

Optimised State-Dependent Sampling Control for Heavy- Haul Trains

G. Breysens

Master of Science Thesis



Optimised State-Dependent Sampling Control for Heavy-Haul Trains

MASTER OF SCIENCE THESIS

For the double degree of Master of Science in Systems & Control and
Embedded Systems at Delft University of Technology

G. Breysens

April 26, 2023

Faculty of Mechanical, Maritime and Materials Engineering (3mE), and Faculty of Electrical
Engineering, Mathematics and Computer Science (EEMCS) · Delft University of Technology

Cover picture adapted from [29].



Copyright © Delft Center for Systems and Control (DCSC)
All rights reserved.



Abstract

This thesis investigates the potential of state-dependent sampling strategies (SDSS) for the control of heavy-haul trains. Event-triggered control (ETC) is a control approach in which data is only sent when some state-dependent condition, the triggering condition, is satisfied. In this way, the number of communications required to stabilise a system can be drastically reduced. Periodic event-triggered control (PETC) is a variant of ETC in which the triggering condition is checked periodically. By sometimes sampling earlier than a PETC-generated deadline, long-term pay-offs in the average inter-sample times are possible. As searching for formal models of early-triggering controllers quickly becomes computationally infeasible as system dimensions grow, a sample-based approach using reinforcement learning was utilised to find the SDSS controller, which takes the form of a neural network mapping states to the time until the following sample, trained to hopefully yield long-term payoffs in sample efficiency. It was found that, by choosing suitable control parameters, the SDSS controller can outperform the PETC baseline in terms of inter-sample times (IST). Next, a hardware-in-the-loop (HIL) setup was made to evaluate the optimised controller's performance in a real-time context controlling a non-linear train system subject to noise, disturbances, and tasked with attaining different speed setpoints while minimising the inter-wagon forces. It was found that the optimised controller did not perform well during these scenarios. Since the controller's robustness was not considered during the training process, performance suffered under these conditions. To improve the robustness of the controller, the system must be trained on realistic (noisy) data. Simulations with more wagons must be performed to assess whether improvements can still be attained when using larger models. Finally, the accuracy of the HIL setup needs to be improved by using an appropriate network stack that would allow each wagon to send its own data and turn off the sensor node radios when otherwise idly listening for incoming data.

Table of Contents

Acknowledgements	xi
1 Introduction	1
1-1 Background	1
1-2 Problem statement	1
1-3 Objective & Contributions	3
1-4 Thesis overview	4
1-5 Notation	4
2 Preliminaries	7
2-1 Choice of train model	7
2-1-1 Brakes	7
2-1-2 Point mass model	9
2-1-3 Coupler train model	12
2-1-4 Consensus control	18
2-1-5 Final choice of model	21
2-2 Improving sampling efficiency	24
2-2-1 Motivation	24
2-2-2 Mathematical framework	25
2-2-3 Event-triggered control	26
2-2-4 Self-triggered control	32
2-2-5 Reinforcement learning tool	36
3 Simulations	43
3-0-1 Controller design results	43
3-0-2 Evaluation	47

4 HIL Design & Implementation	53
4-1 Hardware	53
4-1-1 Firefly	53
4-1-2 Speedgoat	54
4-2 Communication	54
4-3 System architecture	55
4-4 Physical setup	55
4-5 Experimental setup	58
4-5-1 Experiment configuration	58
4-5-2 Simulink simulation environment	59
5 HIL Experiments	63
5-0-1 Experiment results	63
6 Conclusions and Future Work	67
6-1 Overview	67
6-2 Future work	68
6-2-1 Robustness	68
6-2-2 Scalability	68
6-2-3 Improving the HIL setup	69
A Appendix	71
A-1 RL tool settings	71
A-2 Controller parameters	72
Bibliography	75
Glossary	79
List of Acronyms	79
List of Symbols	80

List of Figures

1-1	BHP Iron Ore train [3].	2
1-2	Railway energy-harvesting device made by Viezo [17].	3
2-1	Overview of braking techniques, exemplifying ECP braking [15].	8
2-2	Force diagram for the train modelled as a point mass [24].	9
2-3	Tractive effort against speed for the GM JT26C-2SS locomotive [24].	10
2-4	Example speed profile [24].	11
2-5	Quantisation leading to excessive switching [9].	11
2-6	Bounds to reduce excessive switching [9].	12
2-7	Force diagram of high speed train in longitudinal motion [38].	13
2-8	Resistances due to slope and curvature angles [16].	15
2-9	Controller architecture with integral compensator [38].	16
2-10	Reference tracking results of the DD and PPD train [38].	17
2-11	Disturbance rejection results of the DD and PPD train [38].	17
2-12	Diagram of EMU [34].	18
2-13	Control framework of the finite-time consensus train controller [34].	19
2-14	Plot of elastic coefficient k_{ij} for $l = 0.5$ m [34].	20
2-15	Periodic (left) and aperiodic (right) sampling [22].	25
2-16	Topology of a set of N networked control systems [1].	25
2-17	Schematic of an event-triggered controller [6].	27
2-18	Example of conic regions for a 2D system [26].	33
2-19	Greedy and long-term optimal strategy example [12].	34
2-20	Comparison of triggering instances for PETC and SDSS ($l = 2$) [20].	36

2-21	Comparison of running average of ISTs for 10 different initial conditions under PETC and SDSS ($l = 2$) [20].	36
2-22	Sample-based optimisation lead to gradual improvements [12].	37
2-23	Inner workings of an artificial neuron [10]	38
2-24	Neural network with an input layer, output layer, and three hidden layers [30]. . .	39
2-25	Schematic to demonstrate the working of action masking [12].	41
2-26	Running average IST simulation, comparing the trained model and PETC baseline. 42	
3-1	Periodic controller simulation.	44
3-2	PETC controller 1 with emphasis on regulating the inter-wagon distances, $k_{\max} = 20$. 45	
3-3	PETC controller 2 with speed emphasis, no max. IST.	46
3-4	PETC controller 3 with speed emphasis, $k_{\max} = 20$	46
3-5	Final controller performance evaluation using 20 random initial conditions.	47
3-6	Deadline of 100,000 uniformly distributed ($[-1, 1]$) random points.	48
3-7	Neural network SDSS deadline distribution before enforcing the PETC deadline. .	48
3-8	Deadline distribution of system with $h = 0.05$ s.	49
3-9	Final performance evaluation of new controller ($h = 0.05$ s) using 20 random initial conditions.	50
3-10	Training progress of new system ($h = 0.05$ s).	50
3-11	Deadline plot for $x_0 = [-0.05 \quad 0.01 \quad -0.05 \quad \Delta x_1 \quad \Delta x_2]^T$ with $(\Delta x_1, \Delta x_2) \in [-0.1, 0.1]$	51
3-12	Deadline plot for $x_0 = [0.1 \quad 0.04 \quad -0.02 \quad \Delta x_1 \quad \Delta x_2]^T$ with $(\Delta x_1, \Delta x_2) \in [-0.1, 0.1]$	51
3-13	AIST during Virtual Serial full experiment.	51
3-14	AIST during Virtual Serial reference tracking experiment with noise.	51
4-1	Zolertia Firefly board [25].	53
4-2	Speedgoat Baseline real-time target machine [33].	54
4-3	HIL setup with Speedgoats running the controller and train model, communicating through USB with the Zolertia Firefly boards that exchange information over UDP. 56	
4-4	Speedgoats front view.	56
4-5	Speedgoats back view.	57
4-6	Ethernet switch in the DCSC lab.	58
4-7	Neural network controller with preprocessing and postprocessing functions. . . .	60
4-9	Simulink setup of the Speedgoat model side.	60
4-8	Simulink setup for Firefly experiment.	61
4-10	Simulink setup of the Speedgoat controller side.	62
5-1	Speedgoat simulation result showing reference tracking, actuation effort, triggers, and deadlines.	64

5-2	Round-trip times of Firefly experiment on computer. Mean = 0.1324s.	66
5-3	Round-trip times of Speedgoat experiment. Mean = 0.0870s.	66
6-1	Improved HIL setup with Speedgoats running the controller and train model, communicating through USB with the Zolertia Firefly boards that exchange information over WCB.	69
6-2	Radio savings made possible by ETC and STC [5].	70

List of Tables

2-1	Train parameter values.	24
3-1	Periodic and PETC implementations evaluation.	47
5-1	Total number of samples.	64
5-2	Number of late samples.	64
5-3	Number of lost samples.	65
5-4	AIST for different experiments.	65
5-5	Load experiment using Virtual Serial.	66
5-6	Load experiment using Firefly communication.	66
A-1	RL experiment settings to find SDSS.	71
A-2	Periodic LQR controller parameters.	73
A-3	Parameters of PETC controller with emphasis on controlling the inter-wagon distances.	73
A-4	Parameters of PETC controller with more speed emphasis.	73
A-5	Parameters of PETC controller with improved deadline distribution.	73

Acknowledgements

I want to thank my supervisor, dr.ir. Manuel Mazo Espinosa, for his assistance during the whole process of the master's thesis. I would also like to thank dr.ir. Gabriel de Albuquerque Gleizer for his role in proposing the project and his helpful support throughout this thesis. In addition, I wish to express my gratitude to the DCSC lab technicians for making it possible to perform experiments and for answering questions about the laboratory equipment. Finally, I would like to thank my parents and friends for their sustained support throughout my studies.

Delft, University of Technology
April 26, 2023

G. Breysens

Chapter 1

Introduction

1-1 Background

A heavy-haul or freight train consists of interconnected goods wagons hauled by one or more locomotives. They are commonly used to transport cargo due to their efficiency. To decrease the costs of this means of cargo transport even further, freight trains are increasingly becoming longer. The BHP Iron Ore currently holds the record with a length of 7.352 km and hauling 682 wagons weighing almost 100,000 tons. Figure 1-1 shows the BHP Iron Ore train. Increasing physical dimensions result in ever-higher demands on technology and control algorithms to guarantee low running costs by reducing travel time, energy consumption, and maintenance costs. Reducing travel time means more trains can run during a specific time window. As described by [40], energy consumption depends on the fuel or electricity consumption of the locomotives and relates to actuation and braking, i.e., the control effort. Maintenance costs relate to wear and tear of the train parts, such as the brakes and couplers between the wagons. Excessive forces on these parts can reduce their expected lifetime, introduce model uncertainties by altering material parameters, and pose safety threats such as the risk of derailment. Another source of energy consumption and maintenance costs is that of the sensors that measure the state of the train, i.e., the velocities and positions of the wagons. These sensors are usually powered by power lines that run throughout the train. A downside of this is that it can make interfacing wagons from different manufacturers cumbersome. A solution in which the wagons communicate wirelessly could eliminate these wires. However, the use of wireless communication raises concerns about the maintenance of the sensors, as they require an energy source such as a battery that needs occasional replacement. This problem will be outlined next, and a proposed solution that forms the motivation of this thesis will be introduced.

1-2 Problem statement

This thesis explores the feasibility of a train controller with wireless communication. The scenario considered is that of a train with low-power sensors on the wagons that can measure



Figure 1-1: BHP Iron Ore train [3].

the speed of the wagon it is attached to and the distance between itself and the subsequent wagon. These sensors could be powered by either a battery or an energy-harvesting device. The more energy the sensors require, the sooner batteries need to be replaced or the higher the energy yield from the energy-harvesting devices needs to be in order to be reliable. It is, therefore, essential to find ways to reduce energy consumption. The sensors' primary energy consumption source is radio utilisation, both listening and transmitting. Thus, a control system that enables radios to turn off as often as possible without compromising performance can substantially save energy and thereby reduce the need for maintenance or help attain a self-sustaining system when powered by energy harvesting. Figure 1-2 shows an example of an energy-harvesting sensor device for the railway industry, made by Viezo. The device works by harvesting the energy of vibrations. In the control architecture of the train considered in this thesis, one wagon will host the controller, receiving incoming sensor data and sending control signals to the wagons to regulate the speeds and inter-wagon distances of the wagons. Controllers are usually implemented in a continuous or periodic fashion. Continuous-time controllers can be constructed from analog components such as capacitors, resistors, and operational amplifiers. With the advent of the digital age, analog control implementations have fallen out of favour in most but a few niche applications due to their comparative difficulty in tuning, sensitivity to environmental conditions, and lack of adaptability when the implementation needs to change. Digital controllers are programmable, less costly, and easier to design. Most digital controllers work periodically. In their interaction with the outside world through sensors and actuators, they use communication protocols with limited data rates. Due to the periodic nature of the controller, commands are sent at a fixed rate, irrespective of the state of the system. Hence, even if the system is in equilibrium, the communication channel will be wastefully occupied by the sensors and controller exchanging information. A trivial manner to solve this is by decreasing the sampling frequency. This option is not always feasible due to the risk of instability, and sacrifices responsiveness. Instead, an aperiodic control strategy that sends control updates when the system requires attention while still guaranteeing a specified performance is a more intuitive way to mitigate superfluous updates. This strategy will be explored in Section 2-2.



Figure 1-2: Railway energy-harvesting device made by Viezo [17].

1-3 Objective & Contributions

Concretely, this thesis aims to achieve the following:

- **Objective:**

1. Implement a HIL setup to perform real-time simulations of a train and controller using wireless communication, demonstrating the usage of an aperiodic control mechanism for this system.
2. Implement progressively more realistic simulations that can run on a non-real-time computer to approximate a real-time simulation and observe the consequences of increasingly accurate system conditions.

- **Requirements:**

1. The train model must strike a balance between realism and easy-of-use.
2. The train parameters must be realistic.
3. The controller must be able to control the train around several setpoints.
4. The controller must demonstrate improved communication efficiency.
5. The controller must keep actuator efforts within reasonable bounds.

