Summer Fellowship Report

On

Scilab Case Studies

Submitted by

ALLU RAM CHARAN

B.Tech 3rd Year

Department of Mechanical Engineering

Vishnu Institute of Technology, Bhimavaram

Under the guidance of

Prof. Prabhu Ramchandran

Aerospace Engineering Department

IIT Bombay

Mentor

Mrs. Rashmi Patankar

FOSSEE Summer Fellowship 2026

IIT Bombay

July 2026

Acknowledgment

I would like to express my sincere gratitude to the FOSSEE Team, IIT Bombay, for providing me the opportunity to participate in the FOSSEE Summer Fellowship 2026. This fellowship gave me a valuable opportunity to work with Scilab and Xcos for implementing engineering case studies using open-source tools.

I am deeply thankful to Prof. Prabhu Ramchandran for leading this initiative, which promotes the use of free and open-source software in education and engineering applications.

I would like to express my sincere thanks to Mrs. Rashmi Patankar for her continuous guidance, support, and valuable suggestions throughout the fellowship period. Her feedback helped me improve the quality of my case studies, simulations, GUI implementation, documentation, and overall project presentation.

I would also like to thank the Scilab Team, FOSSEE, IIT Bombay, for providing a platform to contribute to open-source educational resources. Through this fellowship, I improved my understanding of Scilab programming, Xcos simulation, GUI development, technical documentation, and engineering problem implementation.

Finally, I express my gratitude to Vishnu Institute of Technology, Bhimavaram, my faculty members, friends, and family for their support and encouragement during this work.

Contents

Acknowledgment **i**

1 Introduction **1**

2 Case Study 1: Thermal Fault Localization in Pipelines Using Inverse Heat Conduction and Xcos Simulation **2**

2.1 Abstract 2
2.2 Problem Formulation 2
2.3 Flowchart 3
2.4 Results and Discussion 5

3 Case Study 2: Xcos-Based Implementation of Input Shaping Control for Overhead Crane Swing Reduction **12**

3.1 Abstract 12
3.2 Problem Formulation 12
3.3 Flowchart 13
3.4 Results and Discussion 15

4 Case Study 3: Scilab GUI for Analytical Design and Performance Evaluation of an Automotive Disc Brake **21**

4.1 Abstract 21
4.2 Problem Formulation 21
4.3 Flowchart 22
4.4 Results and Discussion 24

5 Learning Outcomes **28**

6 Conclusion **29**

7 References **30**

Chapter 1

Introduction

The Free and Open Source Software for Education (FOSSEE) project at IIT Bombay promotes the use of open-source software tools in technical education. Scilab is one such open-source numerical computation software that can be used for mathematical modelling, simulation, data analysis, graphical user interface development, and engineering applications. Xcos, which is available with Scilab, provides a visual modelling and simulation environment for dynamic systems.

During the FOSSEE Summer Fellowship 2026, three Scilab case studies were completed in the mechanical engineering domain. These case studies focused on thermal analysis, control-based simulation using Xcos, and analytical design using a Scilab GUI dashboard.

The three completed case studies are:

- **Case Study 1:** Thermal Fault Localization in Pipelines Using Inverse Heat Conduction and Xcos Simulation.
- **Case Study 2:** Xcos-Based Implementation of Input Shaping Control for Overhead Crane Swing Reduction.
- **Case Study 3:** Scilab GUI for Analytical Design and Performance Evaluation of an Automotive Disc Brake.

The first case study deals with thermal fault localization in a pipeline using simulated thermal data. The second case study implements input shaping control for reducing payload swing in an overhead crane system using Scilab and Xcos. The third case study develops a Scilab GUI dashboard for the analytical design and performance evaluation of an automotive disc brake.

These case studies demonstrate the application of Scilab and Xcos in thermal engineering, control systems, and mechanical design analysis. The work also highlights the usefulness of open-source tools for academic learning, engineering simulation, and interactive application development.

Chapter 2

Case Study 1: Thermal Fault Localization in Pipelines Using Inverse Heat Conduction and Xcos Simulation

2.1 Abstract

Thermal faults in pipelines may occur due to insulation damage, leakage, or localized heat loss. These faults disturb the normal temperature distribution and may reduce the thermal efficiency of the system. This case study presents a simulation-based Scilab and Xcos implementation for detecting and localizing a thermal fault in a pipeline.

The pipeline is modelled as a one-dimensional transient heat conduction system. Scilab is used to simulate normal and faulty temperature profiles using the finite difference method. Simulated sensor readings with small artificial noise are generated to represent measurement-like conditions. An inverse estimation method is applied by comparing simulated sensor temperature data with simulated model data and selecting the fault condition with minimum error. A fault index method is also used to locate the region of maximum temperature deviation. Xcos is used to visualize the dynamic thermal response under weak, medium, and severe simulated fault conditions.

All experiments in this case study are performed using simulated thermal data only. No real pipeline hardware, real temperature sensors, real pipeline measurements, or experimental validation are used.

2.2 Problem Formulation

Thermal faults in pipelines are difficult to detect when the fault is not directly visible from outside. A fault such as insulation damage, leakage, or local heat loss may not immediately stop the system, but it can slowly reduce thermal efficiency. In a healthy pipeline, temperature changes smoothly along the length of the pipe. When a fault occurs, additional heat loss takes place near the damaged region and a local temperature drop appears in the temperature profile.

In this work, a 1 m long pipeline is considered as a simulation model. The left side of the pipe is maintained at a higher temperature and the right side is maintained at a lower temperature. A thermal fault is introduced between 0.45 m and 0.55 m and is modelled as an additional heat loss region. The normal and faulty temperature profiles are calculated using a one-dimensional finite difference heat conduction model in Scilab.

The solution approach has two main parts. Xcos is used to create a dynamic block model that shows how the simulated pipeline temperature response changes under weak, medium, and severe fault conditions. Scilab is used to calculate the spatial temperature distribution, generate simulated sensor readings, perform inverse estimation, and compute the fault index.

The aim is to estimate the simulated fault location from the simulated temperature response. The case study is not a direct replication of any single research paper. The reference papers are used only to support the background concepts of inverse heat conduction, thermal monitoring, and sensor-based temperature analysis.

2.3 Flowchart

The methodology starts with the formulation of the pipeline thermal fault localization problem. The implementation is then divided into Scilab-based numerical simulation and Xcos-based dynamic response visualization. The Scilab part performs forward heat conduction, simulated sensor data generation, inverse estimation, fault index localization, and sensitivity analysis. The Xcos part shows the dynamic response for different fault severities.

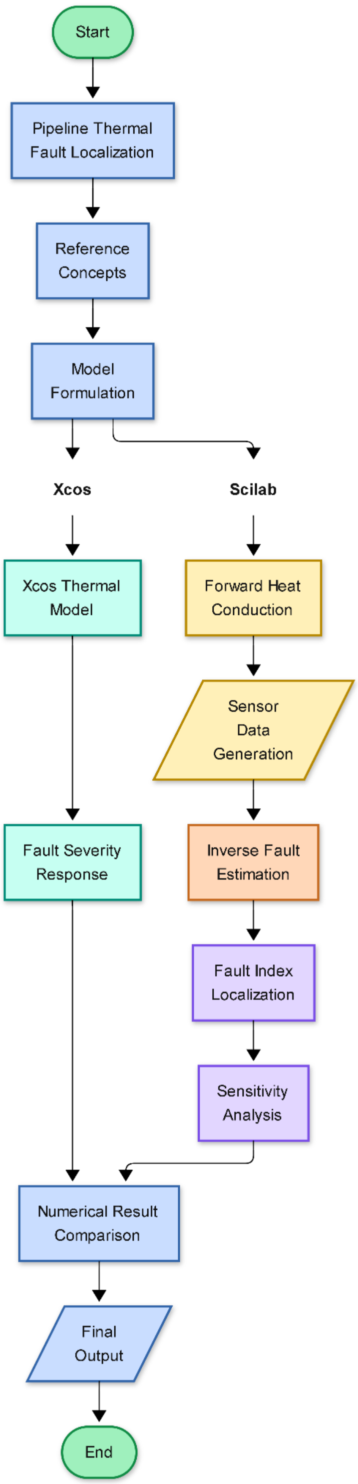

Figure 2.1: Flowchart of the thermal fault localization methodology

2.4 Results and Discussion

The Scilab simulation generates the transient temperature distribution along the pipeline. The pipeline length is taken as 1 m and divided into 201 grid points. The left boundary temperature is fixed at 90°C and the right boundary temperature is fixed at 40°C. The fault is introduced as a localized heat-loss region between 0.45 m and 0.55 m.

