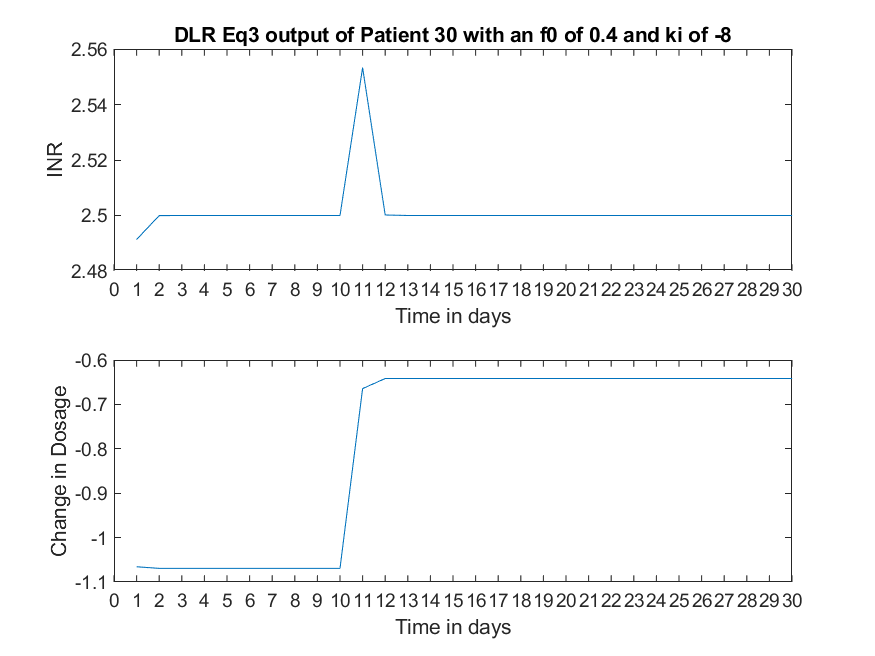
Figure - Shows the response of all patients using common f0 and ki

Using common f0 and ki clearly does not work for all the patients as seen in figure 73. Some of the patients’ output becomes unstable and exponentially increases out of control. This can be seen by the red and green lines in figure 73. This would be a serious issue in a clinical situation, however, the dosage would be monitored and stopped before going out of control.

Therefore individual control variables will be used for testing disturbances and the Monte Carlo simulation.

### 4.5.2 - Controller Disturbances

The same four patients will be looked at as before: patients 30, 85, 166 and 192 for ease of comparison. The disturbance is a step up from 0 to 1 at a time of 10 days.

A close up of a map

Description automatically generated

Figure - Shows the patient specific variables with a disturbance at time 10

A screenshot of a cell phone

Description automatically generatedA picture containing screenshot

Description automatically generated

Figure - Shows the patient specific variables with a disturbance at time 10

With these systems the controllers can get the disturbance reduced within two days of when it occurred. The amplitude of these disturbance does not go out of the limits either.

### 4.5.3 - Monte Carlo Simulation on the DLR Eq1.

A close up of text on a white background

Description automatically generatedA close up of text on a white background

Description automatically generated

Figure - Shows the Monte Carlo simulation of patient 30 and 85

A screenshot of a cell phone

Description automatically generatedA close up of text on a white background

Description automatically generated

Figure - Shows the Monte Carlo Simulation of patient 166 and 192

Table - Shows the original, minimum and maximum c1 value for each patient

|  |  |  |  |
| --- | --- | --- | --- |
| Patient Number | Original c1 | Max c1 | Min c1 |
| 30 | -2.3379 | -2.0441 | -2.6508 |
| 85 | -0.8659 | -0.5722 | -1.1788 |
| 166 | -4.1082 | -3.8145 | -4.4211 |
| 192 | -2.1998 | -1.9061 | -2.5127 |

Figures 76 and 77 show the robustness of these models. The highest INR is approximately 2.53 and the lowest is approximately 2.27. These values are more precise than would be recorded in a clinical situation. The responses are also fast as they reach and settle on the desired output by day 3. These responses can be considered robust.

## 4.6.0 - PIP Conclusions

### 4.6.1 – Introduction

This section will look at three areas of the controller for each model: the speed of the responses to reach the desired limit, how well the control system counteracts the added disturbances and finally how robust the model variables are by looking at the change of speed of the response and the ability to stay within the limits.

### 4.6.2 - Speed of the responses

For the best responses of each system, the Gain system is the fastest, reaching the desired output by the first day. The next best is the DLR Eq1 system for 3 of the patients which reach by day 2 but patient 85 takes 3 days. This is similar for the second order model system, which takes until day 3 to reach the output when the poles are 0, and finally PG73 is the slowest with poles of 0, the system reaches 2.5 at day 4.

However, increases the poles to 0.1 for the second order model system (SOMS) and the PG73 system, the SOMS speed of response doubles, taking until day 6, whereas PG73 hits the output by day 4, the same as when the poles were 0. This small change in the poles for the SOMS changes the speed of the response drastically and therefore is undesired. This also causes the response to become the slowest of the four systems.

The worst responses in terms of speed for the gain model and the DLR Eq1 reach the desired output by day 2 and day 3 respectively, these are quick responses and remain similar to the better responses of the models making them more desirable.

### 4.6.3 - Disturbance Control

Looking at the speed for the counter acting of the disturbances, all the systems take two days from when the step increase was added, day 10, apart from the SOMS, which takes until day 15 for poles of 0 and day 16 for poles of 0.1, 5 and 6 days respectively. These two systems are much slower compared to the other systems.

The amplitude for the disturbance for the PG73, Gain model and DLR Eq1 systems are much more controlled than the SOMS and are dealt with by either reducing the maximum amplitude, increasing the speed to reducing the response back to the desired output or both.

For a step increase of 1 at day 10, PG73, Gain model and DLR Eq1 all keep the maximum output less than 3, keeping the output within the desired boundaries, whereas the SOMS maximum amplitude increases to 3.5, outside of the desired bounds. The gain model and DLR Eq1 system manages to reduce this maximum amplitude greatly, with the maximum output being2.67 and 2.65 respectively, well within the desired boundaries.

The gain model and PG73 with poles of 0 can also handle a disturbance of 2, the maximum amplitude being 2.85 and 2.9 respectively, with the speed of the response being forced back to 2.5 not suffering and still taking 2 days. However when the poles of the PG73 system are increased to 0.1, the maximum amplitude increases to 3.1 for the step of 2, this is not as large as the SOMS but still undesired.

Table - Shows the best systems variables for each model system

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| System | SOMS | PG73 | Gain Model | DLR Eq1 |
| F0 | 0.8813 | 0.94566 | 0.88 | 0.9 |
| F1 | 0.2377 | -0.067233 | - | - |
| G1 | 0.3284 | - | - | - |
| ki | 1.6077 | -4.2079 | -5 | -8 |

The variables chosen for the DLR Eq1 system was the most common.

Using table 16, there is a correlation between the ability to control the disturbance and the size of the ki variable. The larger the size of the value, the better the system is to deal with the disturbance. DLR Eq1 has the largest ki variable and has the smallest change to the response’s amplitude.