- **Evaluation:**

1. The system will be qualitatively evaluated by observing whether it is stabilised around different setpoints and observing the effect of noise and disturbances.
2. The system will be quantitatively evaluated by measuring sampling and communication metrics.

3. The real-time simulation results will be compared to the computer simulations.

The contributions of this thesis are the following:

- The controller design for a train model is described. The controller aims to minimise the number of communications necessary to guarantee a given performance.
- A HIL setup is made to simulate the system. The setup uses Speedgoat Baseline real-time target machines for real-time simulation and Zolertia Firefly boards for wireless communication.
- Experimental results are obtained that study the effect of increasing system realism.
- Suggestions are made for future work.

1-4 Thesis overview

This chapter aimed to situate the problem of designing an aperiodic controller for a train model in the context of removing physical wiring from the train and switching to a wireless control infrastructure in which minimising the number of control updates is paramount to save energy. Chapter 2 goes over the choice of a train model and type of controller by summarising the relevant literature and doubling as a stepping stone to the eventual design choices. It outlines the advantages and limitations of the train models and control strategies encountered in the literature. The theory is analysed in sufficient detail to allow the reader to understand the main points that make a comparison of different methods possible. Sometimes, small mathematical derivations or theorems that could be found elsewhere are nevertheless repeated here to create a reasoned document that stands on its own. The reader is referred to the cited works where details distract from the essence. Chapter 3 implements a periodic controller from the chosen model with realistic train parameters and compares it to PETC implementations. A SDSS is found that uses the PETC-generated sampling times as a deadline. The SDSS controller is evaluated, and its current limitations are highlighted. Chapter 4 goes over the design and implementation of the real-time system. It presents the hardware that will be used to implement the proposed system architecture and goes over implementation details germane to the design. Finally, it describes the experimental setup, detailing the different experiment types and settings that can be selected. Chapter 5 presents the results of the HIL experiments. Chapter 6 summarises the work done for this thesis and the results obtained from the experiments. It outlines several important avenues that merit further interest if the controller design proposed in this thesis is to scale to include more wagons.

1-5 Notation

\mathbb{N} is the set of positive integers. \mathbb{R} is the set of real numbers, \mathbb{R}^+ is the set of positive real numbers, and $\mathbb{R}^{\geq 0}$ is the set of positive real numbers including zero. An empty set is represented by \emptyset . For a matrix $A \in \mathbb{R}^{n \times m}$, we denote its transpose by A^T . For a symmetric square matrix $P \in \mathbb{R}^{n \times n}$, we denote positive (semi-)definiteness by $P \succ 0$ ($P \succcurlyeq 0$) and

negative (semi-)definiteness by $P \prec 0$ ($P \preceq 0$). A function $\beta : \mathbb{R}_0^+ \rightarrow \mathbb{R}_0^+$ is said to be a \mathcal{K} -function if it is continuous, strictly increasing and $\beta(0) = 0$. It is said to be a \mathcal{K}_∞ -function if it is a \mathcal{K} -function and $\beta(s) \rightarrow \infty$ as $s \rightarrow \infty$. A function $f : X \rightarrow Y$ is Lipschitz continuous on $S \subseteq X$ if there exists a positive real constant $L > 0$ such that for each $x_1, x_2 \in S$ we have that $|f(x_1) - f(x_2)| \leq L|x_1 - x_2|$. For a locally integrable signal $w : \mathbb{R}_0^+ \rightarrow \mathbb{R}^{n_w}$, we denote by $\|w\|_{\mathcal{L}_2} = (\int_0^\infty \|w(t)\|^2 dt)^{1/2}$ its \mathcal{L}_2 -norm if the integral is finite. The system $\dot{x}(t) = Ax(t) + B\hat{u}(t) + E\delta(t)$ is said to be globally exponentially stable (GES) if there exist $\sigma \in \mathbb{R}^+$ and $\rho \in \mathbb{R}^+$, such that for any $x(0) = x_0 \in \mathbb{R}^{n_x}$ and $\delta \equiv 0$ all corresponding solutions to this differential equation satisfy: $|x(t)| \leq \sigma|x_0|e^{-\rho t}$ for all $t \in \mathbb{R}_0^+$.

Preliminaries

2-1 Choice of train model

This section describes several full train models or relevant aspects to their modelling, as found in the literature. The advantages and disadvantages of each model will be outlined. The choice of train model impacts the controller architecture. A final train model is chosen that will be used to design a controller. The limitations of this model are described. The choice of train model is usually made with a controller type in mind. Therefore, this section also covers the controller type accompanying the surveyed models. The controller design is more thoroughly examined in Section 2-2.

2-1-1 Brakes

Traditional train brakes use air pressure signals to propagate the brake command from the front to the back of the train. As the signal propagates at the speed of sound, there will be a delay before all wagons have received their braking command. The effect of this delay is that the front wagons will already be braking before the rear ones, resulting in pushing behaviour that leads to large coupler forces [31]. To avoid this problem, electronically controlled pneumatic (ECP) brakes have been developed. ECP uses electronic brake signals to activate the air-powered brakes on the wagons. These signals can either be sent through a dedicated wire running along the train's length or by radio communication. Trains equipped with ECP brakes can control the braking of individual wagons. The communication medium is also used to transmit sensor and diagnostics information. Wabtec's report on the ECP system mentions the following five advantages of ECP braking [27]:

1. Reduced wheel and brake shoe wear, from even distribution of braking and better pressure control.
2. Reduced fuel consumption, from the use of graduated release capability to eliminate power braking.

3. Increased safety from shorter stopping distances (typically over 50% reduction).
4. Reduced in-train forces, wear on equipment, and derailment risks.
5. Car system alarm reporting and diagnostics to aid maintenance.

Wabtec's report also mentions cable-based distributed power control for the end-of-train locomotives. This is an example of an independent distributed power (iDP) system, in which locomotives can be placed in multiple locations in a train. In a post on Next Generation Distributed Power published by Wabtec, the authors state that distributed power "reduces in-train forces by distributing traction and braking forces along the train. The reduction in lateral forces and friction lowers the impact of heavy trains on track infrastructure, provides better adhesion, and increases fuel efficiency" [14]. With the availability of these technologies, the following three train operation strategies are considered in [39]:

- 1-1 strategy: There is one control signal for all the locomotives and one braking control signal for all the wagons.
- 2-1 strategy: The control signal of every locomotive effort may be different, and the braking control signal of all the wagons is the same. This is an iDP-only strategy.
- 2-2 strategy: Every car has an independent control command, including locomotives and wagons. This is a fully ECP/iDP mode.

The paper mentioned above concluded that the 2-2 strategy was the best for the closed-loop controller. Figure 2-1 shows an overview of the braking techniques and the benefits of ECP. The goal is to remove the ECP wire and replace it with wireless communication. Nevertheless, bandwidth issues remain even if the ECP wire is retained.

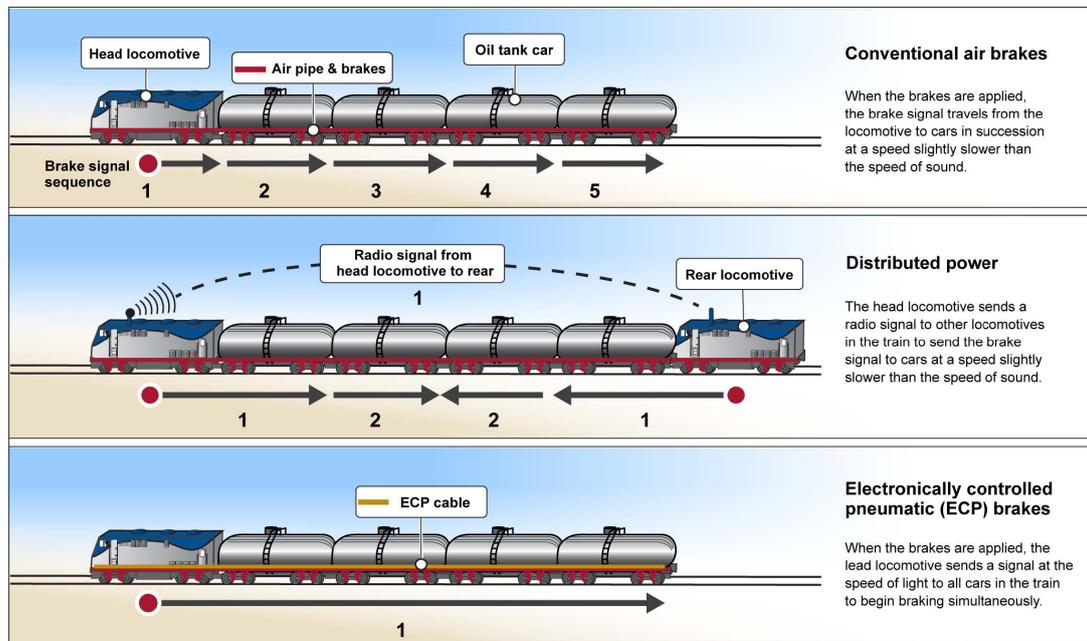


Figure 2-1: Overview of braking techniques, exemplifying ECP braking [15].

Discussion

In a heavy-haul train, it is possible to have multiple locomotives with several unpowered wagons in between. However, the brake and traction will usually be separate signals. In a heavy-haul train, unpowered wagons do not have traction capacity, but could be equipped with brakes. Such a configuration would make the system non-linear, requiring controllers to consider the constraints this places on actuation. To avoid this, wagons can be regarded as uncontrollable masses hauled by locomotives. While more straightforward from a control design perspective, using such a configuration decreases control performance, leading to higher inter-wagon forces and slower braking.

2-1-2 Point mass model

Howlett addresses the problem of a train travelling from one station to the next along a track with a non-constant gradient within a given time while minimising energy consumption [24]. Howlett models the train as a point mass, as shown in Figure 2-2. The train dynamics are

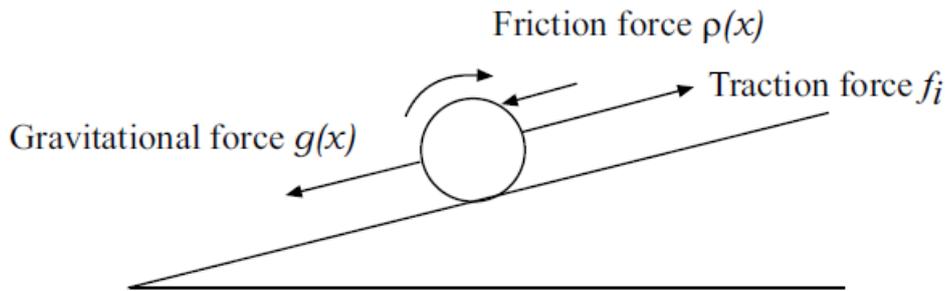


Figure 2-2: Force diagram for the train modelled as a point mass [24].

governed by the following equations.

$$\begin{aligned}\frac{dx}{dt} &= v, \\ \frac{dv}{dt} &= \frac{Hf_i}{v} + K_j - \rho(x) + g(x),\end{aligned}$$

where x is the distance along the track, v is the speed of the train, $\rho(v)$ is the resistive acceleration due to friction, $g(x)$ is the gravitational acceleration due to the track gradient, H is a constant from the traction graph, f_i is the traction force, and K_j is the braking acceleration. Howlett's design considers a diesel-electric locomotive with discrete traction control settings that determine a constant fuel supply rate. Figure 2-3 shows the tractive effort against speed for the GM JT26C-2SS locomotive. Similar graphs exist for other locomotives. A constraint on the number of locomotive traction settings corresponds to the most common scenario for diesel-electric locomotives, which typically have eight settings called notches. A brief sketch of the control solution will now be provided. The reader is referred to the original paper for further details. Consider a control variable j , which can take the values of the finite set

$$\mathcal{C} = \{-1, 0, 1, 2, \dots, m\}, \quad (2-1)$$

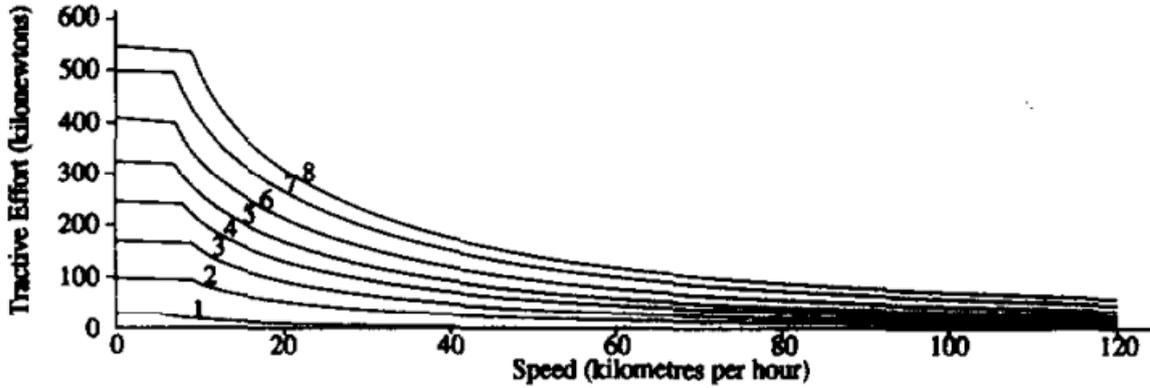


Figure 2-3: Tractive effort against speed for the GM JT26C-2SS locomotive [24].

where the non-negative elements denote a traction setting and -1 denotes braking. The set of possible control sequences is given by the subset

$$\mathcal{S}(\{j(k+1)\}_{k=0,1,\dots,n}).$$

Let f_j be the fuel supply rate corresponding to the control setting j . We have that

$$0 = f_{-1} = f_0 < f_1 < f_2 < \dots < f_m = 1.$$

The control strategy works by considering several points along the train's trajectory. The total distance is divided into segments as follows.

$$0 = x_0 \leq x_1 \leq x_2 \leq x_3 \leq \dots \leq x_{n+1},$$

where x_0 is the initial point and x_{n+1} , $\{x_k\}_{k=1,2,\dots,n}$ the switching points, x_{n+1} the final destination, and $j(k+1)$ is the control setting on the interval (x_k, x_{k+1}) . We define ξ_{k+1} as the length of the interval (x_k, x_{k+1}) . The control strategy with control actions as in Equation 2-1 and switching points as in Equation 2-1-2 is given by

$$S(\{[j(k+1); (x_k, x_{k+1})]\}_{k=0,1,\dots,n}).$$

Let τ_{k+1} denote the time it takes to traverse the interval (x_k, x_{k+1}) , then the total fuel consumption is given by

$$J = \sum_{k=0}^n f_{j(k+1)} \tau_{k+1}.$$

Let X represent the total journey length, and let T represent the allowed journey time to traverse this distance. The problem Howlett addresses is that of finding a control strategy

$$S(\{[j(k+1); (x_k, x_{k+1})]\}_{k=0,1,\dots,n}) \in \mathcal{S}(\{j(k+1)\}_{k=0,1,\dots,n}),$$

with $v(x_0) = 0$, $v(x_{n+1}) = 0$, $\sum_{k=0}^n \tau_{k+1} = T$, and $\sum_{k=0}^n \xi_{k+1} = X$ that minimises J . The track gradient data is assumed to be stored in a digital format. A commercial device under the name *Metromiser* solves the optimisation problem and advises the train driver on traction control. An example trajectory is given in Figure 2-4.