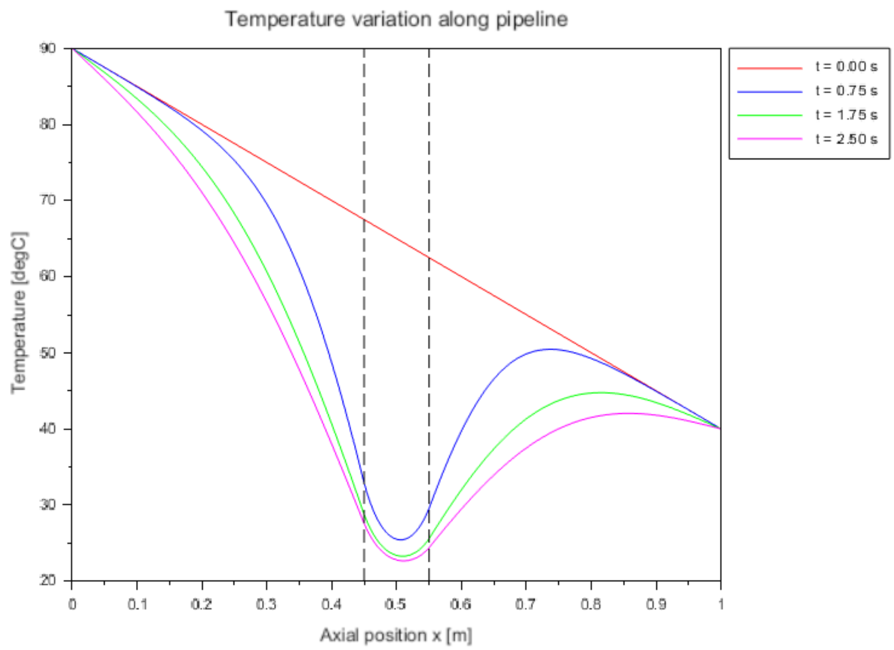

Figure 2.2: Temperature variation along the pipeline

The temperature variation plot shows that the profile develops a clear local drop near the fault zone as the simulation progresses. This occurs because additional heat loss is applied only in the fault region. Hence, the plot confirms that a localized thermal fault produces a visible disturbance in the pipeline temperature distribution.

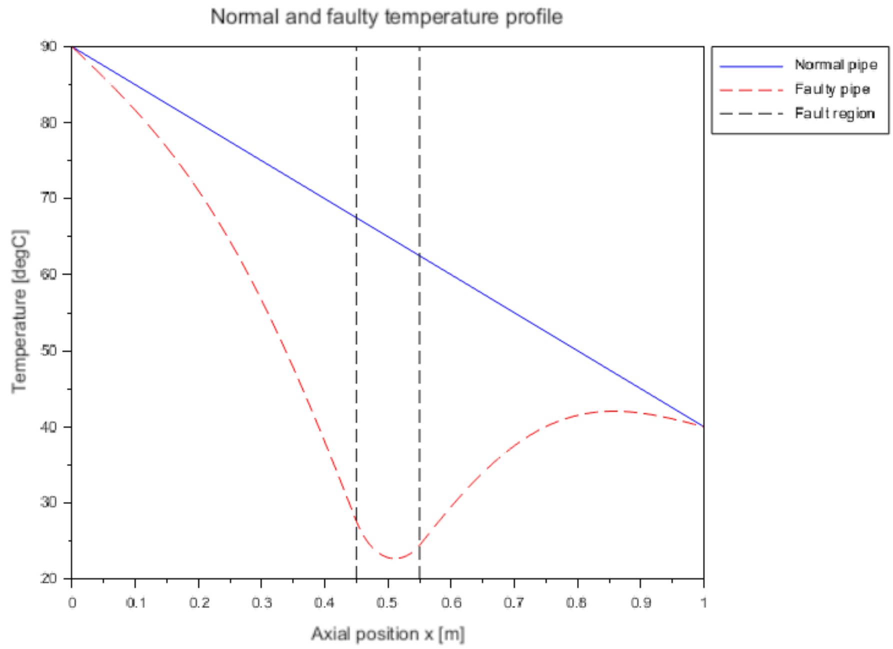

Figure 2.3: Normal and faulty temperature profiles

The normal and faulty temperature profiles are compared to identify the effect of the fault. In the normal case, the temperature varies smoothly from the hot end to the cold end. In the faulty case, a sharp temperature dip appears near the fault region. The normal temperature at the fault centre is 65.00°C, while the faulty temperature at the fault centre is 22.79°C. Therefore, the temperature drop at the fault centre is 42.21°C.

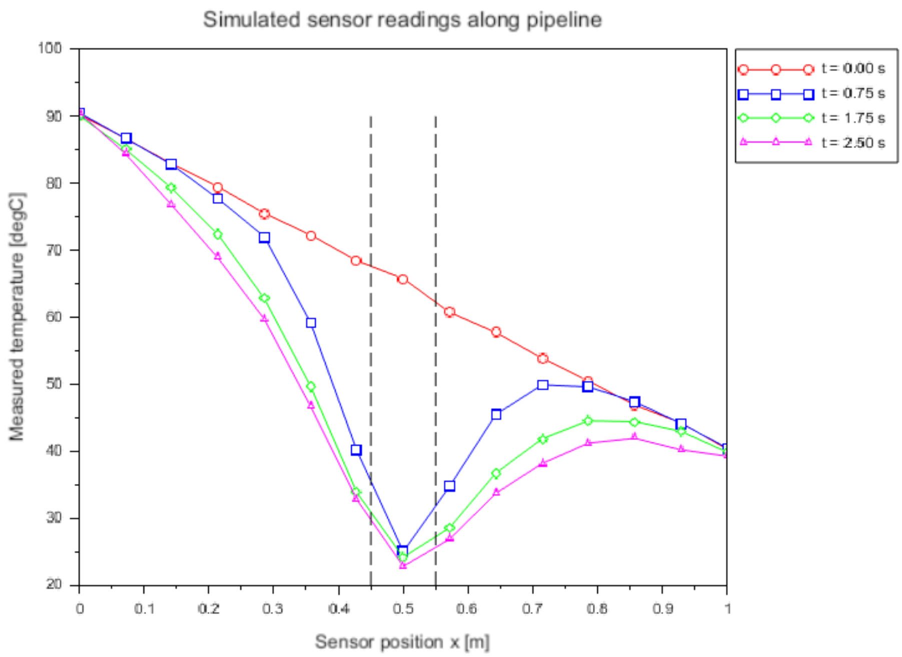

Figure 2.4: Simulated sensor readings along the pipeline

The simulated sensor plot is generated by sampling the temperature profile at selected sensor locations along the pipeline. A small artificial noise value is added to represent measurement-like uncertainty. Even with limited sensor points and small noise, the temperature reduction near the fault region can still be observed.

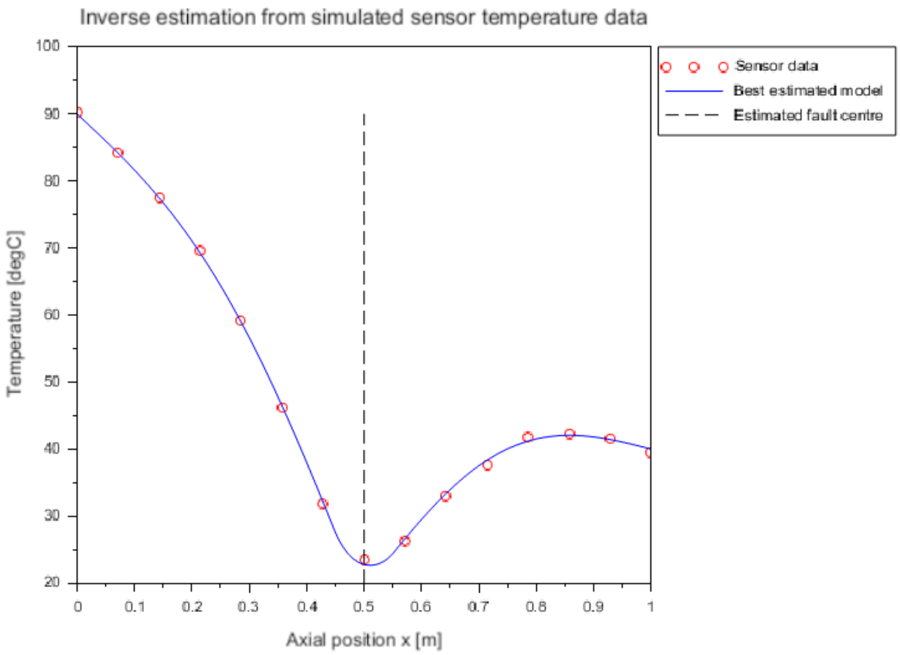

Figure 2.5: Inverse estimation using simulated sensor data

The inverse estimation method compares simulated sensor temperatures with model-generated temperatures for different assumed fault conditions. The case with the minimum error is selected as the estimated fault condition. In the implemented simulation, the inverse estimation predicts the fault centre as 0.500 m, which matches the actual fault centre assigned in the model.

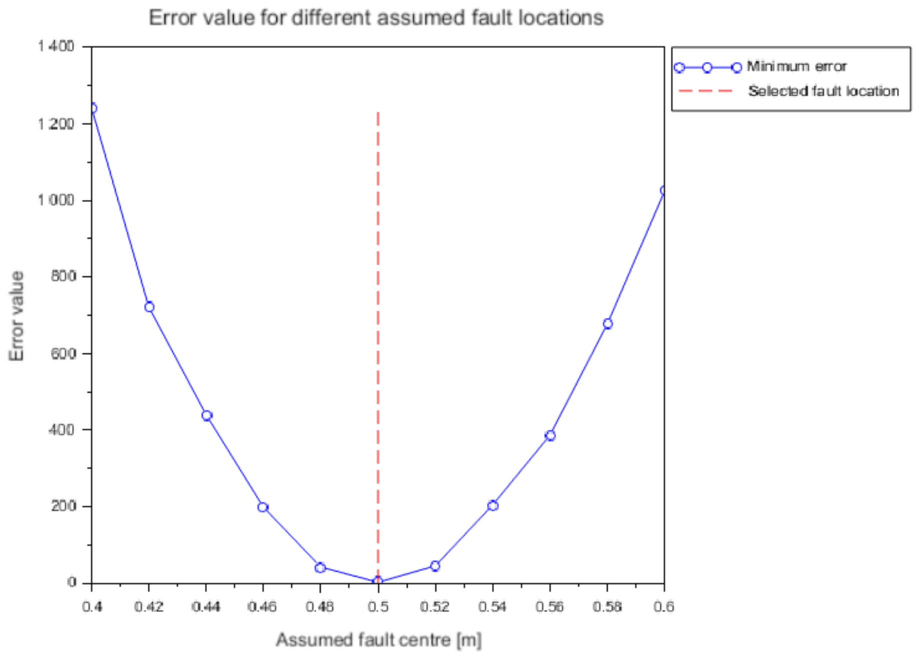

Figure 2.6: Error variation for assumed fault locations

The error variation plot shows how the mismatch between sampled sensor temperatures and simulated model temperatures changes with the assumed fault location. The minimum error point corresponds to the estimated fault location.