### 4.6.4 - Robust Variables

The PG73, gain model and DLR Eq1 systems are much more robust compared to SOMS, for SOMS a1 and a2 are stable, the maximum and minimum amplitudes reached by the responses are 3.1 to 2 and 3.1 to 1.9 respectively, slightly going out of the desired limits. For a1 all the system responses settle at 2.5 by day 12 whereas the a2 responses don’t settle until day 22, these are much longer times compared to the three other systems. However, all the b variables for SOMS are unstable and oscillate out of control, this would be dangerous in a clinical use. SOMS is not a robust system.

PG73 is a robust system at both poles of 0 and -0.5, for poles of 0 the maximum and minimum for all the variables, a1, b1, b2 and b3 are very similar at 2.65 and 2.25, these are within the desired boundaries, however some variables take longer to settle on the desired output, such as a1 and b3 which both take until day 10, b1 and b2 settle at days 6 and 7 respectively. These are not as fast as the gain model or DLR Eq3 system but are still fast enough.

The gain model has a maximum amplitude of 2.7 and a minimum amplitude of 2.1 when testing using the Monte Carlo simulation. These max and min values are within the desired limits however they are further away from the desired output and the absolute change from 2.5 is greater compared to the PG73 system. Although the speed of the responses to reach the desired output is quicker, reaching this point at day 4 compared to day 6 for the quickest response of the PG73 variables. The gain model system is robust.

The DLR Eq1 system for individual patients is quicker still, patient 30, 166 and 192 all reach 2.5 by day 3, but patient 85 reaches 2.5 by day 4, still the same amount of time as the gain model system. The maximum and minimum amplitudes are also closer to the desired output, with the largest being 2.53 and 2.25, the error here being much smaller than the systems. However the generalised controller variables for the DLR Eq1 system is not robust, with some of the responses decreasing dramatically to values in the order of 10^24 or oscillating. These responses are very undesirable and therefore the system has been kept as a system for individual patients.

# 5.0.0 - Conclusions

The best system from the SOMS, PG73, Gain model and DLR Eq1 is the Gain model system. This is due to the ability of the system to model the previous patients effectively with percentages of 55.78% and 17.49% for rt2 values of over 0.2 and 0.5 respectively. The DLR Eq1 model has a greater percentage for the rt2 value at 74.9% and 37% but as the model is bespoke to each patient, this makes it much harder to create the models and previous data would be needed. Here the ease of the Gain model makes it much more desirable.

The worst model tested in the PIP section would be the SOM. This model had the joint second worst percentages out of the nine models tested, with percentages of 4.62% and 0%. As seen in figures 23 to 28, the model did not capture the trend of these six patients at all, even the best response according to the rt2 was far from the data output.

In this application controlling disturbances is especially important. The diet of the patient affects the effectiveness of warfarin, and therefore changes will occur no matter how careful the patient is. An increase or decrease in the intake of vitamin K or alcohol needs to be controlled effectively and quickly. The best SOMS takes over twice as long to reduce the disturbance as compared with the other three systems, making it the worst control system of the four.

The fastest response, best control of the disturbances and most robust variables is the DLR Eq1 system. However again this uses individual control variables for each patient, increasing the complexity for this system massively. Now patient data is needed to create the model and then testing is required to create the best control system for the specific patient. Therefore the best system is the Gain model system as this is a generalised system and will work effectively for new patients straight away without previous data being needed. This control system is also fast, controls disturbances well and is a robust system.

# 6.0.0 - Future Work

## 6.1.0 – Introduction

Future work gives the next steps recommended for furthering this project, split into two sections, testing and integration. Testing considers more investigation into the models and systems covered in this report, whereas integrations is about future ideas into using these systems in a clinical or home use situation.

## 6.2.0 – Testing

Testing these models using more patient data would be certainly be useful, this would give a larger data set and so the conclusions made would be more reliable. Also, some extreme cases, such as large changes to the INR or dosage change, would be helpful to see how the models reacted to certain conditions. This would benefit the project as the robustness could be tested more thoroughly.

In this project, DLRs were briefly touched upon, and a more in-depth investigation of these types of models would be beneficial.

More controllers, such as MPC, are needed to be tested for these models, only one controller was used in this project, which was shown to be effective in this application. However, other controllers may be more efficient in certain areas discussed in the report.

## 6.3.0 – Integration

The main ways the system could be implemented into the medical sector are as a home use testing kit for users of warfarin, overseen by the patient’s medical professional, or as an aid for the medical professional to give the correct dosage to the patient.

Medical professionals could find the control system helpful for themselves, using it in a way to aid their decisions, such as using it as a comparison tool to their own change in dosage recommendation or to reduce the calculations they have to make for dosage.

There is already an idea for self-management of warfarin as discussed in section 1.2.6. Combining this system with a commercial device such as Coaguchek would likely increase the number of patients willing to trial SMW. Self-management is well in practice for people with diabetes, the diabetics monitor their own blood sugar levels and act accordingly to the data they receive, a similar outcome could be achieved using the control system and Coaguchek.

# 7.0.0 – Critical Reflection

I felt that overall, the project went well. I had plans for how to investigate the models and the control systems, which is clearly demonstrated by the systematic layout of section 4.0.0 and the distinct parts of each system investigation.

This was a simulation-based project and so all the timing was in my own hands, I felt that I used the time effectively, the start of the project was slower due to me having to learn MATLAB and so more content was completed in the latter half of this project as I could write the scripts more quickly.

I had close to biweekly meetings with my supervisor, some of which I recorded, with permission, I felt this improved the efficiency of these meetings as I was able to refer to the recordings for any information I required.

The first set back was about the models, I planned to use 3 or 4, these were a patient specific one using RIVID, the first order model (from a previous paper) and the second order model (from a previous paper) and maybe one more. However I found that these models did not give a response that was acceptably close enough to the data set I had received and I had to spend more time investigating models that approximated the data set better, I ended up fully investigating about 9 models, 3 times what I had planned.

The next issue was the controllers, the plan decided for the project was to include 2 or 3 controllers and compare these systems against each other. I took the decision to drop the MPC I had been developing because of two main reasons, first, there was a large increase in time being shifted to the research of the models, and so less time was available for the controllers. And secondly of the four models chosen for the controller research, only one worked using the MPC function in MATLAB, as it did not allow direct feedthrough from the manipulated variable to the output, all the models apart from the first order and second order model required this. The models needed to have a time delay for the MPC to work and as seen between DLR Eq 1 and 2, there is a big difference to the models when including a time delay. I had a meeting with my supervisor, James Taylor, about this issue and he could not find a model that approximated the PG73 model with a time delay.

This shifted the focus of the project from testing and comparing several controllers, to investigating and testing the models themselves. Due to the detail required this resulted in an in-depth piece of work that I felt satisfied the brief.

# 8.0.0 – Appendices

## 8.1.0 – Appendix Contents

8.2.0 – Patient Data Code, used in section 2.0.0

8.3.0 – Patient Specific Models research, used in section 3.4.0

8.4.0 – First Order, Second order and Patient Generalised Models research, using in sections 3.2.0, 3.3.0 and 3.5.0

8.5.0 – Patient Generalised Testing models, used in section 3.5.0

8.6.0 – DLR Model Testing, used in section 3.6.0

8.7.0 – Gain Model Research, used in sections 3.7.0

8.8.0 – Second Order Model Controller, Poles, disturbances and Monte Carlo Simulations, used in section 4.2.0

8.9.0 – PG73 Controller, Poles, disturbances and Monte Carlo Simulations, used in section 4.3.0

8.10.0 – Gain Model Controller, variables testing and Monte Carlo Simulation, used in section 4.4.0

8.12.0 – DLR Eq1 Controller, variable testing and Monte Carlo Simulation, used in section 4.5.0

8.12.0 – Proposed timeline for the project and Actual timeline.