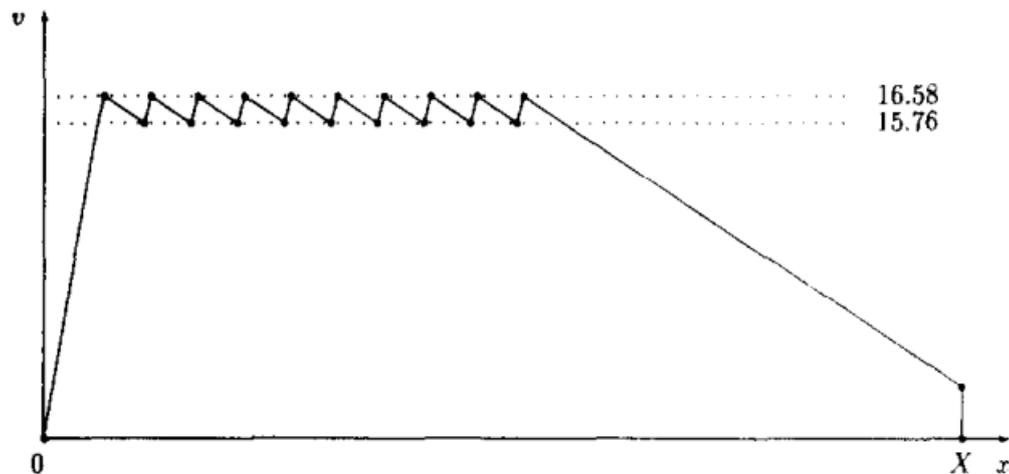


Figure 2-4: Example speed profile [24].

Discussion

A significant limitation in Howlett's work is the modelling of the train as a point mass. For long freight trains, this is inaccurate and neglects the importance of the in-train forces. Furthermore, guidance systems such as *Metronome* work in open-loop, using a predetermined sequence of control updates optimised for a specific track. It thus lacks the robustness, responsiveness, and general applicability that could be expected from a controller and which closed-loop control aims to provide. A positive aspect of Howlett's model is that it includes the limits on actuation freedom as seen in real diesel-electric locomotives. Several proposed train models, some of which will be discussed later, ignore limits on actuation precision. While inaccurate for diesel-electric locomotives, this is not a problem for electric locomotives, which have no notches and can therefore provide any desired traction within the locomotive's limits. The continuous force input must be converted to a corresponding notch input if notches are used. Figure 2-5 shows an example of this quantisation. A significant performance

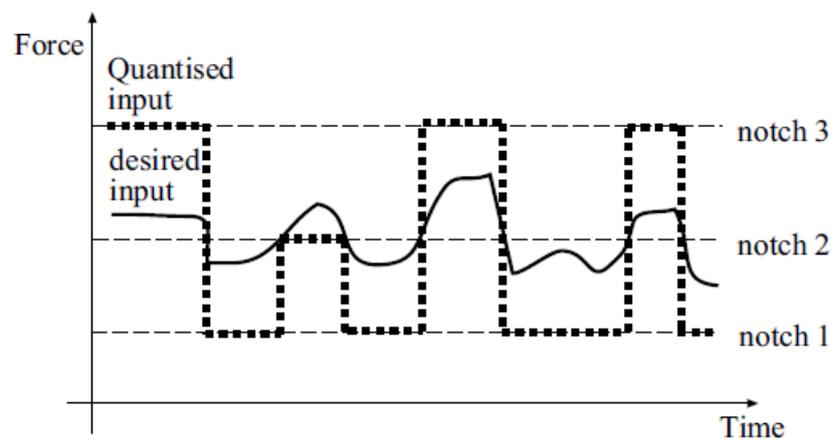


Figure 2-5: Quantisation leading to excessive switching [9].

problem with quantisation is that excessive switching between notches originating from minor

differences in the desired force inputs can occur. This excessive switching can result in large in-train forces. Therefore, bounds can be specified so that no notch switch occurs if these bounds are not exceeded. This is illustrated in Figure 2-6.

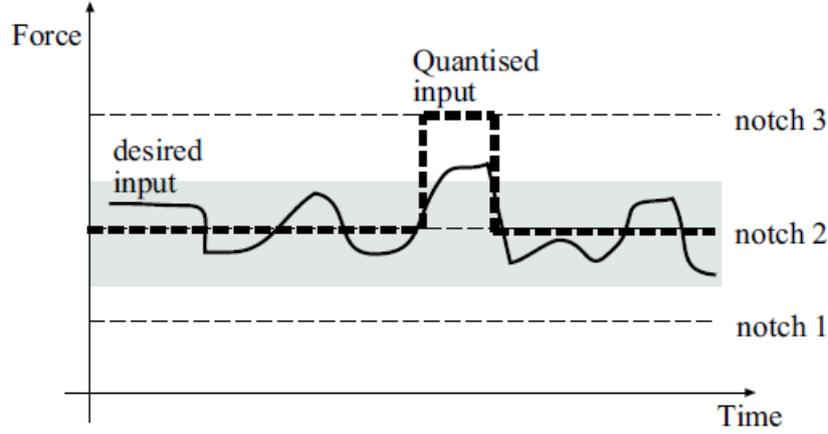


Figure 2-6: Bounds to reduce excessive switching [9].

2-1-3 Coupler train model

The previous train model assumed the train could be modelled as a rigid body. This ignores the effects of the couplers between the wagons, the velocities of individual wagons, and the in-train forces. For a high-speed train, two designs are common: push-pull driving (PPD) and distributed driving (DD) [38]. In PPD, only the front and rear wagons are locomotives. In DD, all wagons are motorised. The model derivations that will now be presented were summarised by [9].

Multi-body dynamics of train

We assume that couplers can be modelled by a linear spring using Hooke's law:

$$f(x) = kx.$$

The train's total running resistance R consists of aerodynamic drag R^a and rolling mechanical resistance R^r and can be expressed using Davis' formula as follows [11].

$$R = \underbrace{c_0 + c_v v}_{R^r} + \underbrace{c_a v^2}_{R^a}, \quad (2-2)$$

where v is the wagon velocity, the coefficients c_0 , c_v and c_a can be obtained from wind tunnel testing, $c_a v^2$ represents the aerodynamic resistance and $c_0 + c_v v$ represents the mechanical resistance. At low velocities, the mechanical resistance dominates; at high velocities, the aerodynamic resistance dominates. Figure 2-7 shows the force diagram of the train in longitudinal motion. It is assumed that the aerodynamic resistance only acts on the first wagon and that the other wagons experience mechanical rolling resistance. The equations of motion

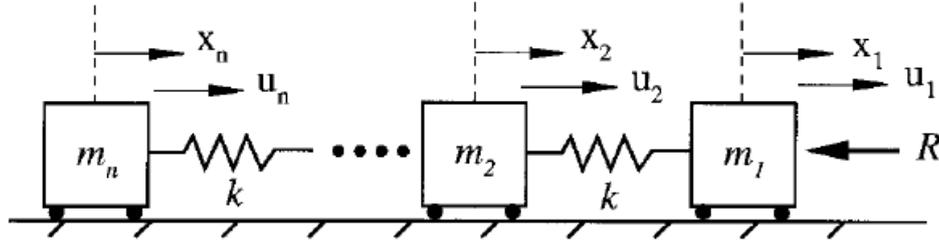


Figure 2-7: Force diagram of high speed train in longitudinal motion [38].

for a train with n wagons can now be derived by applying Newton's second law of motion on each wagon, noting which forces act on the wagon. Equation 2-3 describes the dynamics of the train.

$$\left. \begin{aligned} m_1 \ddot{x}_1 &= u_1 - k(x_1 - x_2) - \underbrace{(c_0 + c_v \dot{x}_1)}_{R_1^r} m_1 - \underbrace{c_a \dot{x}_1^2}_{R^a} \left(\sum_{i=1}^n m_i \right) \\ m_i \ddot{x}_i &= u_i - k(x_i - x_{i-1}) - k(x_i - x_{i+1}) - \underbrace{(c_0 + c_v \dot{x}_i)}_{R_i^r} m_i, \quad i = 2, \dots, n-1 \\ m_n \ddot{x}_n &= u_n - k(x_n - x_{n-1}) - \underbrace{(c_0 + c_v \dot{x}_n)}_{R_n^r} m_n \end{aligned} \right\} \quad (2-3)$$

As we wish to regulate the wagon speeds to a certain reference and bring the inter-wagon distances to zero, we rewrite Equation 2-3 using $v_i = \dot{x}_i$, $\dot{v}_i = \ddot{x}_i$, and $\Delta x_i = x_i - x_{i+1}$, which gives Equation 2-4.

$$\left. \begin{aligned} m_1 \dot{v}_1 &= u_1 - k\Delta x_1 - \underbrace{(c_0 + c_v v_1)}_{R_1^r} m_1 - \underbrace{c_a v_1^2}_{R^a} \left(\sum_{i=1}^n m_i \right) \\ m_i \dot{v}_i &= u_i + k\Delta x_{i-1} - k\Delta x_i - \underbrace{(c_0 + c_v v_i)}_{R_i^r} m_i, \quad i = 2, \dots, n-1 \\ m_n \dot{v}_n &= u_n + k\Delta x_{n-1} - \underbrace{(c_0 + c_v v_n)}_{R_n^r} m_n \end{aligned} \right\} \quad (2-4)$$

To linearise this model, a fixed point must be found to linearise around. For the DD train model, a fixed point is when all wagons have the desired speed v_r and all inter-wagon distances are zero. Mathematically, this is expressed as

$$\begin{aligned} v_1^e, v_2^e, \dots, v_n^e &= v_r \\ \Delta x_1^e, \Delta x_2^e, \dots, \Delta x_n^e &= 0 \end{aligned}$$

It follows that $\forall i = 1, \dots, n : \dot{v}_i^e = 0$. Plugging these values into Equation 2-4 yields the following equilibrium forces:

$$\begin{aligned} u_1^e &= (c_0 + c_v v_r) m_1 + c_a v_r^2 \left(\sum_{i=1}^n m_i \right) \\ u_i^e &= (c_0 + c_v v_r) m_i \end{aligned} \quad (2-5)$$

Let δv_i , δx , and δu represent the deviations from their equilibrium; then we have that $v_i = v_r + \delta v_i$, $\Delta x = \delta x$, and $u_i = u_i^e + \delta u_i$. Filling this into Equation 2-4 and ignoring the quadratic terms yields

$$\left. \begin{aligned} m_1 \delta \dot{v}_1 &= \delta u_1 - k \delta x_1 - c_v m_1 \delta v_1 - 2c_a v_r \left(\sum_{i=1}^n m_i \right) \delta v_1 \\ m_i \delta \dot{v}_i &= \delta u_i + k \delta x_{i-1} - k \delta x_i - c_v m_i \delta v_i, \quad i = 2, \dots, n-1 \\ m_n \delta \dot{v}_n &= \delta u_n + k \delta x_{n-1} - c_v m_n \delta v_n \end{aligned} \right\} \quad (2-6)$$

Equation 2-6 can be rewritten in state-space format with states $x = [\delta x_1 \quad \delta x_2 \quad \dots \quad \delta x_n]^T$ and inputs $u = [\delta u_1 \quad \delta u_2 \quad \dots \quad \delta u_n]^T$.

$$\dot{x} = Fx + Gu = \begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix} x + \begin{bmatrix} G_{11} \\ 0_{(n-1) \times l} \end{bmatrix} u, \quad (2-7)$$

where

$$\begin{aligned} F_{11} &= \begin{bmatrix} -c_v - 2c_a v_r \frac{\sum_{i=1}^n m_i}{m_1} & 0 & \dots & 0 \\ 0 & -c_v & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & -c_v \end{bmatrix}_{n \times n} \\ F_{12} &= \begin{bmatrix} \frac{-k^-}{m_1} & 0 & 0 & \dots & 0 \\ \frac{k^-}{m_2} & \frac{-k^-}{m_2} & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & \frac{k^-}{m_{n-1}} & \frac{-k^-}{m_{n-1}} \\ 0 & \dots & \dots & 0 & \frac{k^-}{m_n} \end{bmatrix}_{n \times (n-1)} \\ F_{21} &= \begin{bmatrix} 1 & -1 & 0 & \dots & 0 \\ 0 & 1 & -1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & 1 & -1 \end{bmatrix}_{(n-1) \times n} \\ F_{22} &= 0_{(n-1) \times (n-1)} \end{aligned}$$

and G_{11} depends on the type of train:

$$G_{11} = \begin{cases} \begin{bmatrix} 1 & \dots & 0 \\ 0 & \dots & \vdots \\ \vdots & \dots & 0 \\ 0 & \dots & 1 \end{bmatrix}_{n \times l}, & \text{for the PPD type} \\ I_{n \times n} & \text{for the DD type} \end{cases}$$