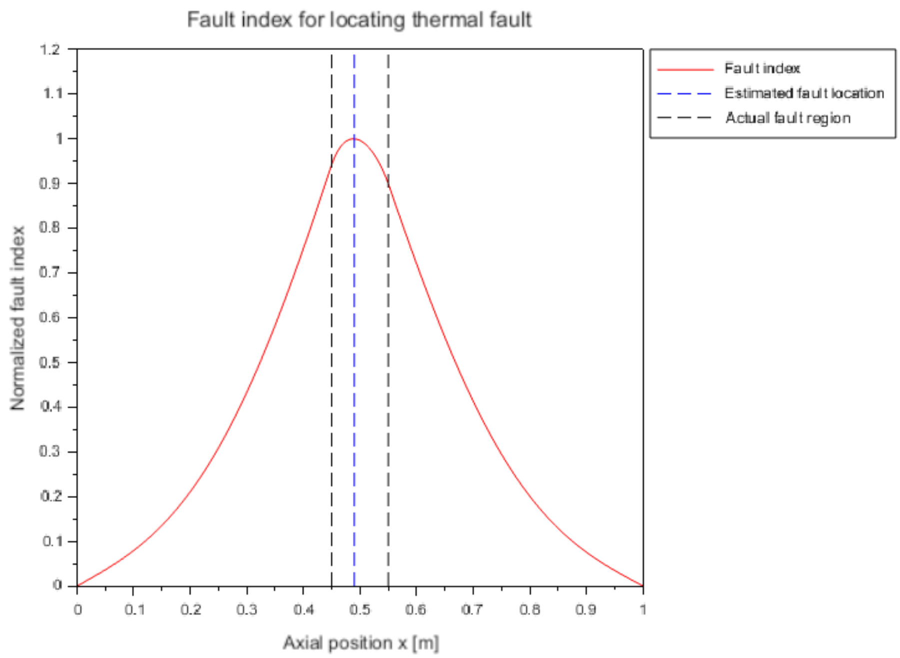

Figure 2.7: Fault index for thermal fault localization

The fault index is calculated using the difference between normal and faulty temperature profiles. The maximum value of the fault index indicates the estimated fault location. In this case, the actual fault centre is 0.500 m and the fault index method estimates the location near 0.490 m, giving an error of 0.010 m.

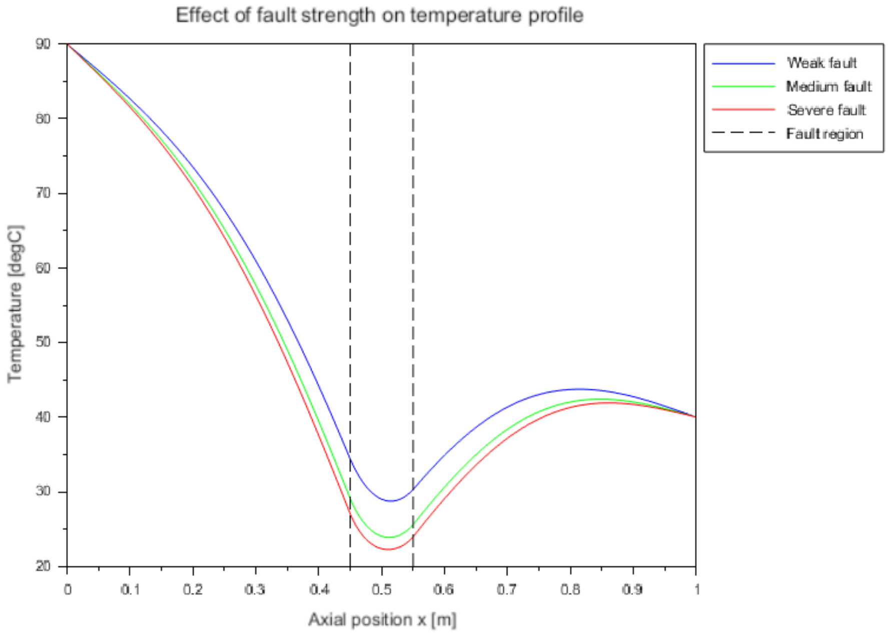

Figure 2.8: Effect of fault strength on temperature profile

The fault strength study is performed for weak, medium, and severe fault cases with fault values of 3, 6, and 9 respectively. As the fault strength increases, the temperature drop near the fault region also increases. This confirms that the model is sensitive to the intensity of the thermal fault.

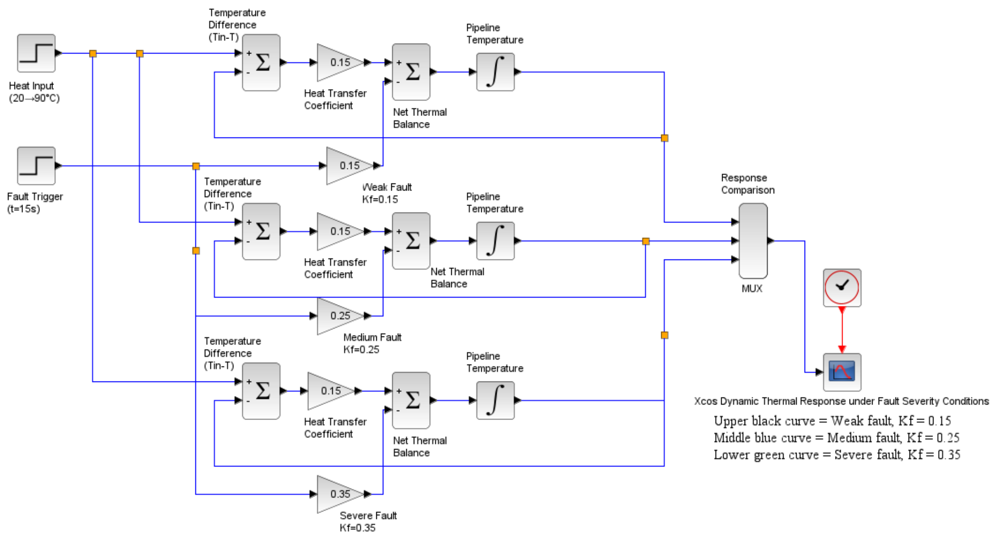

Figure 2.9: Xcos block diagram for dynamic thermal fault response

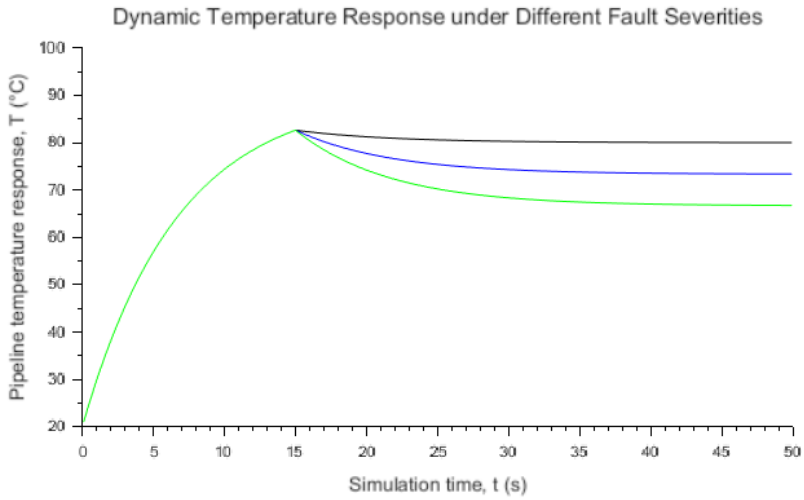

Figure 2.10: Xcos response for different fault severities

The Xcos model visualizes the dynamic thermal response of the system. Before fault activation, the response follows a normal heating trend. After the fault is activated, the response changes depending on fault severity. Weak fault gives a smaller temperature reduction, while severe fault gives a larger temperature reduction.

Table 2.1: Fault localization results

Quantity	**Value**
Fault region considered in the model	0.45 m to 0.55 m
Actual fault centre	0.500 m
Estimated fault centre from inverse estimation	0.500 m
Actual fault value	8.0
Estimated fault value	8.0
Normal temperature at fault centre	65.00°C
Faulty temperature at fault centre	22.79°C
Temperature drop at fault centre	42.21°C
Minimum sensor-model error	2.471
Fault index estimated location	0.490 m
Fault index location error	0.010 m

Table 2.2: Fault strength comparison

Fault Condition	**Fault Value**	**Temperature at Fault Centre**	**Minimum Temperature**
Weak fault	3.0	29.01°C	28.75°C
Medium fault	6.0	24.05°C	23.89°C
Severe fault	9.0	22.38°C	22.27°C

The results show that the thermal fault can be identified by analysing the deviation between normal and faulty temperature profiles. The inverse estimation method correctly identifies the assigned fault centre, and the fault index method locates the fault region with small error.