## 8.2.0 – Patients Data

% Plots all patient data 2

% 03/02/2020

close all

close all

% load outputs i.e. standardised INR values at clinic visits

yy=load('OutputData.dat');

% load inputs i.e. change of dose in mg

uu=load('inputs.dat');

% Figure 1 plots the output data as lines

figure(1)

subplot(211)

plot(yy)

title('Patient data output')

xlim([1 14])

xlabel('Time in days');

ylabel('Change in Log(INR)');

hold on

subplot(212)

plot(uu)

title('Patient data input')

xlim([1 14])

xlabel('Time in days');

ylabel('Change in dosage of Warfarin');

hold off

t = [0:1:15]';

% Lines with significant INR values

inr3 = (log(3))\*ones(16,1); % INR = 3

inr2 = (log(2))\*ones(16,1); % INR = 2

inr9 = (log(9))\*ones(16,1); % INR = 9

inr1 = (log(1))\*ones(16,1); % INR = 1

% Plots the Output data as dots with INR bounds

figure(2)

subplot(211)

hold on

plot(t, inr3, 'r--', 'Linewidth', 1.5)

plot(t, inr2, 'r--', 'Linewidth', 1.5)

plot(t, inr9, 'b--', 'linewidth', 1.5)

plot(t, inr1, 'b--', 'linewidth', 1.5)

%legend({'INR safe bound', '', 'INR Critical band'})

title('Patient data output')

xlim([0 15])

xticks([0:1:15])

xlabel('Time in days');

ylabel('Log(INR)');

grid on

subplot(212)

hold on

title('Patient data input')

xlim([1 14])

xlabel('Time in days');

ylabel('Change in dosage of Warfarin');

grid on

for ii = 1:1:303

subplot(211)

plot(yy(ii,:), 'o')

subplot(212)

plot(uu(ii,:))

end

hold off

hold off

## 8.3.0 – Patient Specific Model

%-----------------------------------------

%

% Patient Specific Model

%

%-----------------------------------------

% Warfarin data and Transfer Function model

% This script will use patient ii input and output

% data and create a best fitting TF using the function

% RIVID or RIV and plot this model output.

% James Taylor

% 14/10/2019

% Edited by George Caddick

% 31/10/19

close all

close all

% load outputs i.e. standardised INR values at clinic visits

yy=load('outputs.dat');

% load inputs i.e. change of dose in mg

uu=load('inputs.dat');

% select a patient

ii=1;

% Plot of the data output

figure(1)

subplot(211)

% Plots the iith patient output data with a black line

plot(yy(ii, :), 'k')

% Limits the axis

xlim([1 14])

% Plot of the input

subplot(212)

% Plots the iith patient input data with a black line

plot(uu(ii, :), 'k')

xlim([1 14])

xlabel('Time in days') %time unit of days

ylabel('Change of Dosage of Warfarin (mg)')

title('Data input')

z=[yy(ii, :)' uu(ii, :)']; % iith row of the data

if 1 % change this value if one specific fuction to be tested

% sc value of 2 puts the table in order of rt2

% changes which column to put highest to lowest

sc=2;

% scans through all the TF and puts the in order that relates to sc

% [at, bt, time delay, noise]

[th, stats, e]=rivid(z, [1 1 0 0; 3 3 5 0], sc);

else

% checks one specific fuction specified by the user

[th, stats, e]=riv(z, [3 3 5 0]);

end

% calls the values calculated in riv/rivid

rt2=stats(3)

[at, bt]=getpar(th)

% save model at bt

% Self calculating Rt2

my\_rt2=1-(var(e))/var(z(:, 1))

% Plots the simulated output and data output

subplot(211)

hold on

% plots patients model output

plot(z(:, 1)-e, 'b--', 'linewidth', 2)

hold on

% Title and axis labels

title("Plot of model output vs data output with patient " + (ii))

xlabel('Time in days') %time unit of days

ylabel('INR')

legend({'data', 'model'}, 'Location', 'Northeast')

## 8.4.0 – Patient Generalised Models, First Order Model and Second Order Model

% Evaluates all patients using stats model from a paper called: Model

% Predictive and Proportional Integral Control of Blood Clotting Speed

% Using Warfarin when Data are missing. Or Robust and

% Adaptive Anticoagulant Control. Or a model made for a patient depending

% on the users choice.

% Calculates the RT2 value all patients output Compared to the actual

% data and produces a table of this results, gives the number and

% percentage of patients with a rt2 of over 0.2 over 0.5.

% 20/01/2020 - 21/01/2020

% close all

% close all

% clear all

% load outputs i.e. standardised INR values at clinic visits

yy=load('outputs.dat');

% load inputs i.e. change of dose in mg

uu=load('inputs.dat');

choice = 2;

gt\_2 = 0;

gt\_5 = 0;

% Choice between first order model, second order model and patient made

% model

if choice == 1

at = [1 -0.4];

bt = [0 0.25];

elseif choice == 2

at = [1 -0.2324 -0.1491];

bt = [0 0.2025 0.2056];

elseif choice == 3

ii = 4;

% Plot of the data output

figure(1)

subplot(211)

% Plots the iith patient output data with a black line

plot(yy(ii, :), 'k')

% Limits the axis

xlim([1 14])

hold on

% Plot of the input

subplot(212)

% Plots the iith patient input data with a black line

plot(uu(ii, :), 'k')

xlim([1 14])

xlabel('Time in days') %time unit of days

ylabel('Change of Dosage of Warfarin (mg)')

title('Data input')

sc = 2;

% Finds the best model for patient ii

z = [yy(ii, :)' uu(ii, :)'];

[th, stats, e] = rivid(z, [1 1 0 0; 3 3 5 0], sc);

% Retrieves values calculated in RIVID

rt2 = stats(3);

[at, bt] = getpar(th);

% RIVID forces the model output to have the correct values for the

% first 3 samples, calculating the model output using filter removes

% this issue

ymtest = filter(bt, at, z(:,2));

% Plots this filtered value

subplot(211)

plot(ymtest, 'r--');

ymtest = ymtest';

% Finds the error using y = fit + error and saves the values in

% errortest

for ee = 1:1:14

errortest(ee,:) = z(ee, 1) - ymtest(ee);

end

% Calculates rt2

myrt2test = 1-(var(errortest))/var(z(:, 1));

hold off

end

% Cycles through all patients (303) and calculates the error

for kk = 1:1:303

zjj=[yy((kk), :)' uu((kk), :)'];

ym1=filter(bt, at, zjj(:,2));

for ee = 1:1:14

error(:,ee) = yy(kk, ee) - ym1(ee);

end

% Saves the patient number and Rt2 value in calrt2

calrt2(kk, 1) = kk;

calrt2(kk, 2) = 1 - (var(error))/var(zjj(:, 1));