A more realistic model can be found by including the effect of damping in the couplers, the resistance due to the gravitational forces on slopes, and the resistance the train experiences

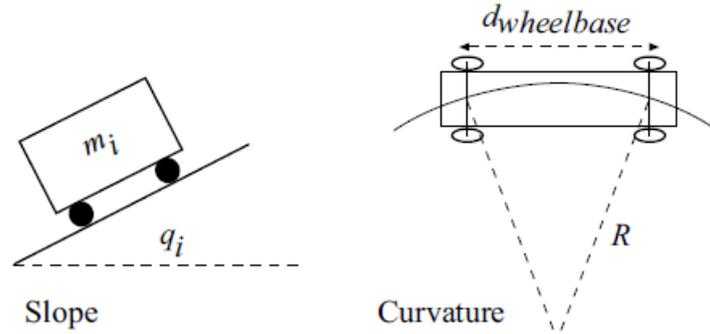


Figure 2-8: Resistances due to slope and curvature angles [16].

in curvatures. Figure 2-8 portrays these slope and curvature resistances. The coupler force modelled as a linear spring-damper system is described by the following equation [16]:

$$f(x) = k(x_1 - x_2) + d(\dot{x}_1 - \dot{x}_2),$$

where k is the spring constant and d is the damping constant. The improved non-linear model can then be described as in Equation 2-4 by the following equations:

$$\left. \begin{aligned} m_1 \dot{v}_1 &= u_1 - k\Delta x_1 - d(v_1 - v_2) - \underbrace{(c_0 + c_v v_1)}_{R_1^r} m_1 - \underbrace{c_a v_1^2}_{R^a} \left(\sum_{i=1}^n m_i \right) \\ &\quad - 9.98 \sin \theta_1 m_1 - 0.004 D_1 m_1 \\ m_i \dot{v}_i &= u_i + k\Delta x_{i-1} - k\Delta x_i + d v_{i-1} - 2d v_i + d v_{i+1} - \underbrace{(c_0 + c_v v_i)}_{R_i^r} m_i \\ &\quad - 9.98 \sin \theta_i m_i - 0.004 D_i m_i, \quad i = 2, \dots, n-1 \\ m_n \dot{v}_n &= u_n + k\Delta x_{n-1} + d v_{n-1} - d v_n - \underbrace{(c_0 + c_v v_n)}_{R_n^r} m_n - 9.98 \sin \theta_n m_n - 0.004 D_n m_n, \end{aligned} \right\} \quad (2-8)$$

where v_i and Δx_i follow the same definition as in Equation 2-4, θ_i is the slope angle of the i -th wagon, and the degree of curvature $D_i = 0.5 d_{wheelbase_i} / R$ where $d_{wheelbase_i}$ is the horizontal distance between the front and rear wheels of the i -th wagon and R is the curve radius. After linearisation, these models lend themselves well to the design of state-feedback controllers. For example, [39] compares open-loop heuristic scheduling and LQR closed-loop control in terms of reference tracking performance, in-train force minimisation, and energy consumption. [38] synthesises a mixed H_2/H_∞ cruise controller for a DD and PPD train by solving linear matrix inequalities (LMIs) to satisfy speed reference tracking performance (captured by H_2 norm) and gust attenuation performance (captured by H_∞ norm). To achieve reference tracking with no steady-state error when the train is subject to step disturbances, integrators need to be added to the controller architecture, as shown in Figure 2-9. In general, integral action to eliminate steady-state error can be added to a linear system as follows [21]. We start with a typical state-space realisation.

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned}$$

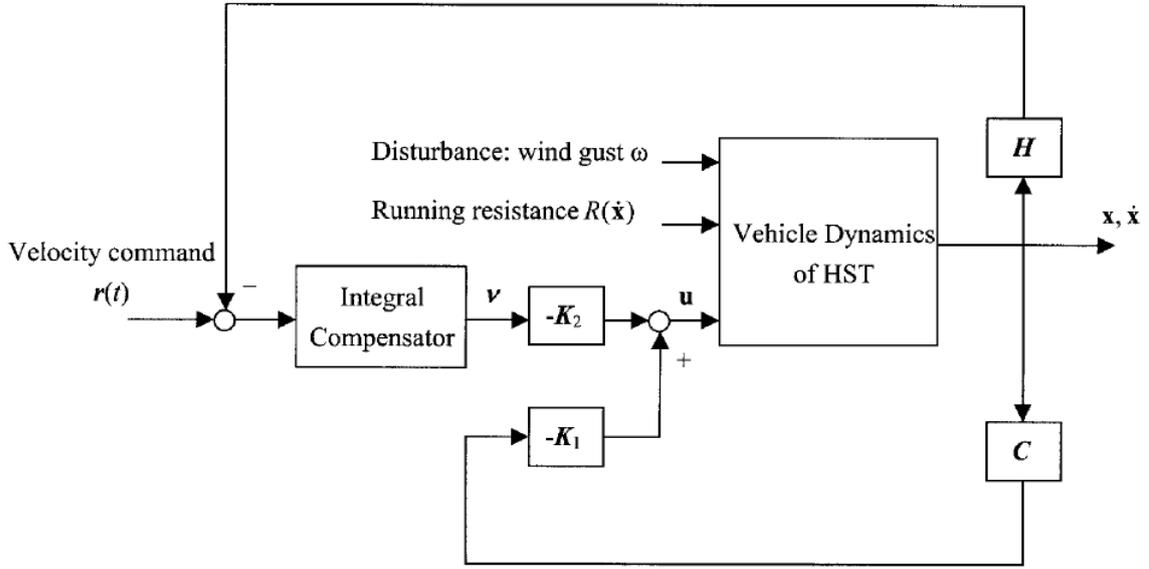


Figure 2-9: Controller architecture with integral compensator [38].

The variable x_i integrates the error between the step reference and the states that should track it, selected using the selection matrix H .

$$x_i = \int (r - Hx) dt$$

It can subsequently be added to the system's dynamical equations as follows.

$$\begin{aligned} \dot{x} &= Ax + Bu \\ \dot{x}_i &= r - Hx \\ y &= Cx \end{aligned}$$

Rewriting gives:

$$\begin{aligned} \begin{bmatrix} \dot{x} \\ \dot{x}_i \end{bmatrix} &= \begin{bmatrix} A & 0 \\ -H & 0 \end{bmatrix} \begin{bmatrix} x \\ x_i \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ 1 \end{bmatrix} r \\ y &= \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} x \\ x_i \end{bmatrix} \end{aligned} \quad (2-9)$$

When a system description is rewritten so that the dynamical equations describe a system we wish to stabilise around the origin, r is zero. A state-feedback controller for the extended system consists of a feedback term on the states and integrated error:

$$u = - \begin{bmatrix} K_x & K_i \end{bmatrix} \begin{bmatrix} x \\ x_i \end{bmatrix} = -K_x x - K_i x_i. \quad (2-10)$$

Figure 2-10 and Figure 2-11 show the results of [38] of reference tracking and disturbance rejection, respectively.

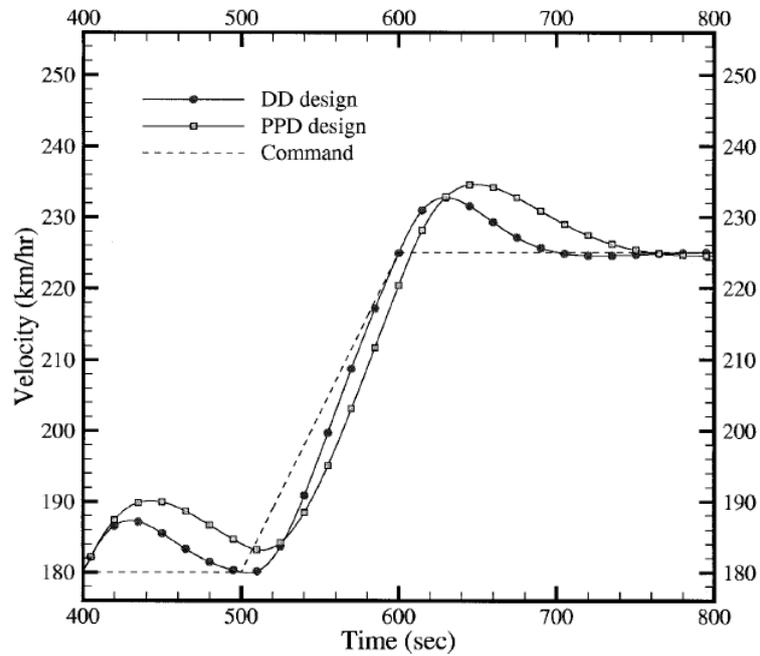


Figure 2-10: Reference tracking results of the DD and PPD train [38].

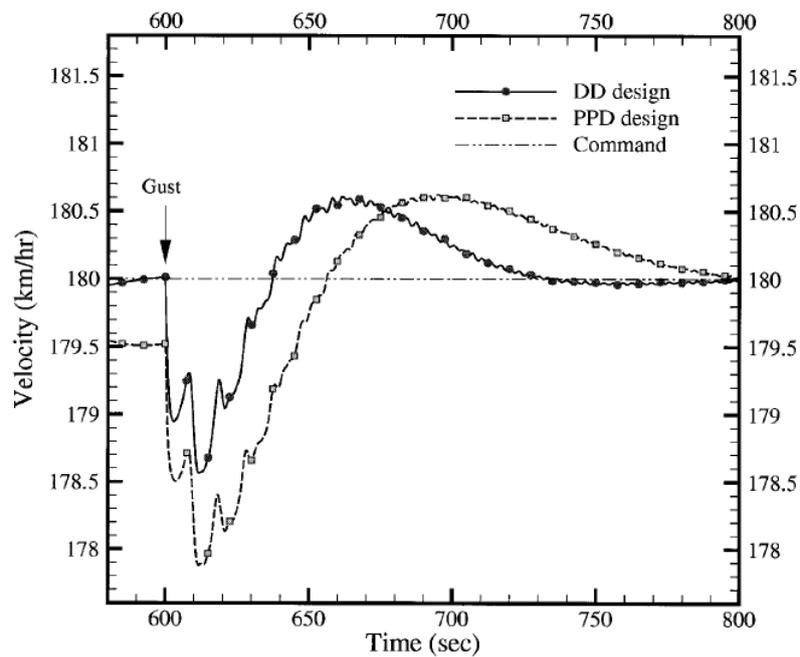


Figure 2-11: Disturbance rejection results of the DD and PPD train [38].

Discussion

By not modelling the train as a rigid body as was done by Howlett, a more realistic model of the train dynamics was attained. Model realism can be improved by adding the effect of

damping to the couplers, and modelling the resistances the train experiences. The impact of disturbances on reference tracking can be handled by using integrators. However, these additions come at the cost of extra complexity. Unlike Howlett's model, the linearised models do not include notches. Moreover, there are no enforced limits to actuation and no signal delays.

2-1-4 Consensus control

EMU model

[34] investigates the potential of event-triggered control for consensus control of a high-speed train. The novel contributions of the paper are threefold:

1. The elastic coupling coefficient is taken as a non-linear function instead of a constant.
2. An event-triggered controller is used to achieve consensus control of the electric multiple unit (EMU).
3. A second event-triggered mechanism is proposed that can achieve tracking in finite time.