Chapter 3

Case Study 2: Xcos-Based Implementation of Input Shaping Control for Overhead Crane Swing Reduction

3.1 Abstract

Overhead cranes are widely used for industrial material handling, but the suspended payload may swing like a pendulum during trolley motion. This residual swing reduces positioning accuracy, increases settling time, and may create safety issues. This case study implements input shaping control for overhead crane swing reduction using Scilab and Xcos.

The crane payload is modelled as a simplified pendulum system, where trolley acceleration acts as the input and rope angle represents the swing response. Three input cases are compared: unshaped input, standard Zero Vibration input shaping, and modified Zero Vibration input shaping for non-zero initial swing conditions. Scilab is used to generate acceleration and speed input profiles, while Xcos is used as the main simulation environment for rope angle and rope angular speed responses.

This case study performs a partial replication of the reference IEEE paper, focusing strictly on the Scilab/Xcos time-domain simulation cases. Hardware implementation, phasor diagrams, jerk-limit analysis, and additional parameter studies are kept outside the scope of the present work.

3.2 Problem Formulation

During the operation of an overhead crane, the trolley moves horizontally while the payload is suspended by a rope. When the trolley accelerates or decelerates, the suspended load experiences inertia and begins to swing. This swing is similar to the motion of a pendulum. If the swing remains after the trolley motion is completed, the payload takes more time to settle at the required position. This reduces the efficiency of crane operation and may also affect safety.

In many basic input shaping methods, it is assumed that the payload is initially at rest before the trolley starts moving. However, this condition may not always be true in practical situations. The payload may already have an initial angle or angular velocity due to previous motion, external disturbance, or manual handling. When such non-zero initial conditions are present, a standard input shaper may not suppress the residual swing effectively.

The problem considered in this case study is to reduce the residual swing of an overhead crane payload under non-zero initial conditions. The crane payload is represented using a simplified pendulum-based dynamic model. Trolley acceleration is taken as the input, and rope angle and rope angular speed are taken as the output responses.

The main aim is to implement the simulation part of the reference IEEE paper using Scilab and Xcos, with Xcos as the primary tool for dynamic response simulation.

3.3 Flowchart

The methodology starts with defining the crane and simulation parameters, followed by selecting the simulation case and assigning the corresponding non-zero initial conditions. The input profiles are generated in Scilab and transferred to the Xcos model for dynamic simulation. The rope angle and rope angular speed responses are extracted and compared for unshaped, standard ZV-shaped, and modified ZV-shaped inputs.

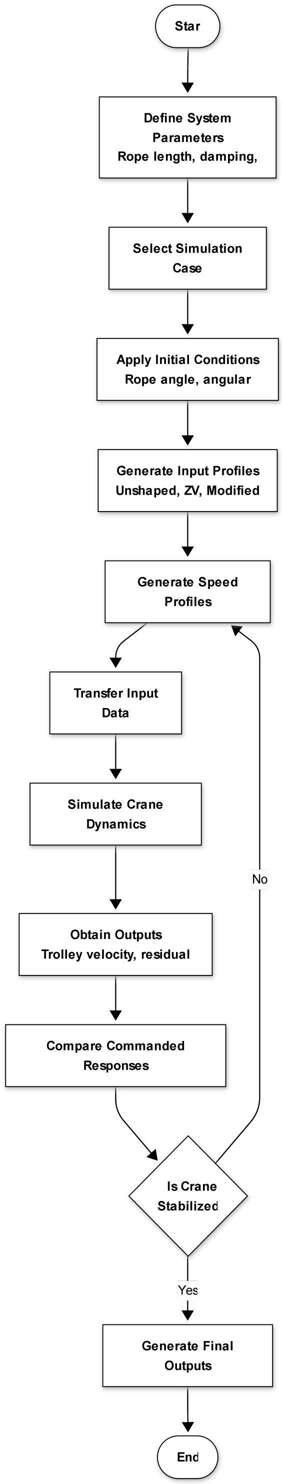

Figure 3.1: Flowchart of the Scilab/Xcos implementation methodology

3.4 Results and Discussion

The important simulation parameters used for the two cases are listed in Table 3.1. The same overhead crane model is used for both cases, but the initial conditions and modified shaper parameters are different.

Table 3.1: Simulation parameters used for Case 1 and Case 2

Parameter	**Case 1**	**Case 2**	**Unit**
Rope length, L	0.51	0.51	m
Acceleration due to gravity, g	9.81	9.81	m/s^2
Initial rope angle, $\phi(0)$	-15	10	degree
Initial rope angular speed, $\dot{\phi}(0)$	50	0	degree/s
Maximum acceleration, a_{max}	2.5	2.5	m/s^2
Maximum velocity, v_{max}	5	5	m/s
Standard ZV amplitude, A_1	0.5	0.5	–
Standard ZV amplitude, A_2	0.5	0.5	–
Standard ZV delay, t_2	0.7163	0.7163	s
Modified ZV amplitude, A_1	0.1683	0.3238	–
Modified ZV amplitude, A_2	0.8317	0.6762	–
Modified ZV delay, t_2	1.0631	0.7559	s

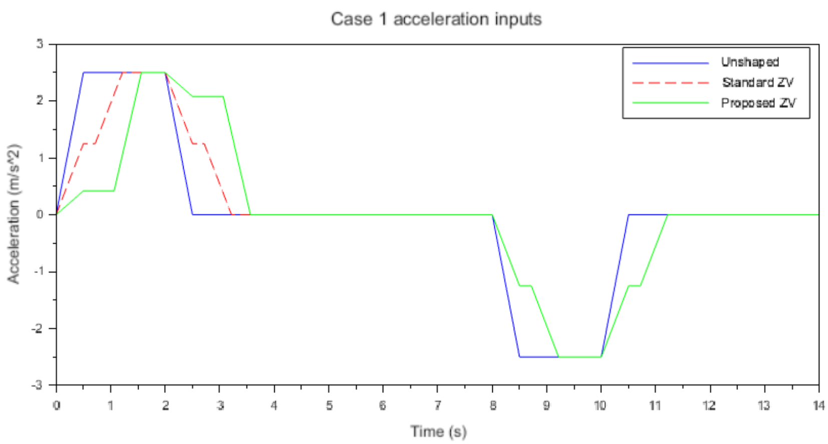

Figure 3.2: Case 1 acceleration input comparison

The Case 1 acceleration plot compares the unshaped input, standard ZV-shaped input, and modified ZV-shaped input. The modified ZV-shaped input uses different amplitude and delay values to account for the initial rope angle and initial rope angular speed.

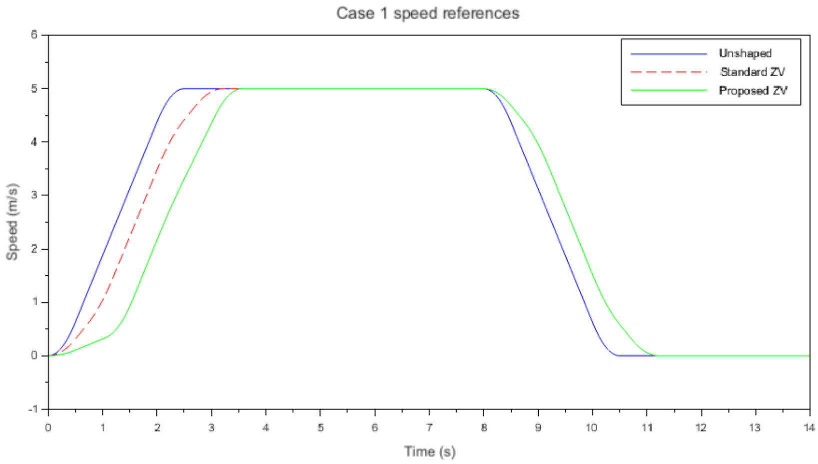

Figure 3.3: Case 1 speed reference comparison

The speed profile is obtained by integrating the corresponding acceleration input. The plot shows that the input shaping process modifies the timing of acceleration and deceleration while maintaining the required trolley motion profile.

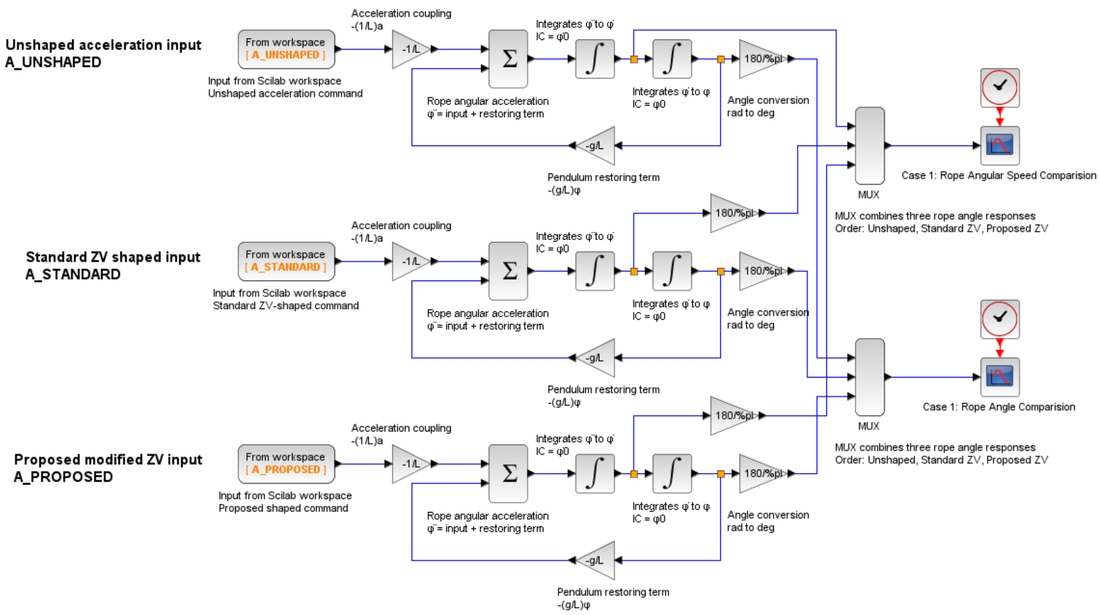

Figure 3.4: Case 1 Xcos model for crane response

The Xcos model represents the pendulum-based crane dynamics. The acceleration input is applied to the crane model, and the rope angle and rope angular speed responses are obtained as output variables.