% Check if value is great than 0.5, if so increases the number of Rt2

% greater than 0.5 and 0.2 by 1.

if calrt2(kk,2) > 0.5

gt\_2 = gt\_2 + 1;

gt\_5 = gt\_5 +1;

% Checks if greater than 0.2 and increase the number of Rt2 greater

% than 0.2 by 1

elseif calrt2(kk,2) > 0.2

gt\_2 = gt\_2 + 1;

end

end

% Calculates the mean, max and min value of Rt2

avgrt2 = mean(calrt2(:,2));

maxrt2 = max(calrt2(:,2));

minrt2 = min(calrt2(:,2));

% Finds the patient number for the max and min Rt2

for kk = 1:1:303

if calrt2(kk,2) == maxrt2

maxpat = calrt2(kk,1);

elseif calrt2(kk,2) == minrt2

minpat = calrt2(kk,1);

end

end

% Places the values for avg, min, max and the patient numbers into the

% calrt2 table

calrt2(1,3) = avgrt2;

calrt2(1,4) = maxrt2;

calrt2(2,4) = maxpat;

calrt2(1,5) = minrt2;

calrt2(2,5) = minpat;

% Gives the number and percentage of values of Rt2 over 0.2

calrt2(1,6) = (gt\_2);

calrt2(2,6) = ((gt\_2/kk)\*100);

% Gives the number and percentage of values of Rt2 over 0.5

calrt2(1,7) = (gt\_5);

calrt2(2,7) = ((gt\_5/kk)\*100);

% Plots the results as two tables. One showing all the Rt2 values and the

% other just giving the average, minimum, maximum, x>0.2 and x>0.5, along

% with the patient numbers when applicable.

VariableNames1 = {'AverageRt2', 'MaxRt2', 'MinRt2', 'x', 'y'};

VariableNames2 = {'PatientNumber', 'CalculatedRt2', 'Average Rt2', 'Max Rt2 and PatientNumber', 'Min Rt2 and PatientNumber', 'x > 0.2(Number and Percentage)', 'x > 0.5(Number and Percentage)'};

t1 = table(calrt2(1:2,3), calrt2(1:2,4), calrt2(1:2,5), calrt2(1:2,6), calrt2(1:2,7), 'VariableNames', VariableNames1)

t2 = uitable(uifigure(1), 'data', calrt2(:, 1:7), 'columnname', VariableNames2, 'Position',[30 30 500 370]);

## 8.5.0 – Patient Generalised Models

% Tests each patient made TF using all the patients (1 to 303) data and

% saves the percentages in a spreadsheet

% Respnses of each patient to each general TF was already known and saved,

% therefore to save time the spreadsheet "GenTf.xlsx" is read, each patient

% has a different sheet for all other patient responses.

close all

close all

% load outputs i.e. standardised INR values at clinic visits

yy=load('outputs.dat');

% load inputs i.e. change of dose in mg

uu=load('inputs.dat');

sc = 2;

myerror(1:303,1) = 1:303;

% First loop for the general patient

for ii = 1:1:303

over0x2 = 0;

over0x5 = 0;

%GenTf = readmatrix("GenTf.xlsx", "sheet", ii);

% These patients were not able to have TFs made

if ii~=121 && ii~=128 && ii~=194 && ii~=224 && ii~=235

z = [yy(ii, :)' uu(ii, :)'];

[th, stats, e]=rivid(z, [1 1 0 0; 3 3 5 0], sc);

rt2=stats(3);

[at, bt]=getpar(th);

% Cycles through all patients for calculations

for kk = 1:1:303

z = [yy(kk, :)' uu(kk, :)'];

ya = filter (bt, at, z(:, 2));

% GenTf(kk, 2:15) = ya;

% ya = GenTf(kk, 2:15)';

% Calculates rt2

my\_rt2 = 1-(var(ya - z(:, 1))/var(z(:, 1)));

% calculated rt2

allrt2(kk, ii) = my\_rt2;

% calculated rt2

myerror(kk, 16) = my\_rt2;

% Error saved

myerror(kk, 2:15) = (ya - z(:, 1));

% Mean absolute of error calculated

myerror(kk, 17) = mean(abs(myerror(kk, 2:15)));

% Varience of error calculated

myerror(kk, 18) = var(myerror(kk,2:15));

if allrt2(kk,ii) > 0.2

over0x2 = over0x2 + 1;

end

if allrt2(kk,ii) > 0.5

over0x5 = over0x5 + 1;

end

end

%writematrix(GenTf, "GenTF.xlsx", "sheet", ii)

myerror(1, 19) = over0x2;

myerror(1, 20) = over0x5;

data(1:303, 1) = myerror(1:303, 1);

data(1:303, 2) = myerror(1:303, 16);

data(1:303, 3) = myerror(1:303, 17);

data(1:303, 4) = myerror(1:303, 18);

data(1:303, 5) = myerror(1:303, 19);

data(1:303, 6) = myerror(1:303, 20);

tablemyerror = array2table(data(1:303,1:6), 'VariableNames', {'PatientNumber', 'RT2', 'MeanError', 'VarError', 'Over0x2', 'Over0x5'})

% Saves table to spreadsheet

%writetable(tablemyerror, "ModelResponseGeneralTF.xlsx", "Sheet", ii)

end

end

## 8.6.0 – DLR Models

% Calculates the best DLR variables for each patient depending on inputs

% given, ie nvr? DLR equation? or Delay?

% 03/02/2020

close all

close all

% load outputs i.e. standardised INR values at clinic visits

yy=load('outputs.dat');

% load inputs i.e. change of dose in mg (No time delay)

%uu=load('inputs.dat');

uu = load('InputDelayData.dat'); % One input time delay

% select a patient

ii = 20;

% Counter variables for rt2 over a number

jj = 0;

kk = 0;

for ii = 1:1:303

y = yy(ii,:)';

x = uu(ii,:)';

% Sets the equation

z = [x];

TVP = 0;

% Sets the nvr

nvr = 0.1;

% Calculates the variables

[fit, fitse, par, parse] = dlr(y, z, TVP, nvr);

for ee = 1:1:14

error(ee) = y(ee) - fit(ee);

end

calrt2(ii,1) = ii;

calrt2(ii,2) = 1-(var(error))/var(y);

if calrt2(ii,2) > 0.5

jj = jj + 1;

end

if calrt2(ii,2) > 0.2

kk = kk + 1;

end

% Various data thats needed

dataRes(ii, 1) = ii;

dataRes(ii, 2) = calrt2(ii, 2);

dataRes(ii, 3) = mean(abs(error(1:14)));

dataRes(ii, 4) = var(error(1:14));

dataRes(ii, 5) = par(1,1);

% figure(1)

% subplot(211)

% title('c1')

% plot(par(:,1))

% hold on

%

% subplot(212)

% title('c2')

% plot(par(:,2))

% hold on

%

end

%saveas(figure(1), 'c1C2allnvr0x01p20.png');

hold off

% Calculates the percentages

calrt2(1,4) = jj;

calrt2(2,4) = jj \* 100 / ii;

calrt2(1,3) = kk;

calrt2(2,3) = kk \* 100 / ii;

% dataRes(1, 6) = kk;

% dataRes(1, 7) = jj;