This part will go over these novel contributions and present the stability theorems. Since a complete derivation of the theorems will not be given, some variables are left undefined. Deriving the meaning of all variables would take up significant space and distract from the demonstrative purpose of these theorems for this thesis. Of course, the original paper can be consulted for these details [34]. Event-triggered control is explained in Section 2-2. Figure 2-12 shows the EMU consisting of n units connected through elastic couplers. Figure 2-13

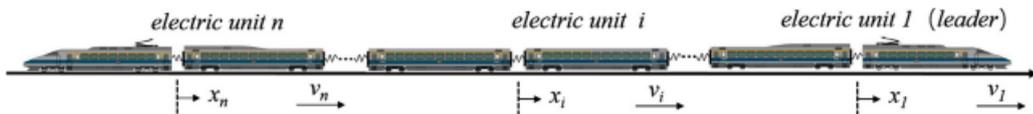


Figure 2-12: Diagram of EMU [34].

shows the control architecture for this system. The control centre can send a command v_r to the electric units, which are tasked to follow this reference command. The electric units can only exchange information with their neighbours. As the consensus controller of each wagon works by sampling data of the neighbouring wagons, it is convenient to write variables in terms of how they relate between two electric units i and j . Only the neighbouring electric units impact the coupler force of electric unit i . If the neighbouring set of electric unit i is defined as \mathcal{N}_i , we have that $\mathcal{N}_i = \{i + 1\}$ for $i = 1$, $\mathcal{N}_i = \{i - 1\}$, and $\mathcal{N}_i = \{i - 1, i + 1\}$ otherwise. Following this notation, the coupler force between electric unit i and j is given as

$$f_{ij} = k_{ij}x_{ij}, \quad i = 1, 2, \dots, n, j \in \mathcal{N}_i, \quad (2-11)$$

where the relative displacement x_{ij} is equal to

$$x_{ij} = x_j(t) - x_i(t) + l_{ij}, \quad i = 1, 2, \dots, n, j \in \mathcal{N}_i, \quad (2-12)$$

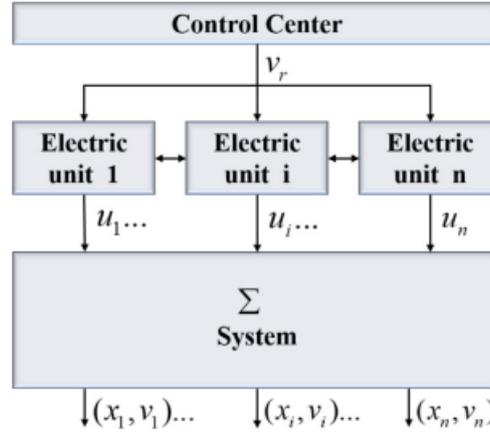


Figure 2-13: Control framework of the finite-time consensus train controller [34].

where the static centroid distance l between two adjacent electric units is equal to

$$l_{ij} = \begin{cases} l, & \text{if } i < j \\ -l, & \text{if } i > j. \end{cases} \quad (2-13)$$

The first novel contribution is the use of a non-linear elastic coefficient. The elastic coefficient is given by

$$k_{ij} = z(x_{ij}), \quad i = 1, 2, \dots, n, \quad j \in \mathcal{N}_i,$$

where function z satisfies the following criteria:

$$\left. \begin{aligned} z(x_{ij}) > 0, \quad \dot{z}(x_{ij}) > 0, \quad \ddot{z}(x_{ij}) > 0, & \text{ if } x_{ij} > 0, \\ z(x_{ij}) = 0, \quad \dot{z}(x_{ij}) = 0, & \text{ if } x_{ij} = 0, \\ z(x_{ij}) > 0, \quad \dot{z}(x_{ij}) < 0, \quad \ddot{z}(x_{ij}) > 0, & \text{ if } x_{ij} < 0, \end{aligned} \right\}$$

Figure 2-14 shows a plot of the elastic coefficient function k_{ij} as given by Equation 2-14, for $l = 0.5$ m, $h_1 = 2$, $h_2 = 2$, $h_3 = 10^5$, and $h_4 = 1$.

$$k_{ij} = h_1 \left(\frac{1}{\|x_{ij}^2 - (0.6)^2\|} - 4 \right) + h_2 \|x_{ij}\| + h_3 \left(e^{h_4 x_{ij}} + e^{-(h_4 x_{ij})} - 2 \right) \quad (2-14)$$

Filling in Equation 2-12 into Equation 2-11 yields

$$f_{ij} = k_{ij}(x_j(t) - x_i(t) + l_{ij}), \quad i = 1, 2, \dots, n, \quad j \in \mathcal{N}_i. \quad (2-15)$$

The modelling of the train follows the same steps as the coupler train model: a non-linear model is used to derive a state-space realisation and the equilibrium forces pertaining to the reference speed. The control force in equilibrium is given by

$$\bar{u}_i(t) = m_i \left(c_{i0} + c_{i1} v_r + c_{i2} v_r^2 + f_r(x_i(t)) + f_t(x_i(t)) \right), \quad i = 1, 2, \dots, n. \quad (2-16)$$

An event-triggered controller with the following form was proposed:

$$\hat{u}_i(t) = m_i \alpha_i \sum_{j=1}^n a_{ij} (v_j(t_k^i) - v_i(t_k^i)) + m_i \gamma_i \sum_{j=1}^n a_{ij} (\hat{x}_j(t_k^i) - \hat{x}_i(t_k^i)), \quad t \in [t_k^i, t_{k+1}^i). \quad (2-17)$$

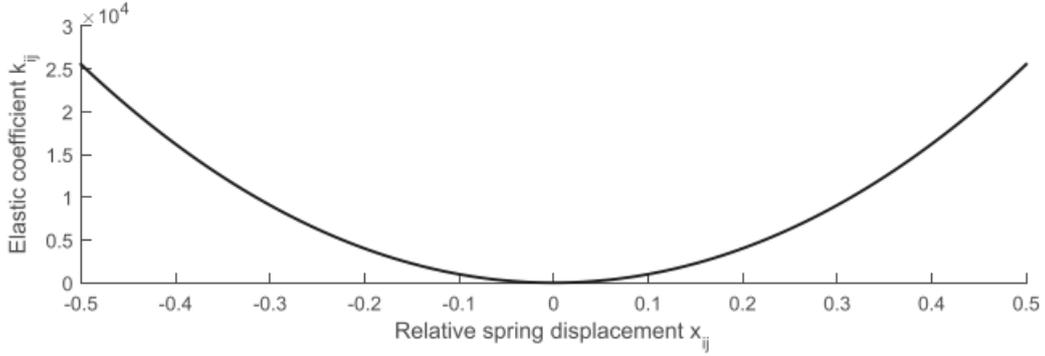


Figure 2-14: Plot of elastic coefficient k_{ij} for $l = 0.5$ m [34].

The triggering function is given by

$$f_i(\epsilon_{ix}(t), \epsilon_{iv}(t), \hat{x}_i(t), \hat{v}_i(t)) = b_i \left(\|\epsilon_{ix}(t)\|^2 + \|\epsilon_{iv}(t)\|^2 \right) - w_i \left(\|\hat{x}_i(t)\|^2 + \|\hat{v}_i(t)\|^2 \right).$$

Under this scheme, the triggering times are given by:

$$t_{k+1}^i = \inf \left\{ t > t_k^i : f_i(\epsilon_{ix}(t), \epsilon_{iv}(t), \hat{x}_i(t), \hat{v}_i(t)) > 0 \right\}, \quad k = 1, 2, \dots \quad (2-18)$$

Asymptotic event-triggered consensus control

Theorem 3.1 of the paper states that, if variables that are part of the controller and triggering condition follow certain conditions, the following results can be attained when the train is controlled by the aforementioned ETC controller:

1. The controller of each electric unit is only updated at its event triggering time instants determined by Equation 2-18, without continuous communication between adjacent electric units.
2. During the operation of the EMU, the speed of each electric unit is asymptotically approaching the desired speed v_r .
3. During the operation, the centroid distance between adjacent electric units is asymptotically stabilised in a safe range near l , and the internal force of the train is bounded and close to 0.

This ETC controller has a positive IST and thus avoids Zeno behaviour. The details and proof can be found in the paper.

Finite-time event-triggered consensus control

A second controller was developed that can achieve tracking in finite time. The controller takes the following form:

$$\begin{aligned} \hat{u}_i(t) = & m_i \alpha_i \sum_{j=1}^n a_{ij} (\hat{v}_j(t_k^i) - \hat{v}_i(t_k^i)) + m_i \gamma_i \sum_{j=1}^n a_{ij} (\hat{x}_j(t_k^i) - \hat{x}_i(t_k^i)) \\ & - m_i \text{sig} \left(\alpha_i \sum_{j=1}^n (\hat{v}_i(t_k^i) - \hat{v}_j(t_k^i)) + \gamma_i \sum_{j=1}^n a_{ij} (\hat{x}_i(t_k^i) - \hat{x}_j(t_k^i)) \right)^\psi, \quad t \in [t_k^i, t_{k+1}^i]. \end{aligned}$$

The triggering function is given by

$$f_i(\epsilon_{ix}(t), \epsilon_{iv}(t), H_i(t_k^i), Y_i(t_k^i)) = \|\mathcal{L}\| \|\alpha_i \epsilon_{iv}(t) + \gamma_i \epsilon_{ix}(t)\| - \eta \|\alpha_i Y_i(t_k^i) + \gamma_i H_i(t_k^i)\|.$$

The triggering times are equal to

$$t_{k+1}^i = \inf \left\{ t > t_k^i : f_i(\epsilon_{ix}(t), \epsilon_{iv}(t), H_i(t_k^i), Y_i(t_k^i)) > 0 \right\}, \quad k = 1, 2, \dots$$

Then, if the controller and triggering condition parameters follow certain conditions, Theorem 3.3 of the paper states that the EMU can achieve speed consensus in finite time. The details and proof can again be found in the paper.

Discussion

The model discussed here is special because of its architecture: each unit of the EMU can only communicate with its neighbour. This has the advantage that an easy communication topology can be used, which does not need signal repeaters. However, each unit in the EMU is a wagon with traction capacity, which is not true for freight trains. Moreover, each wagon must be able to compute actuator signals, instead of having one central controller. This requires that each wagon have computing resources, which might not be available in the low-energy scenario with energy harvesting sensors. A positive aspect of the model is the use of a realistic non-linear coupler. Finally, the use of ETC enables more optimal use of communication resources. Event-triggered control, its advantages and problems, will be explored further in Section 2-2. The paper exemplifies how a continuous event-triggered controller can have a non-zero minimum event-time, and can even be proven to reach consensus in finite time.

2-1-5 Final choice of model

The previous models all have their relative strengths and weaknesses. For each model, a suitable controller type was also presented. Choosing an adequate model is an exercise in balancing realism and practicality. Several considerations that justify the final model choice will now be given. Finally, this model is mathematically formalised.

Model choice considerations

This first model presented, the point mass model, has the limitation that it does not include the effect of the couplers. As reducing the in-train forces reduces wear and tear and therefore reduces maintenance costs, a model that includes these variables is preferred. The model demonstrated in Equation 2-4 considers the train as an interconnection of wagons connected by couplers. The variables in this model are the speeds and inter-wagon distances. An even more realistic model, including the effect of damping in the couplers and the resistances caused by slopes and the curvature angle, was presented in Equation 2-8. As the slope and curvature are generally unknown and varying, they can be considered disturbances to the train modelled as moving along a straight, flat path. Two types of trains were considered: PPD, in which only the first and last wagons are powered, and DD, where all wagons are motorised. A freight train can be composed of several locomotives with multiple unmotorised wagons in between. Therefore, PPD is the more appropriate model. In a heavy-haul train, the wagons can be equipped with brakes. However, this increases the complexity of the design and was thus not modelled, at the cost of increased inter-wagon forces. Integrators will not be added, to simplify the design and to give an indication of the base performance during disturbances. The EMU model described a DD train where communication is limited to the neighbouring wagons. In addition to the fact that DD is not realistic for a freight train, the problem with this model is that the communication topology requires that each wagon have computing resources. This is not realistic when energy resources are scarce. Event-triggered control was used to make more effective use of the communication resources. This idea will be further explored later. Finally, only the model presented by Howlett considers limitations on actuation freedom. Usually, a non-linear model is linearised for the control design. The controller is then designed so that these details, e.g., limits on actuation effort, have minimal effect. Therefore, a model that does not include these details will be used. This model will now be presented.

Model formalisation

The model that includes the damping effect of the couplers and ignores the slope and curvature disturbances and uses a linear spring-damper for the couplers is given by

$$\left. \begin{aligned} m_1 \dot{v}_1 &= u_1 - k\Delta x_1 - d(v_1 - v_2) - \underbrace{(c_0 + c_v v_1)}_{R_1^r} m_1 - \underbrace{c_a v_1^2}_{R^a} \left(\sum_{i=1}^n m_i \right) \\ m_i \dot{v}_i &= u_i + k\Delta x_{i-1} - k\Delta x_i + dv_{i-1} - 2dv_i + dv_{i+1} - \underbrace{(c_0 + c_v v_i)}_{R_i^r} m_i, \quad i = 2, \dots, n-1 \\ m_n \dot{v}_n &= u_n + k\Delta x_{n-1} + dv_{n-1} - dv_n - \underbrace{(c_0 + c_v v_n)}_{R_n^r} m_n, \end{aligned} \right\}$$

Following the same logic to derive Equations 2-6 and 2-5, the linearisation around the reference velocity can be attained as shown in Equation 2-19. The equilibrium forces are the same as

in Equation 2-5.

$$\left. \begin{aligned} m_1 \delta \dot{v}_1 &= \delta u_1 - k \delta x_1 - d \delta v_1 + \delta d v_2 - c_v m_1 v_1 - 2c_a v_r v_1 \sum_{i=1}^n m_i \\ m_i \delta \dot{v}_i &= \delta u_i - k \delta x_i + k \Delta x_{i-1} + d \delta v_{i-1} - 2d \delta v_i + d \delta v_{i+1} - c_v m_i v_i, \quad i = 2, \dots, n-1 \\ m_n \delta \dot{v}_n &= \delta u_n + k \delta x_{n-1} - d \delta v_n + d \delta v_{n-1} + c_v m_n \delta v_n \end{aligned} \right\} \quad (2-19)$$

Rewriting Equation 2-19 as a state-space as in Equation 2-7 leads to the following matrices:

$$\begin{aligned} F_{11} &= \begin{bmatrix} -d - c_v - 2c_a v_r \frac{\sum_{i=1}^n m_i}{m_1} & d & 0 & \dots & 0 \\ d & -2d - c_v & d & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & d & -d - c_v \end{bmatrix}_{n \times n} \\ F_{12} &= \begin{bmatrix} \frac{-k^-}{m_1} & 0 & 0 & \dots & 0 \\ \frac{k^-}{m_2} & \frac{-k^-}{m_2} & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & \frac{k^-}{m_{n-1}} & \frac{-k^-}{m_{n-1}} \\ 0 & \dots & \dots & 0 & \frac{k^-}{m_n} \end{bmatrix}_{n \times (n-1)} \\ F_{21} &= \begin{bmatrix} 1 & -1 & 0 & \dots & 0 \\ 0 & 1 & -1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & 1 & -1 \end{bmatrix}_{(n-1) \times n} \\ F_{22} &= 0_{(n-1) \times (n-1)}, \end{aligned}$$

and

$$G_{11} = \begin{bmatrix} 1 & \dots & 0 \\ 0 & \dots & \vdots \\ \vdots & \dots & 0 \\ 0 & \dots & 1 \end{bmatrix}_{n \times l}.$$

As mentioned before, PPD is the most realistic option for a freight train. As noted by [38], the assumption that $\Delta x_1 = \Delta x_2 = \dots = \Delta x_n = 0$ does not hold for a PPD train in equilibrium, as there are not enough control inputs to cancel the rolling resistance in each wagon. Nevertheless, the assumption is still a good approximation, i.e., $\Delta x_1 \approx \Delta x_2 \approx \dots \approx \Delta x_n \approx 0$. The equilibrium values are required to accurately approximate the linearised system around the equilibrium. If we assume the train has reached an equilibrium velocity, the equations describing the uncontrollable wagons (i.e., wagons $i = 2, \dots, n-1$) are given by

$$0 = k \Delta x_{i-1} - k \Delta x_i - \underbrace{(c_0 + c_v v_i)}_{R^r} m_i, \quad i = 2, \dots, n-1 \quad (2-20)$$

The previous equation can be transformed to the following matrix equation:

$$Yx - Z = 0, \quad (2-21)$$

where

$$Y = \begin{bmatrix} -k & k & 0 & \dots & 0 \\ 0 & -k & k & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & -k & k \end{bmatrix}, \quad Z = \begin{bmatrix} (c_0 + c_v v_2)m_2 \\ \vdots \\ (c_0 + c_v v_{n-1})m_{n-1} \end{bmatrix}, \quad x = \begin{bmatrix} \Delta x_1 \\ \vdots \\ \Delta x_{n-1} \end{bmatrix}.$$

Since Equation 2-21 has no unique solution, a constraint on x must be imposed to find a solution. To minimise the maximum force in the train, the following optimisation problem can be solved:

$$\begin{aligned} \min \quad & \|x\|_\infty \\ \text{s.t.} \quad & Yx - Z = 0 \end{aligned}$$

Table 2-1 shows the parameter values of the train that will be used in this thesis. These values were taken from [38].