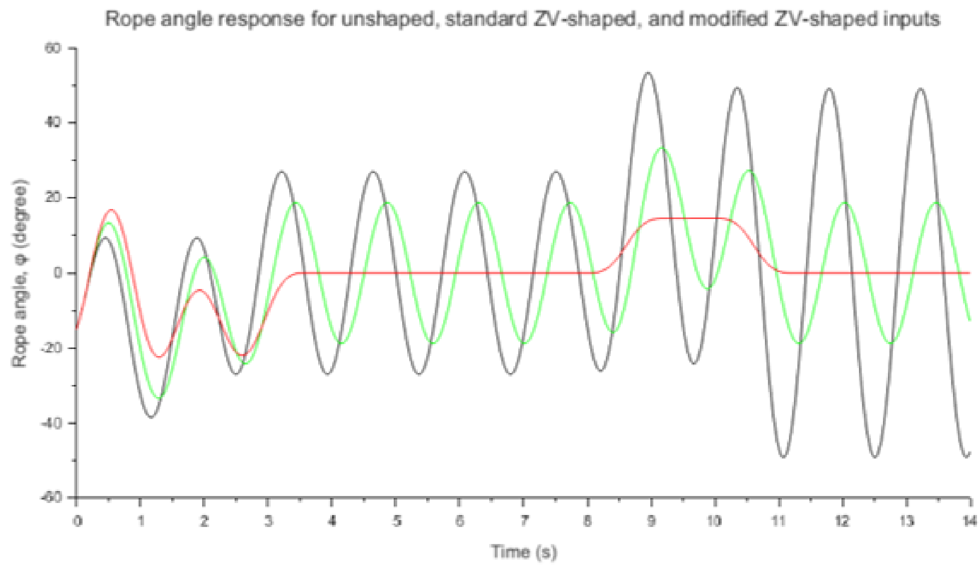

Figure 3.5: Case 1 rope-angle response

The Case 1 rope-angle response shows that the unshaped input produces larger payload oscillations. The standard ZV-shaped input reduces the swing compared with the unshaped case. The modified ZV-shaped input gives the best reduction among the three because it accounts for the non-zero initial swing condition.

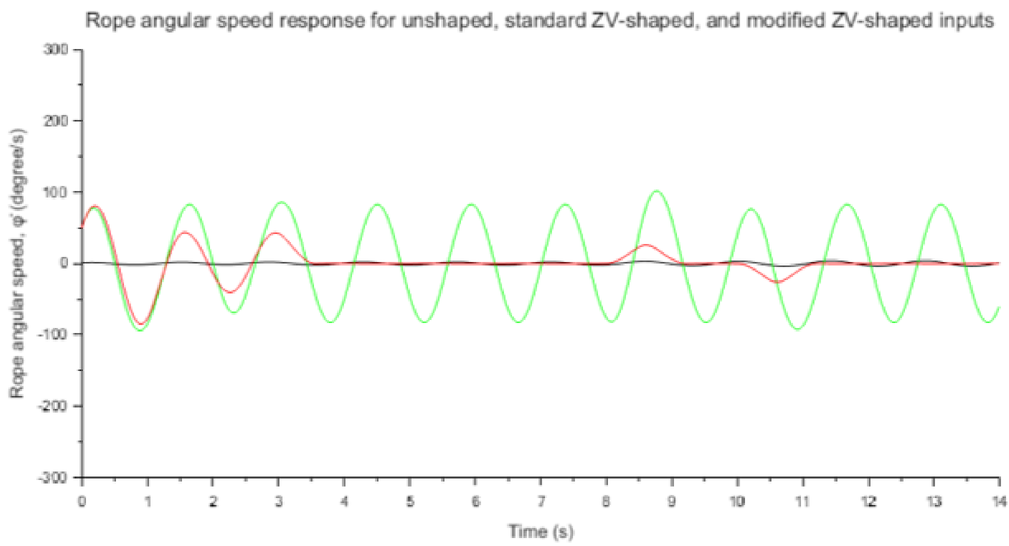

Figure 3.6: Case 1 rope angular-speed response

The Case 1 rope angular-speed response confirms the same trend. The unshaped response shows larger angular speed variations, while the modified ZV-shaped response remains closer to zero for most of the motion.

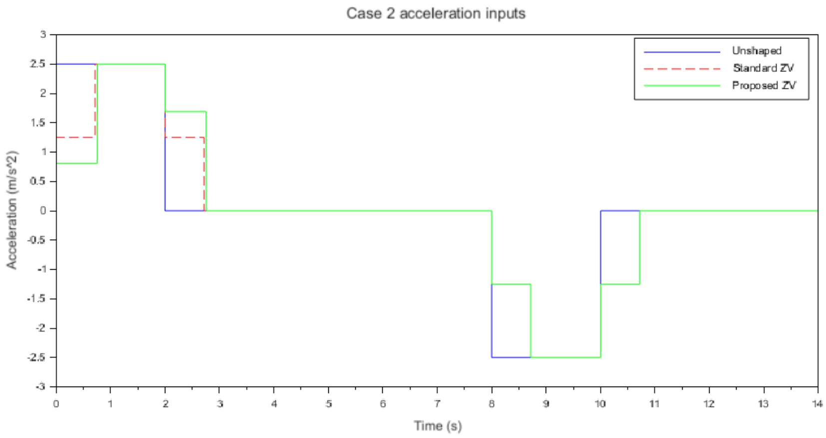

Figure 3.7: Case 2 acceleration input comparison

In Case 2, the initial rope angle is non-zero, while the initial rope angular speed is zero. The acceleration input comparison shows how the modified shaper changes the original command based on the Case 2 initial condition.

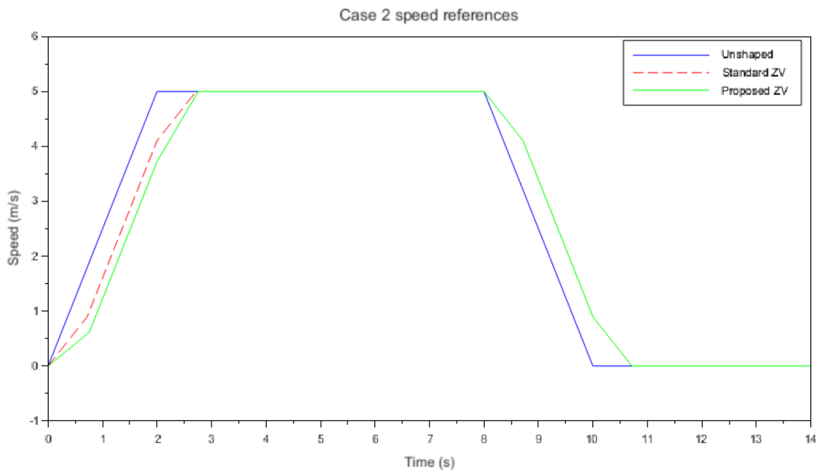

Figure 3.8: Case 2 speed reference comparison

The Case 2 speed profile shows the velocity response generated from the shaped and unshaped acceleration commands. The input shaping method changes the command timing to reduce residual vibration.

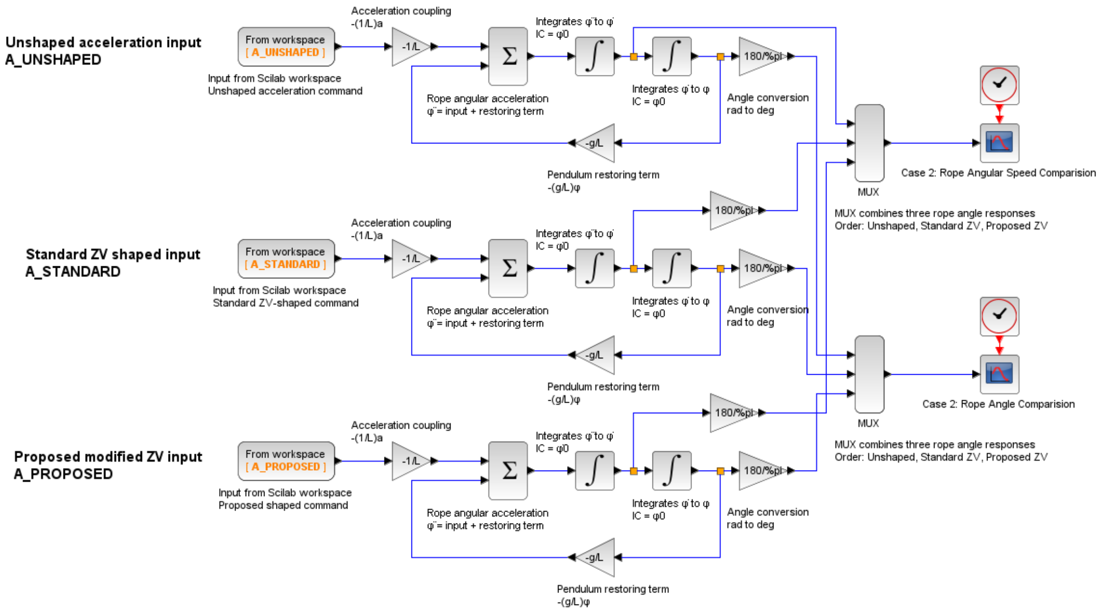

Figure 3.9: Case 2 Xcos model for crane response

The Case 2 Xcos model uses the second set of initial conditions and generated acceleration inputs. The same pendulum-based crane dynamics are simulated to obtain rope angle and angular speed responses.