% Saves the data as spreadsheet

% dlrRes = array2table(dataRes(1:303,1:5), 'VariableNames', {'PatientNumber', 'RT2', 'MeanError', 'VarError', 'c1'});

% writetable(dlrRes, "DLRModelResponseDelay.xlsx")

## 8.7.0 – Gain Model

### 8.7.1 – Gain Research

% Model Testing simple multiplier

% 10/02/2020

close all

close all

clear all

% load outputs i.e. standardised INR values at clinic visits

yy=load('outputs.dat');

% load inputs i.e. change of dose in mg

uu=load('inputs.dat');

gt\_2 = 0;

gt\_5 = 0;

% Cycles through all patients (303) and calculates the error

% for steps of 0.01 from -1 to 1, kk = 1:1:210

% multi = (kk/100)-1.01

% For steps of 0.01 from -2 to 2, kk = 1:1:410

% multi = (kk/100)-2.01

% Specific Range of multi between -1.219 and 0 in steps of 0.001

% kk = 1:1:1220, multi = (kk/1000)-1.22

for kk = 1:1:1220

multi = (kk/1000) - 1.22; % Multiplier between -1 and 1 insteps of 0.01

avgrt2(kk,1) = multi;

for ii = 1:1:303

% Model fit saved in patients row

fit(ii,:) = uu(ii,:)\*multi;

% Calculates the error for each point

for ee = 1:1:14

error(:,ee) = yy(ii, ee) - fit(ii, ee);

end

% Saves the Rt2 value in the patients row and

% in the multipliers column

calrt2(ii, kk) = 1 - (var(error))/var(yy(ii,:));

if calrt2(ii,kk) > 0.5

gt\_2 = gt\_2 + 1;

gt\_5 = gt\_5 +1;

% Checks if greater than 0.2 and increase the number of Rt2 greater

% than 0.2 by 1

elseif calrt2(ii,kk) > 0.2

gt\_2 = gt\_2 + 1;

end

end

% Finds the average Rt2 value for each multiplier over

% All patients

avgrt2(kk,2) = mean(calrt2(:,kk));

% Finds the percentage of Rt2 values over 0.2

% for each multiplier over all patients

avgrt2(kk,3) = (gt\_2\*100)/303;

% Finds the percentage of Rt2 values over 0.5

% for each multiplier over all patients

avgrt2(kk,4) = (gt\_5\*100)/303;

gt\_2 = 0;

gt\_5 = 0;

end

% Fifth column, first row is the max percentage rt2 value

% over 0.2

avgrt2(1, 5) = max(avgrt2(:,3));

% Fourth column, second row is the max percentage rt2 value

% over 0.5

avgrt2(1, 6) = max(avgrt2(:,4));

for kk = 1:1:1220

if avgrt2(1,5) == avgrt2(kk,3)

avgrt2(2,5) = avgrt2(kk,1);

end

if avgrt2(1,6) == avgrt2(kk,4)

avgrt2(2,6) = avgrt2(kk,1);

end

end

VariableNames1 = {'Over02Rt2', 'Over05Rt2'};

RowNames1 = {'Percentage', 'Mulitplier'};

t1 = table(avgrt2(1:2,5), avgrt2(1:2,6), 'VariableNames', VariableNames1, 'RowNames', RowNames1)

VariableNames2 = {'Mulitplier', 'AverageRt2 for all patients at multiplier', 'Percentage over 0.2RT2', 'Percentage over 0.5RT2', 'Highest percentage over 0.2 and that mulitplier', 'Highest percentage over 0.5 and that mulitplier'};

t2 = uitable(uifigure(1), 'data', avgrt2(:,:), 'columnname', VariableNames2, 'Position',[30 30 500 370]);

### 8.7.2 – More in-depth Gain Model Research

% Model Testing simple multiplier

% More precise investigation for gain model

% 10/02/2020

close all

close all

clear all

% load outputs i.e. standardised INR values at clinic visits

yy=load('outputs.dat');

% load inputs i.e. change of dose in mg

uu=load('inputs.dat');

gt\_2 = 0;

gt\_5 = 0;

error(1,1:14) = 0;

rt2(1:401, 1:303) = 0;

gain(1:401,1:3) = 0;

% Counter changes depending on the range and precision wanted, in this

% example the range is -4 to 0 in steps of 0.01

for counter = 1:1:401

gt\_2 = 0;

gt\_5 = 0;

% Calculates the gain and saves in the variable

gain(counter,1) = (counter/100)-4.01;

out = gain(counter,1)\*uu(1:303,:);

for ii = 1:1:303

% Calculates the error

for ee = 1:1:14

error(1,ee) = yy(ii, ee) - out(ii,ee);

end

rt2(counter, ii) = 1 - (var(error(1,:)))/var(yy(ii,:));

if rt2(counter,ii) > 0.5

gt\_5 = gt\_5 + 1;

end

if rt2(counter,ii) > 0.2

gt\_2 = gt\_2 + 1;

end

end

gain(counter,2) = gt\_2;

gain(counter,3) = gt\_5;

end

% Saves the responses found and various error calculations

% tablemyerror = array2table(gain(1:401,1:3), 'VariableNames', {'Gain', 'Over0x2', 'Over0x5'})

% writetable(tablemyerror, "ModelResponseGain.xlsx")

## 8.8.0 – Second Order Model Controller

### 8.8.1 – Poles and Disturbances

% Plots the Output of a second order TF using a simulink PIP model

% with coefficents set from pole placement.

% George Caddick

% 20/01/2020

clear all

% Set Sample Time

T = 50;

% Set values for transfer fucntion and poles

at0 = [1 -0.2324 -0.1491]; % [1 a1z^-1 a2z^-2]

bt0 = [0 0.2025 0.2056]; % [0 b1z^-1 b2z^-2]

at = at0;

bt = bt0;

% Change Poles Here

poles = [0.0];

% Calculates f0 and ki

v = pip(at(2:3), bt(2:3), poles);

% Sets values for f0 and ki

f0 = 0.5

f1 = v(2,1)

g1 = v(3,1)

ki = v(4,1)

% Loads the Simulink model

sim('secondOrderModel');

% Plots the Model Output

figure(1)

subplot(211)

plot(ans.yk, 'b')

xlabel('Time in days');

ylabel('INR');

xticks([0:2:T])

xlim([0 T])

%ylim([-0.5 2])

title({"Second Order Model output with poles of "+num2str(poles)})

grid on

% Plots the Model Input

subplot(212)

plot(ans.uk, 'k')

xlabel('Time in days');

xticks([0:2:T])

xlim([0 T])

ylabel('Change in Dosage');

grid on

%ylim([1 3])

% Displays Controller Variables

Names = {'f0', 'f1', 'g1', 'ki'};

VariableNames1 = {'f0', 'f1', 'g1', 'ki'};

xx = v(1:4,1)';

t1 = table(xx(1), xx(2), xx(3), xx(4), 'VariableNames', Names);

% Saves the Model figures

%saveas(figure(1), "secondOrderpoles0x7.png")

### 8.8.2 – Monte Carlo Simulation

% Monte Carlo simulation on a second order model

% 03/02/20

clear all

close all

% Set Sample Time

T = 30;

% Set tick change

n = 2;

% Original Values

at0 = [1 -0.2324 -0.1491];

bt0 = [0 0.2025 0.2056];

% Simulation Values

at = at0;

bt = bt0;

poles = [0];

% Calculates Controller Coefficients

v = pip(at(2:3), bt(2:3), poles);

% Sets values for controller coeffients

f0 = v(1,1);

f1 = v(2,1);

g1 = v(3,1);

ki = v(4,1);

% Loads the Simulink model

sim('secondOrderModel');

% Plot the original output

figure(1)

subplot(211)

plot(ans.yk, 'b')

xlabel('Time in days');

ylabel('INR');

xticks([0:n:T])

xlim([0 T])

%ylim([-0.5 2])

title('Second Order Model output')

grid on

subplot(212)

plot(ans.uk, 'k')

xlabel('Time in days');

grid on

xticks([0:n:T])

ylabel('Change in Dosage');

%ylim([1 3])

% Creates a randomly ditributed

% error term centered around the

% original term.