Table 2-1: Train parameter values.

Symbol	Value	Unit
m	40×10^3	kg
v_r	User	m/s
c_0	0.01176	N/kg
c_v	0.00077616	N s/m kg
c_a	1.6×10^{-5}	N s ² /m ² kg
k	80×10^3	N/m
d	800	N s/m

2-2 Improving sampling efficiency

This section goes over strategies to improve sampling efficiency, taken to be the number of samples used to obtain a specific control performance.

2-2-1 Motivation

Modern digital controllers are typically implemented in a periodic fashion, i.e., they sample a plant at some predefined fixed rate and update the control inputs according to a feedback law. These controllers' ease of design and implementation still makes them the preferred approach for many control problems. However, there are disadvantages to periodic control that have led to a search for alternative, aperiodic control methods. In periodic control, sampling must be fast enough to guarantee stability. This comes at the cost of wasted resources when the system does not require sampling, e.g., when the system is in equilibrium or the sensor reading has not changed. Aperiodic forms of control aim to address this problem by not requiring that sampling take place at a fixed time interval; instead, the next sampling instant is determined dynamically. Figure 2-15 exemplifies the difference between periodic and aperiodic sampling. If sampling is cheap and periodicity can easily be accommodated, periodic controllers have the advantage of being simple to design and tune. Furthermore, they come with an existing body



Figure 2-15: Periodic (left) and aperiodic (right) sampling [22].

of mathematical theory on digital control, which is taught in undergraduate and graduate classes on systems and control theory and is thus familiar to many engineers. However, there are scenarios where these assumptions do not hold, and samples may be difficult or costly to obtain, or a limited bandwidth necessitates frugal sampling. A realistic and common scenario is that of a Networked Control System (NCS). In a NCS, sensors, controllers, and actuators typically share a communication channel. A depiction of a NCS with N control loops sharing the same communication channel is shown in Figure 2-16. An advantage of using NCSs is

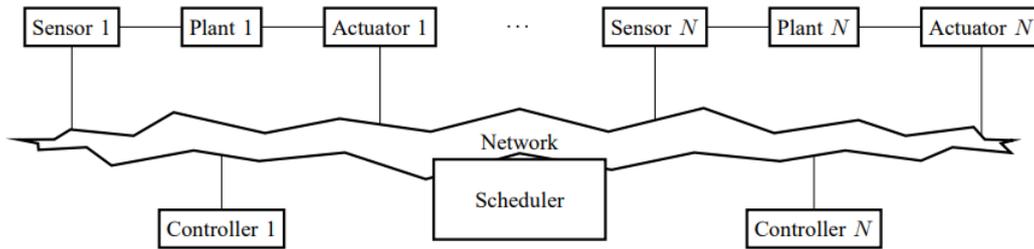


Figure 2-16: Topology of a set of N networked control systems [1].

that they reduce the amount of physical wiring between sensors, plant, and actuators. Thus, they have a more flexible architecture, which can reduce maintenance costs and make it easier to alter the control architecture topology. A disadvantage is that a wireless communication channel comes with the constraint of limited bandwidth. The limitation of communication resources puts a restraint on the number of control loops that can be interconnected and the sampling frequencies at which they can be run. The study of aperiodic control aspires to mitigate these problems. The mathematical framework for this is introduced next.

2-2-2 Mathematical framework

We start by considering a non-linear system

$$\dot{x}(t) = f(x(t), u(t), w(t)), \quad (2-22)$$

where the states $x \in \mathbb{R}^{n_x}$, control input $u \in \mathbb{R}^{n_u}$, and exogenous input $w \in \mathbb{R}^{n_w}$ are time-dependent vectors of the given dimensions. In a sampled data system with sampling times t_k , $k \in \mathbb{N}$, the following samples are obtained under a zero-order sample-and-hold sampling method:

$$\hat{x}(t) = x(t_k), \quad \forall t \in [t_k, t_{k+1}).$$

Zero-order sample-and-hold entails that the sample is held constant in between sampling instants. In a feedback control system with control function $g(x)$, the control input is a function of the sampled system state, i.e.,

$$u(t) = g(\hat{x}(t)).$$

It follows that the control signal is also held constant between samples. Often, a linearised system is used to design the controller. A general linearisation of Equation 2-22 is given by 2-23.

$$\dot{x} = A^p x + B^p \hat{u} + B^w w, \quad (2-23)$$

where the states $x \in \mathbb{R}^{n_x}$, control input $\hat{u} \in \mathbb{R}^{n_u}$, and exogenous input $w \in \mathbb{R}^{n_w}$ are, again, time-dependent vectors of the given dimensions. A state-feedback controller is given by

$$\hat{u}(t) = Kx(t_k), \quad t \in (t_k, t_{k+1}].$$

We define the inter-sample time (IST) as:

$$\tau(x(t_k)) = \tau_k = t_{k+1} - t_k. \quad (2-24)$$

The minimum inter-sample time (MIST) τ_{\min} is defined as the minimum inter-sample time that can show up in the system. Formally:

$$\tau_{\min} = \inf_{k \in \mathbb{N}} \tau_k.$$

Next, we consider different aperiodic control methods that address the issue of improving the inter-sample times.

2-2-3 Event-triggered control

One form of aperiodic control is event-triggered control (ETC). In ETC, sensing and actuation are done when the system needs attention. This is assessed by monitoring a triggering function \mathcal{C} ; if this triggering function is violated ("triggers"), a new measurement is sent to the controller. This is formalised in the following equation:

$$\hat{x}(t) = \begin{cases} x(t_k), & \text{when } \mathcal{C}(x(t_k), \hat{x}(t_k)) > 0 \\ \hat{x}(t_k), & \text{when } \mathcal{C}(x(t_k), \hat{x}(t_k)) \leq 0 \end{cases}$$

Figure 2-17 shows a schematic of an event-triggered controller. The control loop is closed when the event-triggering condition is met. Two forms of ETC can be distinguished: continuous event-triggered control (CETC) and periodic event-triggered control (PETC).

Continuous event-triggered control

In CETC, the triggering condition is monitored continuously. The triggering function is a function of the current system state and the last state sent to the controller. The triggering function triggers when $\mathcal{C}(x(t), x(t_k)) > 0$, resulting in the following triggering times:

$$t_0 = 0, \quad t_{k+1} = \inf \{t > t_k \mid \mathcal{C}(x(t), \hat{x}(t)) > 0\}. \quad (2-25)$$

Since the continuous-time behaviour of CETC is at odds with digital hardware, implementations require special hardware or are approximated by a very high sampling frequency. A significant disadvantage is that CETC can have a MIST of zero, leading to so-called Zeno behaviour, in which an infinite number of triggers occur in a finite amount of time. Digital implementations cannot fulfil this demand. However, there are CETC schemes where a positive MIST can be proven to exist.

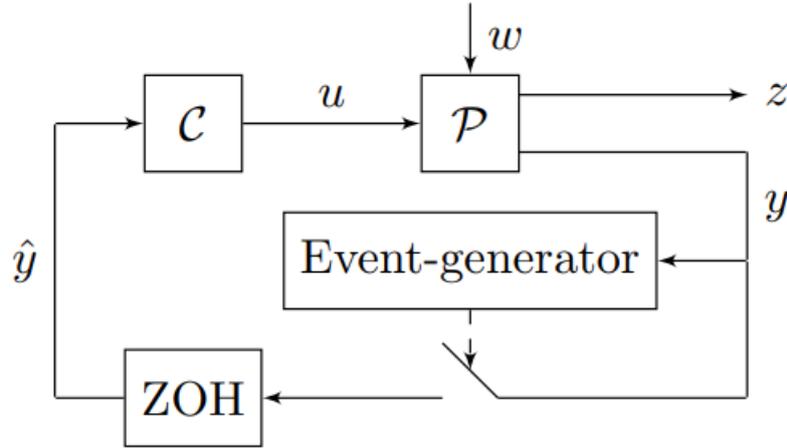


Figure 2-17: Schematic of an event-triggered controller [6].

Periodic event-triggered control

As discrete implementations lend themselves better to implementations in hardware, periodic event-triggered control (PETC) has been researched. In PETC, the triggering function is monitored periodically instead of continuously. Control updates will, therefore, only be possible at discrete time instances. The triggering times are then given by:

$$t_0 = 0, \quad t_{k+1} = \inf \{kh > t_k, k \in \mathbb{N} \mid \mathcal{C}(x(t), \hat{x}(t)) > 0\} \quad (2-26)$$

The fundamental checking period h ensures a positive MIST. Because of this, Zeno behaviour is averted.

Triggering conditions

Different triggering conditions can be designed to stabilise the ETC system. The next parts will explore a subset of possible options. Of particular interest are triggering conditions that are based on a quadratic triggering matrix

$$\mathcal{C}(\xi(t_k)) = \xi^T(t_k) Q \xi(t_k),$$

with $Q \in \mathbb{R}^{n_\xi \times n_\xi}$ and $\xi(t_k) = [x(t_k)^T \quad \hat{x}(t_k)^T]^T$.

State error

A commonly-used triggering condition is based on the state error. The triggering condition triggers when the magnitude of the difference between the measurement $\hat{x}(t_k)$ that was used for the last control update and the current state measurement $x(t_k)$ exceeds a value that depends on the magnitude of the current state and a design parameter σ .

$$\|\hat{x}(t_k) - x(t_k)\| > \sigma \|x(t_k)\|,$$

for $k \in \mathbb{N}$ and $\sigma > 0$. The triggering matrix for this triggering condition is given by

$$Q = \begin{bmatrix} (1 - \sigma^2)I & -I \\ -I & I \end{bmatrix}.$$

Input error

Another triggering condition is based on the input error:

$$\|K\hat{x}(t_k) - Kx(t_k)\| > \sigma\|Kx(t_k)\|,$$

for $k \in \mathbb{N}$ and $\sigma > 0$. As $u(t_k) = Kx(t_k)$, this can be rewritten as

$$\|\hat{u}(t_k) - u(t_k)\| > \sigma\|u(t_k)\|.$$

The triggering matrix for this condition is given by

$$Q = \begin{bmatrix} (1 - \sigma^2)K^T K & -K^T K \\ -K^T K & K^T K \end{bmatrix}$$

Lyapunov-based

The following part will introduce Lyapunov-based triggering conditions and lay the ground-work necessary to prove the stability of PETC schemes. The theorems and argumentative structure presented here are taken from [23]. A Lyapunov-based triggering condition can be derived using the discretisation of Equation 2-23 for $w = 0$, which gives

$$x(t_{k+1}) = A_d x(t_k) + B_d u(t_k), \quad (2-27)$$

where

$$A_d = e^{A^p h}, \quad B_d = \int_0^h e^{A^p s} B^p ds \quad (2-28)$$

and $u(t_k) = Kx(t_k)$ if feedback control is used. If a controller K is designed such that $A+BK$ has its roots in the open unit circle, there exists a quadratic Lyapunov function $V(x) = x^T P x$, $x \in \mathbb{R}^{n_x}$ such that

$$P \succ 0 \text{ and } (A+BK)^T P (A+BK) \preceq \lambda P \quad (2-29)$$

with $0 \leq \lambda < 1$. This implies that the Lyapunov function is non-increasing, i.e.,

$$V(x(t_{k+1})) \leq \lambda V(x(t_k)), \quad \forall k \in \mathbb{N}.$$

An event-triggering condition based on the violation of this decrease can be constructed. If the Lyapunov function no longer declines with a factor β while applying control input $u(t_k) = Kx(t_k)$, i.e., if

$$(A_d x(t_k) + B_d K \hat{x}(t_k))^T P (A_d x(t_k) + B_d K \hat{x}(t_k)) > \beta x^T(t_k) P x(t_k) \quad (2-30)$$

then \hat{x} and u are updated. The triggering matrix for Equation 2-30 is given by

$$Q = \begin{bmatrix} A_d^T P A_d - \beta P & A_d^T P B_d K \\ (B_d K)^T P A_d & (B_d K)^T P B_d K \end{bmatrix}$$