Figure 3.10: Case 2 rope-angle response

The Case 2 rope-angle response shows that the modified ZV-shaped input reduces residual swing more effectively than the standard ZV-shaped input and the unshaped input.

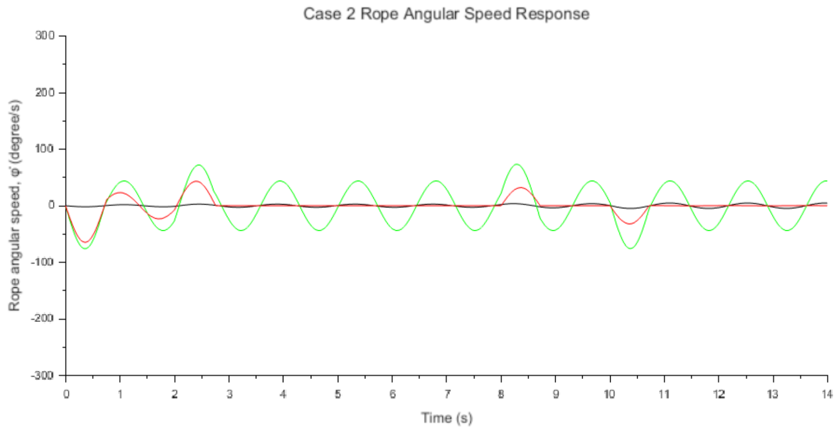

Figure 3.11: Case 2 rope angular-speed response

The Case 2 rope angular-speed response shows that the modified input shaping method reduces the angular speed oscillation more effectively. Overall, for both simulation cases, the modified ZV-shaped input gives the best response among the three input conditions.

Although the reference paper reports zero residual oscillation for the modified ZV shaper, the present Xcos results may still show a small residual oscillation. This difference exists because this case study performs a partial time-domain replication using the reported shaper parameters and a simplified Xcos pendulum model, without reproducing the complete analytical parameter-loop calculation used in the reference paper. Therefore, the modified ZV-shaped response is treated as improved swing reduction, not exact zero residual oscillation.

Chapter 4

Case Study 3: Scilab GUI for Analytical Design and Performance Evaluation of an Automotive Disc Brake

4.1 Abstract

This case study presents the development of a Scilab GUI dashboard for the analytical design and performance evaluation of an automotive disc brake. The work is based on the analytical equations and design guidelines presented in the IEEE paper titled "The Disc Brake Design and Performance Analysis." The dashboard calculates important parameters such as recommended disc-diameter range, disc-thickness check, brake-pad area and radius, hydraulic-cylinder area, clamping force, braking torque, maximum pad pressure, final disc temperature, synchronization coefficient, braking time, and front and rear specific energy dissipation rates.

The analytical model is implemented in a separate Scilab file, while the GUI is used to enter inputs, display results, check the design conditions, and generate performance plots. The implementation is limited to analytical calculations and GUI-based evaluation. Pro/E modelling, ANSYS structural analysis, ANSYS thermal analysis, and experimental validation are not included in the present implementation.

4.2 Problem Formulation

The design of an automotive disc brake involves several interrelated parameters such as disc diameter, disc thickness, brake-pad area, hydraulic pressure, friction coefficient, effective braking radius, braking torque, maximum pad pressure, temperature rise, and energy dissipation rate. These parameters directly affect the braking performance and thermal behaviour of the brake system.

In the selected reference paper, an analytical design procedure is used to calculate the important disc-brake parameters, followed by modelling and analysis using Pro/E and ANSYS. However, carrying out these analytical calculations manually for different input values can be time-consuming and may lead to calculation errors. A fixed numerical calculation also does not clearly show how brake performance changes when parameters such as hydraulic pressure, friction coefficient, disc thickness, initial speed, or rim diameter are varied.

The problem addressed in this case study is to develop a Scilab GUI dashboard that converts the analytical disc-brake design procedure into an interactive calculation tool. The dashboard allows the user to enter vehicle and brake parameters, calculate design and performance outputs, check the selected values based on implemented analytical guidelines, and generate performance plots for parameter study.

4.3 Flowchart

The flowchart represents the working sequence of the developed Scilab GUI dashboard. The GUI program first loads the analytical calculation file, initializes the default values, and displays the dashboard window. The user can either use the default input values taken from the reference paper or modify the input values for another design case. After pressing the calculate button, the program validates the input values and performs the analytical disc-brake calculations.

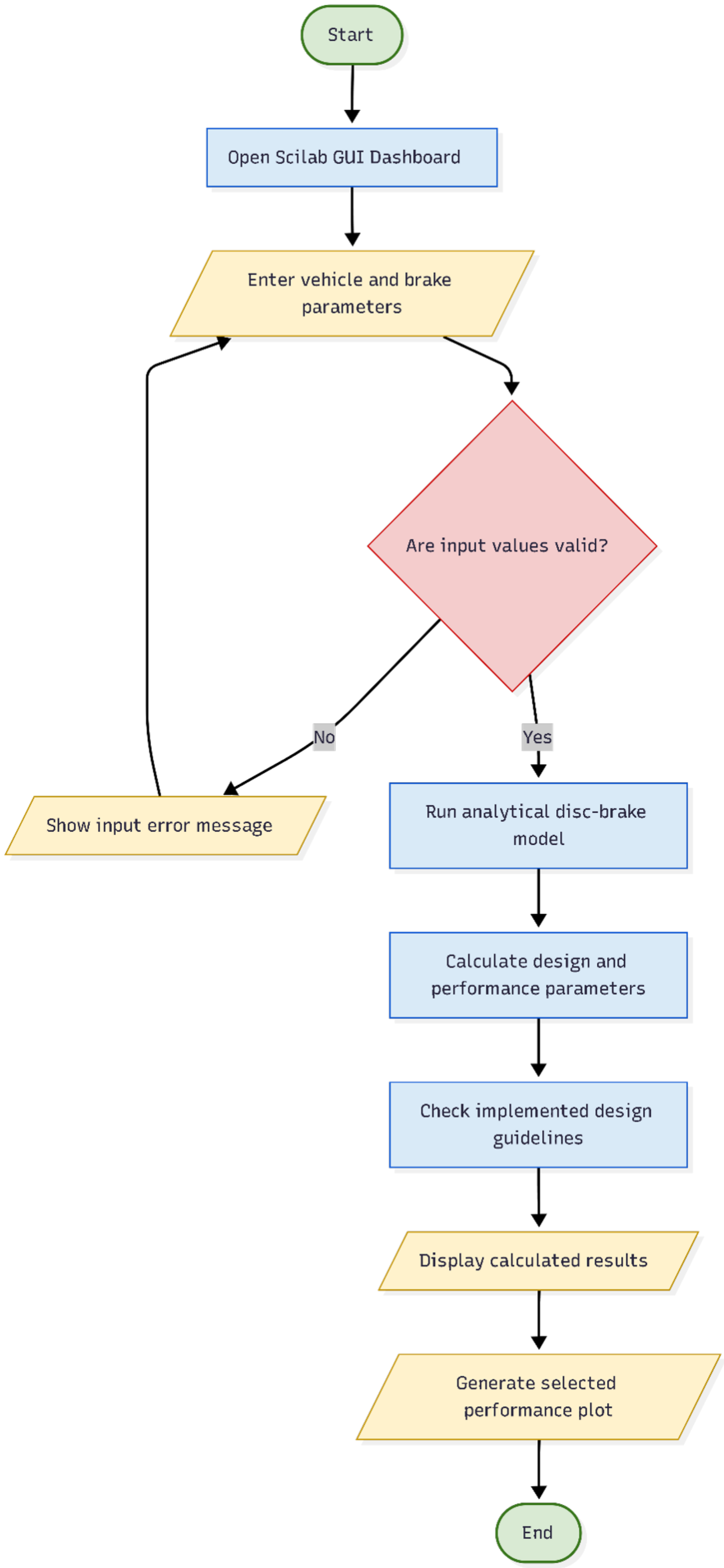

Figure 4.1: Flowchart of the Scilab GUI dashboard

4.4 Results and Discussion

The developed Scilab GUI dashboard was executed using the default input values taken from the reference paper. The dashboard calculates the analytical disc-brake design and performance parameters and displays the results in separate output sections. The output is also compared with the results reported in the reference paper to verify the correctness of the Scilab implementation.