L = 1000; % Easy choice of how many numbers

a2error = randn(L,1)/10; % Creates the error

a2error = a2error + at0(1,2); % Adds to orginal coefficient

for k = 1:1:L % 1:1:L

at(1,2) = a2error(k);

sim('MPC\_1\_test');

figure(3)

plot(ans.yk)

hold on

end

figure(3)

xlabel('Time in days');

ylabel('INR');

xticks([0:n:T])

xlim([0 T])

%ylim([0 2])

grid on

title('Second Order Model output with changes around the orignal value of a1')

hold off

at(1,2) = at0(1,2); % Sets errored term back to original

a3error = randn(L,1)/10; % Creates the error

a3error = a3error + at0(1,3); % Adds to orginal coefficient

for k = 1:1:L

at(1,3) = a3error(k);

sim('MPC\_1\_test');

figure(4)

plot(ans.yk)

hold on

end

figure(4)

xlabel('Time in days');

ylabel('INR');

xticks([0:n:T])

xlim([0 T])

%ylim([0 2])

grid on

title('Second Order Model output with changes around the orignal value of a2')

hold off

at(1,3) = at0(1,3); % Sets errored term back to original

b1error = randn(L,1)/10; % Creates the error

b1error = b1error + bt0(1,1); % Adds to orginal coefficient

for k = 1:1:L

bt(1,1) = b1error(k);

sim('MPC\_1\_test');

figure(5)

plot(ans.yk)

hold on

end

figure(5)

xlabel('Time in days');

ylabel('INR');

xticks([0:n:T])

xlim([0 T])

grid on

%ylim([-0.5 1.5])

title('Second Order Model output with changes around the orignal value of b1')

hold off

bt(1,1) = bt0(1,1); % Sets errored term back to original

b2error = randn(L,1)/10; % Creates the error

b2error = b2error + bt0(1,2); % Adds to orginal coefficient

for k = 1:1:L

bt(1,2) = b2error(k);

sim('MPC\_1\_test');

figure(6)

plot(ans.yk)

hold on

end

figure(6)

xlabel('Time in days');

ylabel('INR');

xticks([0:n:T])

xlim([0 T])

grid on

%ylim([-0.5 1.5])

title('Second Order Model output with changes around the orignal value of b2')

hold off

bt(1,2) = bt0(1,2); % Sets errored term back to original

b3error = randn(L,1)/10; % Creates the error

b3error = b3error + bt0(1,3); % Adds to orginal coefficient

for k = 1:1:L

bt(1,3) = b3error(k);

sim('MPC\_1\_test');

figure(7)

plot(ans.yk)

hold on

end

figure(7)

xlabel('Time in days');

ylabel('INR');

xticks([0:n:T])

xlim([0 T])

%ylim([0 3])

title('Second Order Model output with changes around the orignal value of b3')

grid on

hold off

## 8.9.0 – PG73 Controller

### 8.9.1 – Poles and Disturbances

% Plots the Output of generalised patient 73 TF using a simulink PIP model

% with coefficents set from pole placement.

% George Caddick

% 20/01/2020

clear all

% Set Sample Time

T = 30;

% Set values for transfer fucntion and poles

at0 = [1 0.237]; % [1 a1z^-1 a2z^-2]

bt0 = [0 -0.2545 0.01685]; % [0 b1z^-1 b2z^-2]

at = at0;

bt = bt0;

poles = [0];

% Calculates f0 and ki

v = pip(at(2), bt(2:3), poles);

% Sets values for f0 and ki

f0 = v(1)

f1 = v(2)

ki = v(3)

% Loads the Simulink model

sim('PatientGen73Model');

figure(1)

subplot(211)

plot(ans.yk, 'b')

xlabel('Time in days');

ylabel('INR');

xticks([0:2:T])

xlim([0 T])

%ylim([-0.5 2])

% title({"PG73 Model output with poles of "+num2str(poles(1))+" and "+num2str(poles(2))})

title({"PG73 Model output with poles of "+num2str(poles(1))})

grid on

subplot(212)

plot(ans.uk, 'k')

xlabel('Time in days');

xticks([0:2:T])

xlim([0 T])

ylabel('Change in Dosage');

grid on

%ylim([1 3])

Names = {'f0', 'f1', 'ki'};

VariableNames1 = {'f0', 'f1', 'ki'};

xx = v(1:3,1)';

t1 = table(xx(1), xx(2), xx(3), 'VariableNames', Names);

### 8.9.2 – Monte Carlo Simulation

% Monte Carlo simulation on the PG73 System

% 03/02/20

clear all

close all

% Set Sample Time

T = 30;

% Set tick change

n = 2;

% Original Values

at0 = [1 0.237]; % [1 a1z^-1]

bt0 = [-1.089 -0.2545 0.01685]; % [0 b1z^-1 b2z^-2]

% Simulation Values

at = at0;

bt = bt0;

poles = [-0.5];

% Calculates Controller Coefficients

v = pip(at(2), bt(2:3), poles);

% Sets values for controller coeffients

f0 = v(1,1);

f1 = v(2,1);

ki = v(3,1);

% Loads the Simulink model

sim('Gen73Model');

% Plot the original output

figure(1)

subplot(211)

plot(ans.yk, 'b')

xlabel('Time in days');

ylabel('INR');

xticks([0:n:T])

xlim([0 T])

%ylim([-0.5 2])

title('PG73 Model output')

grid on

subplot(212)

plot(ans.uk, 'k')

xlabel('Time in days');

grid on

xticks([0:n:T])

ylabel('Change in Dosage');

%ylim([1 3])

% Creates a randomly ditributed

% error term centered around the

% original term.