For any choice of β , Equation 2-30 is a Lyapunov function for the discrete-time system given by Equation 2-27. It holds that

$$V(x(t_{k+1})) \leq \max(\lambda, \beta) V(x(t_k)), \quad \forall k \in \mathbb{N}.$$

The PETC system can be rewritten as an impulsive system as follows:

$$\begin{aligned} \frac{d}{dt} \begin{bmatrix} \xi \\ \tau \end{bmatrix} &= \begin{bmatrix} \bar{A}\xi + \bar{B}w \\ 1 \end{bmatrix}, \text{ when } \tau \in [0, h] \\ \begin{bmatrix} \xi^+ \\ \tau^+ \end{bmatrix} &= \begin{cases} \begin{bmatrix} J_1\xi \\ 0 \end{bmatrix}, & \text{when } \xi^T Q \xi > 0, \tau = h \\ \begin{bmatrix} J_2\xi \\ 0 \end{bmatrix}, & \text{when } \xi^T Q \xi \leq 0, \tau = h \end{cases} \\ z &= \bar{C}\xi + \bar{D}w, \end{aligned} \quad (2-31)$$

where

$$\bar{A} = \begin{bmatrix} A^p & B^p K \\ 0 & 0 \end{bmatrix}, \bar{B} = \begin{bmatrix} B^w \\ 0 \end{bmatrix}, J_1 = \begin{bmatrix} I & 0 \\ I & 0 \end{bmatrix}, J_2 = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}$$

and with $\xi = [x^T \ \hat{x}^T]^T$ and where $z \in \mathbb{R}^{n_z}$ is the performance output and τ tracks the time since the last sampling instant. [23] introduces the Hamiltonian matrix H :

$$H = \begin{bmatrix} \bar{A} + \rho I + \gamma^{-2} \bar{B} M \bar{D}^T \bar{C} & \bar{B} M \bar{B}^T \\ -\bar{C}^T L \bar{C} & -(\bar{A} + \rho I + \gamma^{-2} \bar{B} M \bar{D}^T \bar{C})^T \end{bmatrix}$$

with $L = \gamma^2 I - \bar{D} \bar{D}^T$. It also introduces the following matrix exponential:

$$F(\tau) = e^{-H\tau} = \begin{bmatrix} F_{11}(\tau) & F_{12}(\tau) \\ F_{21}(\tau) & F_{22}(\tau) \end{bmatrix}.$$

Let $\bar{F}_{11} = F_{11}(h)$, $\bar{F}_{12} = F_{12}(h)$, $\bar{F}_{21} = F_{21}(h)$, $\bar{F}_{22} = F_{22}(h)$, and \bar{S} such that $\bar{S} \bar{S}^T = -\hat{F}_{11}^{-1} \hat{F}_{12}$, then [23] presented the following theorem.

Theorem 1. *Consider the impulsive system 2-31 and let $\rho > 0$, $\gamma > \sqrt{\lambda_{\max}(\bar{D}^T \bar{D})}$, and Assumption III.1 hold. Suppose that there exist a matrix $P_h \succ 0$ and scalars $\mu_i \geq 0$, $i \in \{1, 2\}$, such that for $i \in \{1, 2\}$*

$$\begin{bmatrix} P_h + (-1)^i \mu_i Q & J_i^T \bar{F}_{11}^{-T} P_h \bar{S} & J_i^T (\bar{F}_{11}^{-T} P_h \bar{F}_{11}^{-1} + \bar{F}_{21} \bar{F}_{11}^{-1}) \\ \star & I - \bar{S}^T P_h \bar{S} & 0 \\ \star & \star & \bar{F}_{11}^{-T} P_h \bar{F}_{11}^{-1} + \bar{F}_{21} \bar{F}_{11}^{-1} \end{bmatrix} \succ 0. \quad (2-32)$$

Then, the PETC system 2-31 is GES with convergence rate ρ (when $w = 0$) and has an \mathcal{L}_2 -gain from w to z smaller than or equal to γ .

In the absence of disturbances, this theorem can be simplified, and GES can be proven with the following corollary [23].

Corollary 1.1. *Consider the impulsive system 2-31 and let ρ be given. Assume there exist a matrix $P_h \succ 0$ and scalars $\mu_i \geq 0$, $i \in \{1, 2\}$ such that*

$$\begin{bmatrix} e^{-2\rho h} P_h + (-1)^i \mu_i Q & J_i^T e^{\bar{A}^T h} P_h \\ \star & P_h \end{bmatrix} \succ 0, i \in \{1, 2\}. \quad (2-33)$$

Then, the PETC system 2-31 is GES (for $w = 0$) with decay rate ρ .

[23] formulated less conservative conditions based on a discrete-time piecewise linear (PWL) model, obtained by discretising Equation 2-31 at sampling times $t_k = kh$, $k \in \mathbb{N}$, $\tau(0) = h$, $w = 0$ and defining $\xi_k = \xi(t_k)$, the following bimodal PWL model is obtained:

$$\xi_{k+1} = \begin{cases} A_1 \xi_k, & \text{when } \xi_k^T Q \xi_k > 0, \\ A_2 \xi_k, & \text{when } \xi_k^T Q \xi_k \leq 0, \end{cases} \quad (2-34)$$

where

$$A_1 := e^{\bar{A}h} J_1 = \begin{bmatrix} A_d + B_d K & 0 \\ I & 0 \end{bmatrix}, A_2 := e^{\bar{A}h} J_2 = \begin{bmatrix} A_d & B_d K \\ 0 & I \end{bmatrix}, \quad (2-35)$$

and A_d and B_d as in Equation 2-28. Using the PWL model given by Equation 2-34 and a piecewise quadratic Lyapunov function

$$V(\xi) = \begin{cases} \xi^T P_1 \xi, & \text{when } \xi^T Q \xi > 0 \\ \xi^T P_2 \xi, & \text{when } \xi^T Q \xi \leq 0 \end{cases}, \quad (2-36)$$

the GES of the PETC system 2-31 is guaranteed under the conditions specified by Theorem 2.

Theorem 2. *The PETC system 2-31 is GES with decay rate ρ , if there exist matrices P_1, P_2 and scalars $\alpha_{ij} \geq 0, \beta_{ij} \geq 0$, and $\kappa_{ij} \geq 0, i, j \in \{1, 2\}$, satisfying*

$$e^{-2\rho h} P_i - A_i^T P_j A_i + (-1)^i \alpha_{ij} Q + (-1)^j \beta_{ij} A_i^T Q A_i \succeq 0, \quad \text{for all } i, j \in \{1, 2\}, \text{ and} \quad (2-37)$$

$$P_i + (-1)^i \kappa_i Q \succ 0, \quad \text{for all } i \in \{1, 2\}$$

[23] proves that Theorem 1 can never outperform Theorem 2 in terms of stability analysis. All other proofs and details of the argumentation can also be found in the aforementioned paper.

Dynamic triggering

While the literature usually considers static event-triggering mechanisms, it can be shown that dynamic triggering conditions can guarantee the same performance while requiring fewer events. [18] proposes an event-triggering mechanism with an internal variable η that satisfies the following differential equation:

$$\dot{\eta} = -\beta(\eta) + \sigma\alpha(\|x\|) - \gamma(\|e\|), \eta(0) = \eta_0,$$

where β is a locally Lipschitz continuous class \mathcal{K}_∞ function, and $\sigma \in (0, 1), \eta_0 \in \mathbb{R}_0^+$ are design parameters. The dynamic variable η can be considered a filtered version of the static triggering condition $\sigma\alpha(\|x\|) - \gamma(\|e\|) = 0$. The motivation behind the proposed triggering mechanism is that closed-loop stability does not require that $\sigma\alpha(\|x\|) - \gamma(\|e\|)$ is always non-negative; it suffices that it is negative on average. This fact can save on the number of control updates. The event-triggered rule is thus given by

$$t_0 = 0 \wedge t_{i+1} = \inf \{t \in \mathbb{R} \mid t > t_i \wedge \eta(t) + \theta (\sigma\alpha(\|x(t)\|) - \gamma(\|e(t^-)\|)) \leq 0\},$$

where $\theta \in \mathbb{R}_0^+$ is an additional design parameter. As $\theta \rightarrow \infty$, the dynamic triggering condition becomes equal to the static triggering condition. The paper proves that this CETC mechanism has a positive MIST.

Since CETC is not well-suited to implementation on digital hardware, [6] has investigated a periodic implementation of a dynamic triggering condition.

Relaxed triggering condition

Dynamic triggering reduces some of the conservatism of static event generators. Usually, the stability of a PETC scheme is proven through solving LMIs. LMI approaches are conservative and usually lead to the Lyapunov function decaying much quicker than prescribed. Therefore, designing a triggering condition with a prescribed decay rate that closely matches reality can further save on communication resources. This is the problem that [35] tackles by introducing a relaxed triggering condition, where the monitored Lyapunov function is bounded by an exponentially decaying function $x_{x_0}(t)^T P x_{x_0}(t) > x_0^T P x_0 e^{-\lambda t}$ with $0 < \lambda < \lambda_0$ being the desired decay rate. First, the paper introduces an auxiliary discrete-time variable $\eta : \mathbb{R} \rightarrow \mathbb{R}^{n_x}$ with the following evolution:

$$\begin{aligned}\eta(0) &= P^{\frac{1}{2}}x(0) \\ \eta(t_{k+1}) &= Ie^{-0.5\lambda\Delta}\eta(t_k^+) \\ \eta(t_i^+) &= P^{\frac{1}{2}}x(t_i)\end{aligned}$$

Introducing $\zeta = \begin{bmatrix} x^T & \hat{x}^T & \eta^T \end{bmatrix}^T$, the triggering times $\{t_i\}_{i \in \mathbb{N}}$ that guide the evolution of η are given by

$$\begin{aligned}t_0 &= 0, \\ t_{i+1} &= \inf \left\{ t > t_i \mid \zeta(t)^T Q_1 \zeta(t) > 0 \vee \zeta(t)^T Q_2 \zeta(t) > 0 \vee t = t_i + N_{\max}\Delta, t = k\Delta, k \in \mathbb{N} \right\},\end{aligned}\tag{2-38}$$

where

$$\begin{aligned}Q_1 &:= \begin{bmatrix} P & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -I \end{bmatrix}, \\ Q_2 &:= \begin{bmatrix} A^T P A & A^T P B K & 0 \\ (B K)^T P A & (B K)^T P B K & 0 \\ 0 & 0 & -I e^{-\lambda\Delta} \end{bmatrix}, \\ A &:= e^{A^p \Delta}, \quad B := \int_0^\Delta e^{A^p s} B^p ds,\end{aligned}$$

and three parameters are design choices: the triggering condition sample interval Δ , the desired decay rate λ , and the maximum inter-event number of steps N_{\max} . In Equation 2-38, $N_{\max}\Delta$ is the maximum inter-event time. These parameters can be chosen as follows. P is taken from the Lyapunov function of the closed-loop system, i.e., the ideal performance system. λ must be chosen smaller than the ideal decay λ_0 . If this is done, there exists a guaranteed τ_{\min} :

$$\tau_{\min} = \min \left\{ \tau \in \mathbb{R}^+ : \det M(\tau) = 0 \right\},$$

where

$$\begin{aligned}M(\tau) &:= C \left(e^{F^T \tau} C^T P C e^{F \tau} - C^T P C e^{-\lambda \tau} \right) C^T, \\ F &:= \begin{bmatrix} A^p + B^p K & B^p K \\ -A^p - B^p K & -B^p K \end{bmatrix}, \quad C := \begin{bmatrix} I & 0 \end{bmatrix}.\end{aligned}$$

Finally, Δ must be chosen such that $\Delta < \tau_{\min}$. [35] proves that these considerations render the PETC system stable.

2-2-4 Self-triggered control

While ETC implementations reduce the number of samples to obtain a given performance, they still require the continuous or periodic evaluation of the triggering function. This means that sensors would have to send state information to the controller repeatedly. Transmitting and listening for transmissions are significant sources of energy consumption. In self-triggered control (STC), the next triggering instant is computed along with the control signal. This allows the sensor radios in a wireless control network to temporarily shut down until the next triggering instant is about to occur, thereby saving energy. A further advantage is that knowing the triggering times beforehand is useful when scheduling multiple control loops. A disadvantage of STC is that it runs in open-loop during sampling and is therefore less robust to disturbances. In an STC implementation of PETC, the triggering times are given by

$$t_{i+1} = \inf \{t \in kh, k \in \mathbb{N} \mid t > t_i \text{ and } \mathcal{C}(t - t_i, x(t), \hat{x}(t))\},$$

$$\mathcal{C}(\tau, x, \hat{x}) = \begin{bmatrix} x \\ \hat{x} \end{bmatrix}^T Q \begin{bmatrix} x \\ \hat{x} \end{bmatrix} > 0 \text{ or } \tau > \bar{\tau},$$

where $\bar{\tau}$ is the maximum inter-event time, chosen by the user to give the system a "heartbeat". Without this, the system could run in open-loop for an undetermined amount of time and be subject to destabilising disturbances.

Abstractions

The notion of abstractions will be introduced to analyse a system's sampling behaviour. [36] defines a generalised transition system, which is a general model of how a system evolves. A transition system \mathcal{S} is a tuple $(\mathcal{X}, \mathcal{X}_0, \mathcal{U}, \mathcal{E}, \mathcal{Y}, \mathcal{H})$ where:

- \mathcal{X} is the set of states,
- $\mathcal{X}_0 \subseteq \mathcal{X}$ is the set of initial states,
- \mathcal{U} is the set of actions,
- $\mathcal{E} \subseteq \mathcal{X} \times \mathcal{U} \times \mathcal{X}$ is the set of edges (or transitions),
- \mathcal{Y} is the set of outputs, and
- $\mathcal{H} : \mathcal{X} \rightarrow \mathcal{Y}$ is the output map.