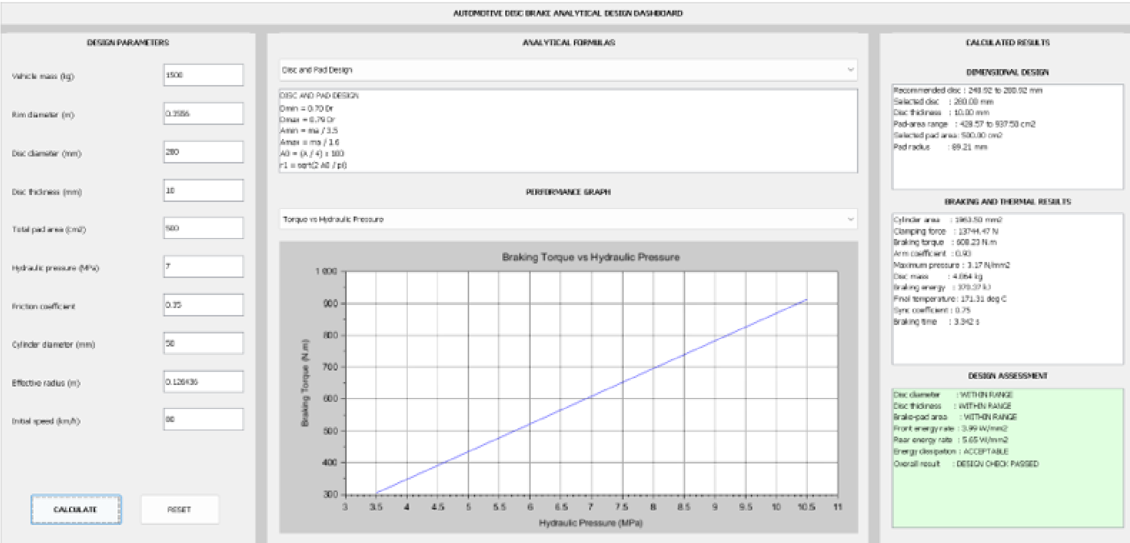

Figure 4.2: Scilab GUI dashboard with default input values

The GUI dashboard consists of three main sections. The left side contains the input parameters such as vehicle mass, rim diameter, disc diameter, disc thickness, brake-pad area, hydraulic pressure, friction coefficient, cylinder diameter, effective radius, and initial speed. The middle section displays the selected analytical formulas and the performance graph. The right side displays the calculated outputs under dimensional design, braking and thermal results, and design assessment.

Table 4.1: Comparison of Scilab GUI results with reference paper results

S. No.	Parameter	Scilab GUI Result	Reference Paper Result
1	Recommended disc-diameter range	248.92 to 280.92 mm	248.9 to 280.9 mm
2	Selected disc diameter	280 mm	280 mm
3	Disc thickness	10 mm	10 mm
4	Brake-pad area range	428.57 to 937.50 cm^2	429 to 936 cm^2
5	Selected brake-pad area	500 cm^2	500 cm^2
6	Equivalent brake-pad radius	89.21 mm	89 mm
7	Effective braking radius	0.126436 m	0.1265 m
8	Braking torque	608.23 N·m	608.23 N·m
9	Pad arm coefficient	0.93	0.93
10	Maximum pad pressure	3.17 N/mm^2	3.17 N/mm^2
11	Final disc temperature	171.31°C	171.3°C
12	Synchronization coefficient	0.75	0.75
13	Front specific energy dissipation rate	3.99 W/mm^2	3.99 W/mm^2
14	Rear specific energy dissipation rate	5.65 W/mm^2	5.65 W/mm^2

The comparison shows that the Scilab GUI results are in close agreement with the analytical results reported in the reference paper. Minor differences in some values are due to rounding of intermediate values and numerical precision used in Scilab.

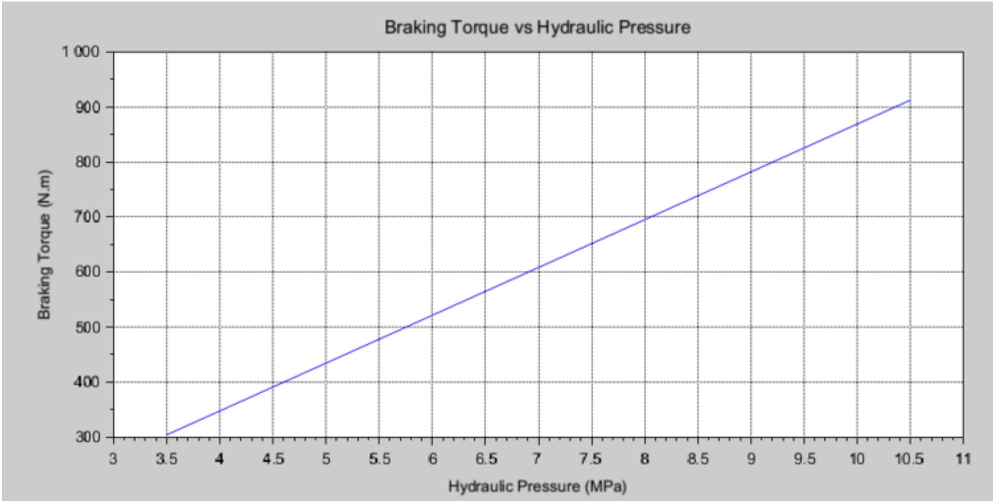

Figure 4.3: Braking torque variation with hydraulic pressure

The braking torque increases linearly with hydraulic pressure. This is because the clamping force is directly proportional to hydraulic pressure, and braking torque depends on clamping force.

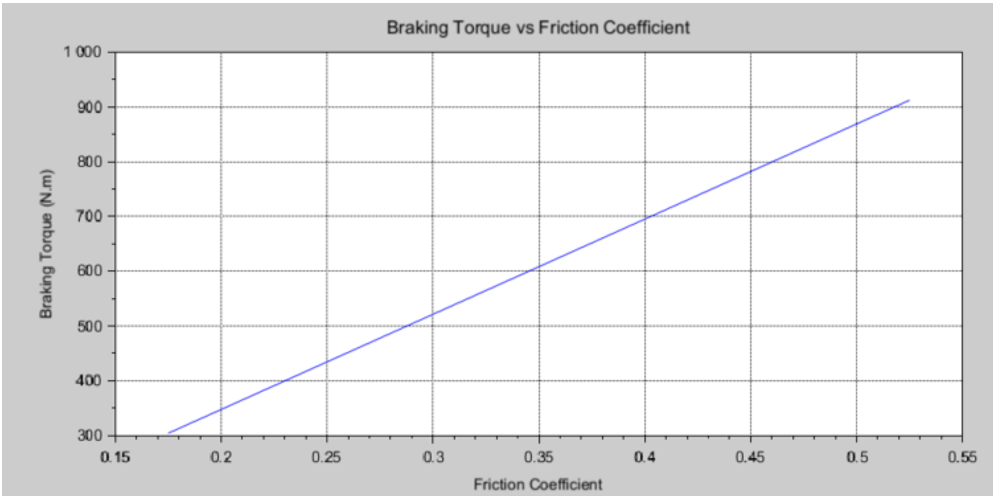

Figure 4.4: Braking torque variation with friction coefficient

The braking torque also increases linearly with the coefficient of friction. For the default friction coefficient of 0.35, the dashboard gives a braking torque of approximately 608.23 N·m.

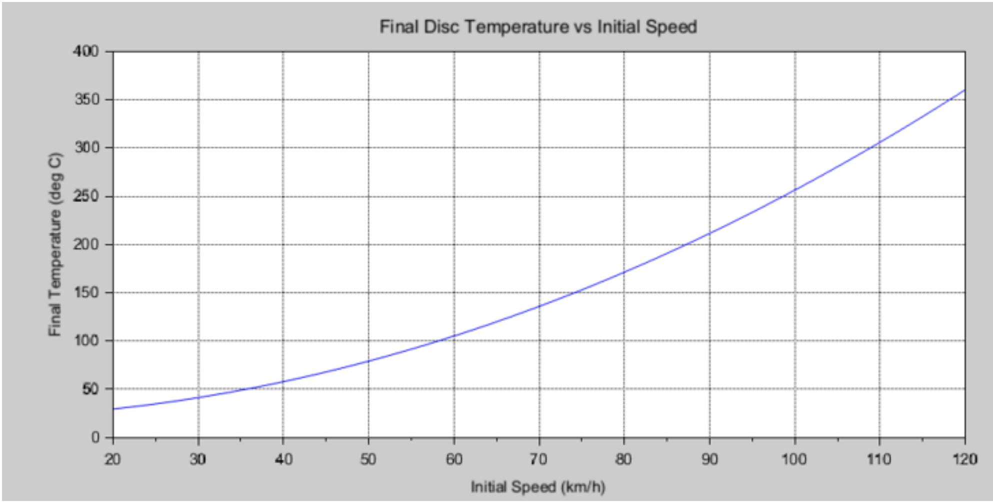

Figure 4.5: Final disc temperature variation with initial speed

The final disc temperature increases rapidly with initial vehicle speed. This trend is nonlinear because braking energy depends on the square of vehicle speed.