L = 1000; % Easy choice of how many numbers

a2error = randn(L,1)/2; % Creates the error

a2error = a2error + at0(1,2); % Adds to orginal coefficient

for k = 1:1:L % 1:1:L

at(1,2) = a2error(k);

sim('Gen73Model');

figure(3)

plot(ans.yk)

hold on

end

figure(3)

xlabel('Time in days');

ylabel('INR');

xticks([0:n:T])

xlim([0 T])

%ylim([0 2])

grid on

title('PG73 Model output with changes around the orignal value of a1')

hold off

at(1,2) = at0(1,2); % Sets errored term back to original

b1error = randn(L,1)/2; % Creates the error

b1error = b1error + bt0(1,1); % Adds to orginal coefficient

for k = 1:1:L

bt(1,1) = b1error(k);

sim('Gen73Model');

figure(5)

plot(ans.yk)

hold on

end

figure(5)

xlabel('Time in days');

ylabel('INR');

xticks([0:n:T])

xlim([0 T])

grid on

%ylim([-0.5 1.5])

title('PG73 Model output with changes around the orignal value of b1')

hold off

bt(1,1) = bt0(1,1); % Sets errored term back to original

b2error = randn(L,1)/2; % Creates the error

b2error = b2error + bt0(1,2); % Adds to orginal coefficient

for k = 1:1:L

bt(1,2) = b2error(k);

sim('Gen73Model');

figure(6)

plot(ans.yk)

hold on

end

figure(6)

xlabel('Time in days');

ylabel('INR');

xticks([0:n:T])

xlim([0 T])

grid on

%ylim([-0.5 1.5])

title('PG73 Model output with changes around the orignal value of b2')

hold off

bt(1,2) = bt0(1,2); % Sets errored term back to original

b3error = randn(L,1)/2; % Creates the error

b3error = b3error + bt0(1,3); % Adds to orginal coefficient

for k = 1:1:L

bt(1,3) = b3error(k);

sim('Gen73Model');

figure(7)

plot(ans.yk)

hold on

end

figure(7)

xlabel('Time in days');

ylabel('INR');

xticks([0:n:T])

xlim([0 T])

%ylim([0 3])

title('PG73 Model output with changes around the orignal value of b3')

grid on

hold off

## 8.10.0 – Gain Model Controller

### 8.10.1 – Controller Variable Testing

% Testing the Controller variables to find the optimum for this gain

clear all

% Set Sample Time

T = 30;

% Set tick change

n = 2;

% Set the model gain

gain = -1.141;

counter = 1;

% For loops go through testing variables f0 and ki and saves the response

for f0 = 0.8:0.01:1

for ki = -5:0.1:-3

sim('GainModel');

responseout(counter, :) = ans.yk';

varout(counter, 1) = f0;

vareout(counter, 2) = ki;

responsein(counter,:) = ans.uk';

counter = counter +1;

end

end

### 8.10.2 – Monte Carlo Simulation

% Monte Carlo Simulation of the gain using the optimum controller variables

% found earlier

% George Caddick

% 01/03/2020

% clear all

% Set Sample Time

T = 30;

% Set tick change

n = 2;

% Set the model gain

gain0 = -1.141;

f0 = 0.88;

ki = -5;

v = [f0, ki];

gain = gain0;

% Loads the Simulink model

sim('GainModel');

figure(1)

subplot(211)

plot(ans.yk, 'b')

xlabel('Time in days');

ylabel('INR');

xticks([0:1:T])

xlim([0 T])

%ylim([-0.5 2])

title({"Gain model output with an f0 of "+num2str(f0)+" and ki of "+num2str(ki)})

subplot(212)

plot(ans.uk, 'k')

xlabel('Time in days');

xticks([0:1:T])

xlim([0 T])

ylabel('Change in Dosage');

%ylim([1 3])

Names = {'f0', 'ki'};

VariableNames1 = {'f0', 'ki'};

xx = v(1,1:2)';

t1 = table(xx(1), xx(2), 'VariableNames', Names)

L = 1000; % Easy choice of how many numbers

gainerror0 = randn(L,1)/5; % Creates the error

gainerror = gainerror0 + gain0; % Adds to orginal coefficient

for k = 1:1:L % 1:1:L

gain = gainerror(k);

sim('GainModel');

figure(2)

plot(ans.yk)

hold on

end

figure(2)

xlabel('Time in days');

ylabel('INR');

xticks([0:n:T])

xlim([0 T])

%ylim([0 2])

grid on

title('Gain Model output with changes around the orignal value gain')

hold off

## 8.11.0 – DLR Eq1 Controller

### 8.11.1 – Controller Variable Testing

% DLR Eq1 optimum control variables

% 22/04/2020

close all

close all

% load outputs i.e. standardised INR values at clinic visits

yy=load('outputs.dat');

% load inputs i.e. change of dose in mg (No time delay)

uu=load('inputs.dat');

% Load C1 variables for each patient

c1 = readmatrix("DLRc1.xlsx");

% Set Time

T = 30;

% Cycle through all patients for testing

for ii = 1:1:303

clear errorsorted

clear error

% Finds the correct c1 variable for this patient

gain = c1(ii, 2);

counter = 1;

% Finds the best controller variables for this patient

for f0 = 0.1:0.1:0.9

for ki = -8:0.25:-5

sim('GainModel');

output = (ans.yk)-2.5;

error(counter, 1) = mean(abs(output));

error(counter, 2) = f0;

error(counter, 3) = ki;

counter = counter +1;

end

end

% Sorts by mean error and saves these variables

errorsorted(:, :) = sortrows(error, 1, "ascend");

patientvar(ii, 2:4) = errorsorted(1, 1:3);

end

% Saves te variables as a spreadsheet

dlrResPI = array2table(patientvar(1:303, 1:4), 'VariableNames', {'PatientNumber', 'MeanAbsError', 'f0', 'ki'});

writetable(dlrResPI, "DLRModelPI.xlsx")

### 8.11.2 – Monte Carlo Simulation

% Monte Carlo Simulation for four specific patients chosen randomly

% Set Sample Time

T = 30;

% Set tick change

n = 2;

% Load C1 variables for each patient

c1 = readmatrix("DLRc1.xlsx");

% Load controller variables

contVar = readmatrix("DLRModelPI.xlsx");

pat = [30 85 166 192];

gain0 = [c1(30, 2), c1(85, 2), c1(166, 2), c1(192, 2)];

f00 = [contVar(30,3), contVar(85,3), contVar(166,3), contVar(192,3)];

ki0 = [contVar(30,4), contVar(85,4), contVar(166,4), contVar(192,4)];

L = 1000; % Easy choice of how many numbers

error0 = randn(L,1)/5; % Creates the error

for j = 1:1:4

gainerror = error0 + gain0(j); % Adds to orginal coefficient

ki = ki0(j);

f0 = f00(j)

for k = 1:1:L % 1:1:L

gain = gainerror(k);

sim('GainModel');

figure(j)

plot(ans.yk)

hold on

end

figure(j)

xlabel('Time in days');

ylabel('INR');

xticks([0:n:T])

xlim([0 T])

%ylim([0 2])

grid on

%title('DLR Eq3 Model output with changes around the orignal gain for patient '+num2str(pat(1)))

hold off

end

## 8.12.0 - Proposed Timeline and Actual Timeline

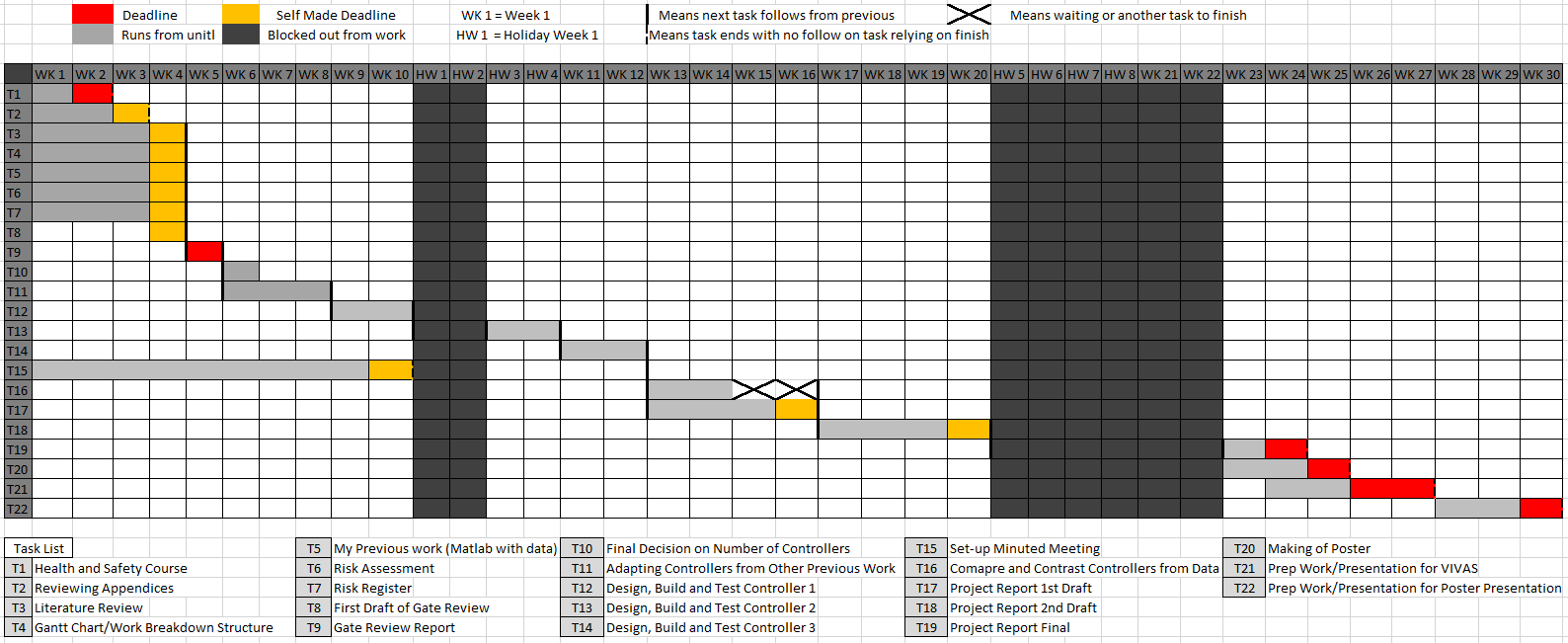


Figure 78 - Shows the proposed timeline (LEFT) and the Actual timeline (RIGHT)



As seen in the Gantt Chart, the beginning section deadlines were kept to, however the increase of work for the models meant that tasks were dropped, such two tasks for building controllers, and the time was used to develop the models.

# 9.0.0 - References

Murphy, S. (2003). Optimal dynamic treatment regimes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 65(2), pp.331-355.

Lim, G (2019). *Milestone 2: Warfarin: from rat poison to clinical use*. [online] Nature Reviews Cardiology. Available at: https://www.nature.com/articles/nrcardio.2017.172 [Accessed 28 Oct. 2019].

Milner, K. and Bonaventura, K. (2019). *Self-management of Warfarin: An Approach to Increase Patient Engagement in Care*.

MedicineNet. (2019). *Definition of Thromboembolism*. [online] Available at: https://www.medicinenet.com/script/main/art.asp?articlekey=25032 [Accessed 28 Oct. 2019].

www.heart.org. (2019). *A Patient's Guide to Taking Warfarin*. [online] Available at: https://www.heart.org/en/health-topics/arrhythmia/prevention--treatment-of-arrhythmia/a-patients-guide-to-taking-warfarin [Accessed 29 Oct. 2019].

Keeling, D. (2019). [online] Ouh.nhs.uk. Available at: https://www.ouh.nhs.uk/services/referrals/specialist-medicine/documents/anticoagulant-protocols.pdf [Accessed 29 Oct. 2019].

Wysowski, D.K., Nourjah, P. and Swartz, L., 2007. Bleeding complications with warfarin use: a prevalent adverse effect resulting in regulatory action. *Archives of internal medicine*, *167*(13), pp.1414-1419.

Makohusová, M. and Bátorová, A. (2019). *Comparison of Two Different Techniques Of Warfarin Dosing Determination - A Chemometrics Study*. [online] PubMed Central (PMC). Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6706742/ [Accessed 15 Oct. 2019].

Baglin, T., Keeling, D. and Watson, H. (2019). *Guidelines on oral anticoagulation (warfarin): third edition--2005 update. - PubMed - NCBI*. [online] Ncbi.nlm.nih.gov. Available at: https://www.ncbi.nlm.nih.gov/pubmed/16409292 [Accessed 10 Oct. 2019].

nhs.uk. (2019). *Alcohol units*. [online] Available at: https://www.nhs.uk/live-well/alcohol-support/calculating-alcohol-units/ [Accessed 29 Oct. 2019].

Wilson, E., Clairon, Q., Henderson, R. and Taylor, C. (2019). *Model Predictive and Proportional Integral Control of Blood Clotting Speed Using Warfarin when Data are Missing - IEEE Conference Publication*. [online] Ieeexplore.ieee.org. Available at: https://ieeexplore.ieee.org/document/8567452 [Accessed 26 Sep. 2019].

Coaguchek.co.uk. (2019). *CoaguChek® INRange*. [online] Available at: http://www.coaguchek.co.uk/coaguchek\_patient/en\_GB/home.html [Accessed 29 Oct. 2019].

Carmen V. Doran., J. Geoffrey Chase., Geoffrey M. Shaw., Katherine T. Moorhead., Nicolas H. Hudson. (2005). *Derivative weighted active insulin control algorithms and intensive care unit trials - Control Engineering Practice, Volume 13, Issue 9, September 2005, Pages 1129-1137.* Available at: https://www.sciencedirect.com/science/article/pii/S0967066104002163 [Accessed 29 Oct. 2019].

Avery, P., Clairon, Q., Henderson, R., Taylor, C., and Wilson, E. (2019). *Robust and Adaptive Anticoagulant Control.* Submitted under review.

Pagano, M. and Chandler, W., 2012. *Bleeding Risks And Response To Therapy In Patients With INR Higher Than 9. - Pubmed - NCBI*. [online] Ncbi.nlm.nih.gov. Available at: <https://www.ncbi.nlm.nih.gov/pubmed/23010709> [Accessed 27 February 2020].

Bragg, L. and Amir, J., 2003. [online] My.clevelandclinic.org. Available at: <https://my.clevelandclinic.org/ccf/media/Files/anticoagulation-clinics/practical-tips-for-warfarin-dosing-and-monitoring.pdf> [Accessed 28 February 2020].

Cambridgeshireandpeterboroughccg.nhs.uk. 2017. [online] Available at: <https://www.cambridgeshireandpeterboroughccg.nhs.uk/easysiteweb/getresource.axd?assetid=11888&type=0&servicetype=1> [Accessed 28 February 2020].

Taylor, C., Chotai, A. and Young, P., 2001. Design and application of PIP controllers: robust control of the IFAC93 benchmark. *Transactions of the Institute of Measurement and Control*, 23(3), pp.183-200.