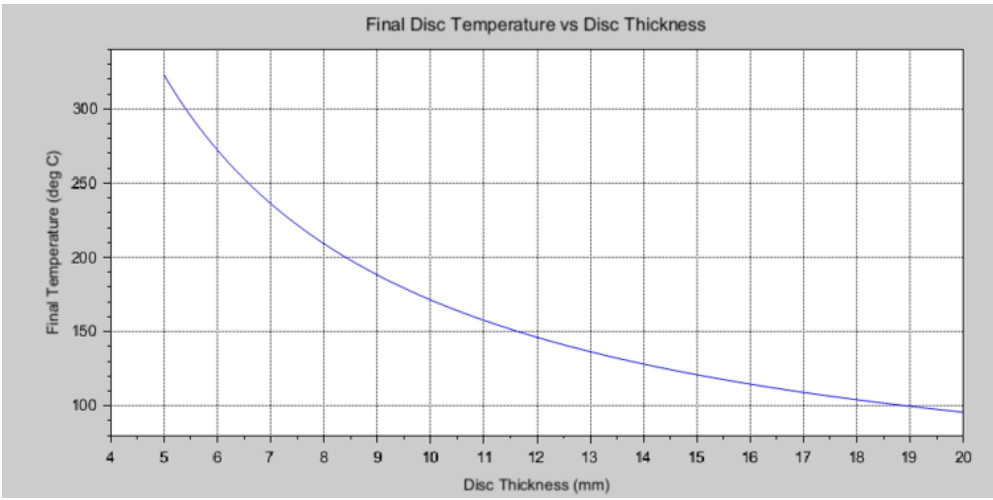

Figure 4.6: Final disc temperature variation with disc thickness

The final disc temperature decreases with increase in disc thickness. This happens because increasing disc thickness increases the mass of the brake disc, allowing it to absorb more braking energy with a lower temperature rise.

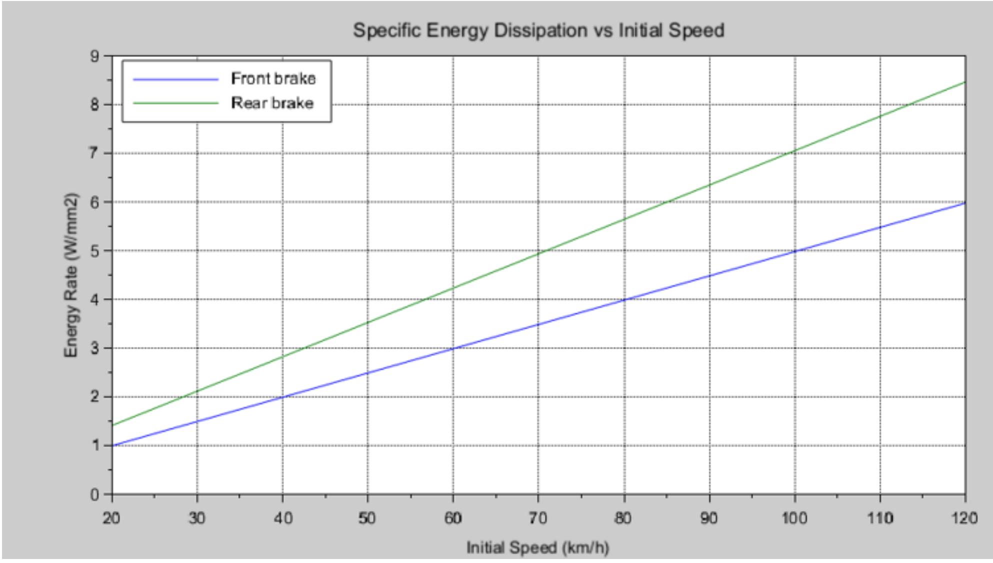

Figure 4.7: Specific energy dissipation variation with initial speed

The front and rear specific energy dissipation rates increase with initial speed. For the default speed of 80 km/h, the front and rear energy dissipation rates are below the implemented limit of 6 W/mm^2.

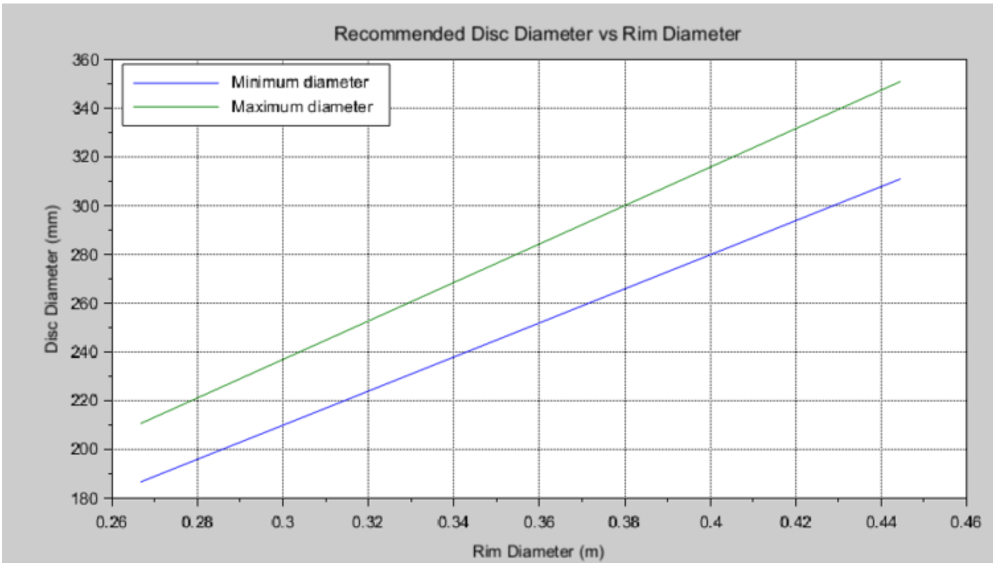

Figure 4.8: Recommended disc diameter variation with rim diameter

The recommended disc-diameter range increases linearly with rim diameter. For the default rim diameter of 0.3556 m, the selected disc diameter of 280 mm lies within the recommended range.

Chapter 5

Learning Outcomes

During this fellowship, I gained practical experience in using Scilab and Xcos for solving engineering problems through simulation, modelling, graphical analysis, and GUI development. The work helped me understand how theoretical concepts can be converted into computational case studies using open-source software.

From the first case study, I learned how a thermal engineering problem can be represented using a one-dimensional heat conduction model. I also understood the role of simulated sensor data, inverse estimation, and fault index methods in identifying a localized thermal fault.

From the second case study, I learned how a mechanical vibration and control problem can be modelled using Xcos. The overhead crane case study helped me understand input shaping, dynamic response simulation, and comparison of unshaped and shaped input commands.

From the third case study, I learned how analytical design calculations can be converted into an interactive Scilab GUI dashboard. This improved my understanding of GUI development, parameter handling, result display, graph generation, and validation of computed results with reference values.

Overall, the fellowship improved my skills in Scilab programming, Xcos modelling, numerical simulation, GUI-based engineering computation, result interpretation, and technical documentation.

Chapter 6

Conclusion

This summer fellowship provided a valuable opportunity to implement mechanical engineering case studies using Scilab and Xcos. Across the three completed case studies, Scilab was used for numerical computation, simulation, GUI development, graph generation, and engineering analysis.

In Case Study 1, thermal fault localization in a pipeline was implemented using a one-dimensional transient heat conduction model. The simulation demonstrated how a localized thermal fault can be identified using the difference between normal and faulty temperature profiles. The inverse estimation method correctly identified the assigned fault centre, and the fault index method located the fault region with small error.

In Case Study 2, input shaping control was implemented for overhead crane swing reduction. The results showed that shaped input commands reduce payload oscillations compared with unshaped commands. Among the considered input cases, the modified ZV-shaped input produced better swing reduction for non-zero initial conditions.

In Case Study 3, a Scilab GUI dashboard was developed for analytical disc-brake design and performance evaluation. The dashboard calculated important design and performance parameters and compared the results with the analytical values reported in the reference paper. The implementation was limited to the analytical part of the paper, while 3D modelling, ANSYS analysis, and experimental validation were outside the scope.

Overall, the three case studies demonstrate the usefulness of Scilab and Xcos as open-source tools for engineering computation, simulation, control system analysis, thermal modelling, and interactive GUI-based applications. The work also highlights how open-source software can support technical education by making engineering analysis more accessible, reproducible, and cost-effective.

Chapter 7

References

[1] "Heat and Mass Transfer: Fundamentals & Applications (4th ed.)," Y. A. Çengel and A. J. Ghajar, McGraw-Hill Education.

[2] "Fundamentals of Heat and Mass Transfer (4th ed.)," C. P. Kothandaraman, New Age International Publishers.

[3] "Hybrid Approach for Solution of Inverse Heat Conduction Problems," I. Felde and W. Shi, IEEE Xplore, 2014. Available: https://ieeexplore.ieee.org/document/6974539

[4] "A Computational Model of Thermal Monitoring at a Leakage in Pipelines," M. A. S. Bhuiyan, M. A. Hossain, and J. M. Alam, International Journal of Heat and Mass Transfer, vol. 92, pp. 330–338, 2016. Available: https://www.sciencedirect.com/science/article/pii/S0017931015009357?via%3Dihub

[5] "Distributed Temperature Sensing: Review of Technology and Applications," A. Ukil, H. Braendle, and P. Krippner, IEEE Sensors Journal, vol. 12, no. 5, pp. 885–892, 2012. Available: https://ieeexplore.ieee.org/document/5955066

[6] "Input Shaping for Non-Zero Initial Conditions and Arbitrary Input Signals with an Application to Overhead Crane Control," IEEE Xplore. Available: https://ieeexplore.ieee.org/document/9729261

[7] "The Disc Brake Design and Performance Analysis," IEEE Xplore. Available: https://ieeexplore.ieee.org/document/5768733

[8] "Scilab Spoken Tutorials," Spoken Tutorials. Available: https://scilab.in/spoken-tutorial