Let $\tau(x)$ be the time until the next sample and let $\xi_x(\tau(x))$ be the state at the next sample, then an ETC timing system can be formalised as follows:

- $\mathcal{X} = \mathbb{R}^n$,
- $\mathcal{X} = \mathbb{R}^n$,
- $\mathcal{U} = \emptyset$
- $(x, x') \in \mathcal{E}$ iff $\xi_x(\tau(x)) = x'$

- $\mathcal{Y} \subset \mathbb{R}^+$
- $\mathcal{H} : \mathbb{R}^n \rightarrow \mathbb{R}^+$ where $\mathcal{H}(x) = \tau(x)$

Since this system evolves over \mathbb{R}^n , it is an infinite system, which is difficult to work with. An abstraction can be constructed to make the system easier to deal with. Abstractions are simplified expressions of a system that reduce the complexity of the original system by neglecting some details, i.e., a system \mathcal{S}_b is said to be an abstraction of a system \mathcal{S}_a if it retains some properties of \mathcal{S}_a while being simpler [20]. Abstractions can make it possible to analyse some properties of interest that the original system does not allow for. A finite-state abstraction maps a possibly infinite state-space to a finite number of states. An example of a finite-state abstraction is given by [28]. The authors are inspired by the work on SDSS described by [13]. SDSS is a form of STC that uses a mapping from state to IST. This mapping is computed offline to reduce the real-time computational load. One way to do this, as in [13], is by partitioning the state-space into a finite number of conic regions and finding an upper bound on the IST for each of these regions that will guarantee the stability of the system. An example of conic covering for a 2D system is shown in Figure 2-18. Upon retrieving a mapping, it suffices to find in which cone the state lies and look up which IST it is associated with, instead of having to dynamically compute the IST. [28] partitions the state-space into a finite number of convex polyhedral cones pointing at the origin, associates sampling times to these regions, and finds the transitions between discrete states (representing the conic regions) through reachability analysis. The result is a model of sampling traffic in the form of a timed automaton, which can be used for scheduling and formal analysis, e.g., in [19].

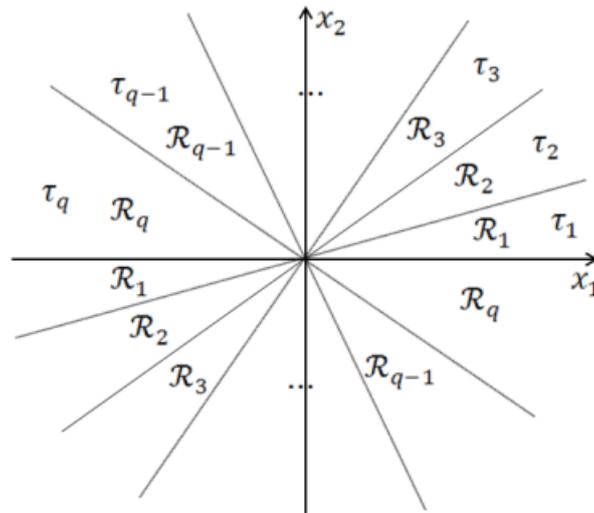


Figure 2-18: Example of conic regions for a 2D system [26].

Early triggering

In previous parts, some approaches to reducing the number of control actions have been described, yet they all fall under the same paradigm: sampling must be done as early as

possible. This greedy approach has no guarantees that it will result in optimal behaviour in the long run [20]. Figure 2-19 shows an example of why a greedy approach is not necessarily optimal. The numbers in the nodes represent the reward obtained when visiting that node. In the greedy case, a higher immediate reward is obtained, to the detriment of long-term pay-offs. Two metrics are important in assessing the performance of a control strategy: the

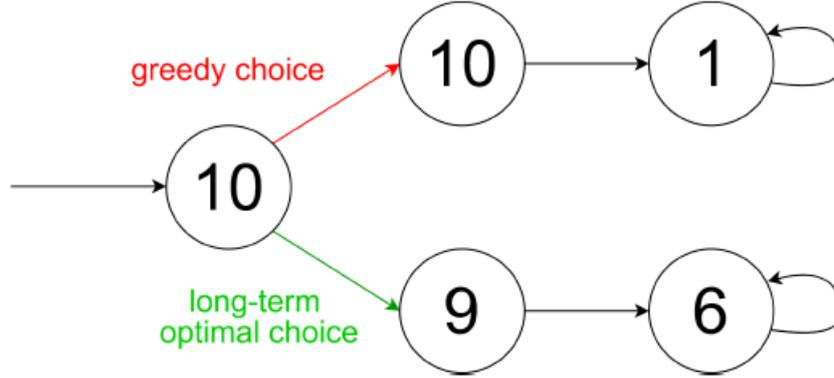


Figure 2-19: Greedy and long-term optimal strategy example [12].

average inter-sample time (AIST) and the smallest average inter-sample time (SAIST), a worst-case metric that denotes the smallest AIST across all initial conditions. The AIST is defined as

$$\text{AIST}(x, s) := \liminf_{n \rightarrow \infty} \frac{1}{n+1} \sum_{i=0}^n \tau_i(x)$$

and the SAIST is defined as

$$\text{SAIST}(s) := \inf_{x \in \mathbb{R}^{n_x}} \text{AIST}(x, s).$$

[20] considers the problem of maximising the AIST for a given control performance by considering the PETC-generated triggering times as deadlines. We defined the deadline $d : \mathbb{R}^{n_x} \rightarrow \mathcal{T}$ as

$$d(x) = \max \{ \tau \in \mathcal{T} \mid \mathcal{C}(\bar{\tau}, \xi_x(\tau), x) \},$$

where $\mathcal{T} = \{h, 2h, \dots, \bar{\tau}\}$. Sampling earlier than these deadlines always results in equal or better performance [20]. To do this, a finite-state abstraction of a PETC system is first made. This abstraction is then extended by allowing for controllable triggering times, up until the sampling deadline. A mean-payoff game is solved to find the strategy that optimises the AIST. [7] extended the description of a generalised transition system to include a weight function $\gamma : \mathcal{E} \rightarrow \mathbb{Q}$ to the transitions, giving the notion of a weighted transition system $\mathcal{S} = (\mathcal{X}, \mathcal{X}_0, \mathcal{U}, \mathcal{E}, \mathcal{Y}, \mathcal{H}, \gamma)$. An infinite transition system that models the triggering traffic and where the sampling times constitute controllable actions (up to the sampling deadline) is given by

$$\begin{aligned} \mathcal{X} &= \mathcal{X}_0 = \mathbb{R}^{n_x}, \\ \mathcal{U} &= \mathcal{Y} = \{1, 2, \dots, K\}, \\ \mathcal{E} &= \{(x, u, x') \mid hu \leq d(x) \text{ and } x' = \xi_x(hu)\}, \\ H(x) &= d(x)/h, \\ \gamma(x, u, x') &= hu. \end{aligned}$$

To capture the properties of this infinite system in a finite way, [20] starts with an l -complete system and adds early triggers. An l -complete system can predict the next l deadlines of a PETC system. The state-space is abstracted in terms of these deadline sequences. For a given sequence length l , [20] defines $\mathcal{R}_l \subseteq \mathcal{X} \times \mathcal{Y}^l$ for the relation $(x, k_1, k_2, \dots, k_l) \in \mathcal{R}_l$ if and only if

$$\begin{aligned} x &\in \mathcal{Q}_{k_1}, \\ M(hk_1)x &\in \mathcal{Q}_{k_2}, \\ M(hk_2)M(hk_1)x &\in \mathcal{Q}_{k_3}, \\ &\vdots \\ M(hk_{l-1})\dots M(hk_1)x &\in \mathcal{Q}_{k_l}, \end{aligned}$$

where M is the state transition matrix, and

$$\begin{aligned} \mathcal{Q}_k &:= \mathcal{K}_k \setminus \left(\bigcap_{j=1}^{k-1} \mathcal{K}_j \right) = \mathcal{K}_k \cap \bigcap_{j=1}^{k-1} \bar{\mathcal{K}}_j, \\ \mathcal{K}_k &:= \begin{cases} \{x \in \mathcal{X} \mid \mathcal{C}(\cdot) > 0\}, & k < K, \\ \mathbb{R}^{n_x}, & k = K. \end{cases} \end{aligned}$$

The matrices \mathcal{Q}_k contain the states that trigger at time instant hk . A transition from some state in \mathcal{Q}_σ to some state in \mathcal{Q}'_σ exists if, for a u that respects the deadline, $\exists x \in \mathbb{R}^{n_x}$:

$$\begin{aligned} x &\in \mathcal{Q}_\sigma \\ M(hu)x &\in \mathcal{Q}'_\sigma. \end{aligned} \tag{2-39}$$

Finally, [20] defines an abstraction that partitions the state-space into subsets defined by the subsequent l triggering instants of the PETC system. Transitions are determined by verifying reachability using Equation 2-39.

$$\begin{aligned} \mathcal{X}_l &:= \pi_{\mathcal{R}_l}(\mathcal{X}), \\ \mathcal{E}_l &:= \{(\sigma, u, \sigma') \in \mathcal{X}_l \times \mathcal{U} \times \mathcal{X}_l \mid u \leq \sigma(1), \exists x \in \mathbb{R}^{n_x} : \text{Equation 2-39 holds}\}, \\ H_l(k_1 k_2 \dots k_l) &:= k_1, \\ \gamma_l(\sigma, u, \sigma') &:= hu, \end{aligned}$$

where $\pi_{\mathcal{R}}(\mathcal{X}_a) := \{x_b \in \mathcal{X}_b \mid (x_a, x_b) \in \mathcal{R} \text{ for some } x_a \in \mathcal{X}_a\}$ represents the natural projection of \mathcal{X}_a onto \mathcal{X}_b . Further information and proofs can be found in [20]. Figure 2-20 and Figure 2-21 show the results of the numerical example taken from [20]. It can be seen that by allowing early triggering, the PETC bursts as in Figure 2-20 can be avoided. The running average IST is improved, while guaranteeing the same control performance as the reference PETC controller. Unfortunately, creating the abstraction is a computationally intensive process, and the method suffers from the curse of dimensionality. For small systems (e.g., three-dimensional), the problem can already be intractable. The next section aims to address that problem.

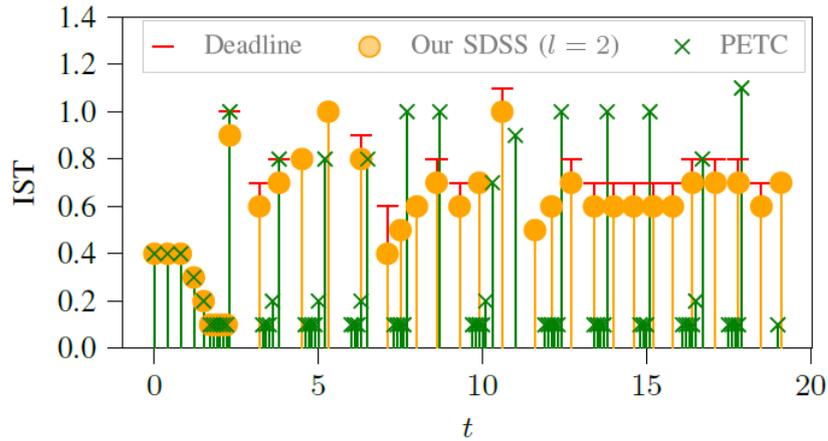


Figure 2-20: Comparison of triggering instances for PETC and SDSS ($l = 2$) [20].

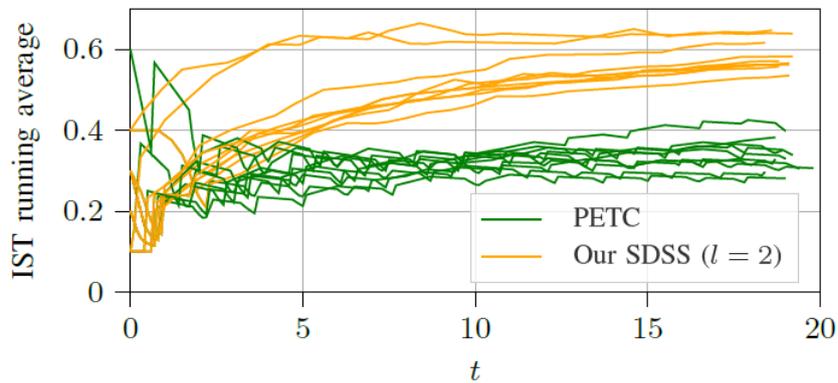


Figure 2-21: Comparison of running average of ISTs for 10 different initial conditions under PETC and SDSS ($l = 2$) [20].

2-2-5 Reinforcement learning tool

Motivation

As the depth of the abstraction increases, the number of possible transitions grows infeasibly large, quickly making the method impractical for systems with increasing dimensions and complexity. To tackle this, [12] describes a sample-based method for constructing the abstraction. Through step-by-step exploration, visited states only need to be held in memory temporarily and can later be discarded, which reduces the cost of increasing the look-ahead distance and hereby tames the curse of dimensionality. Furthermore, due to the iterative nature of a sample-based approach, the abstraction is improved gradually instead of requiring the completion of an exhaustive abstraction. Figure 2-22 illustrates this point.

Implementation details

Reinforcement learning (RL) is an area of machine learning concerned with how an agent should take actions in an environment to optimise for a reward. Formally, the reinforcement

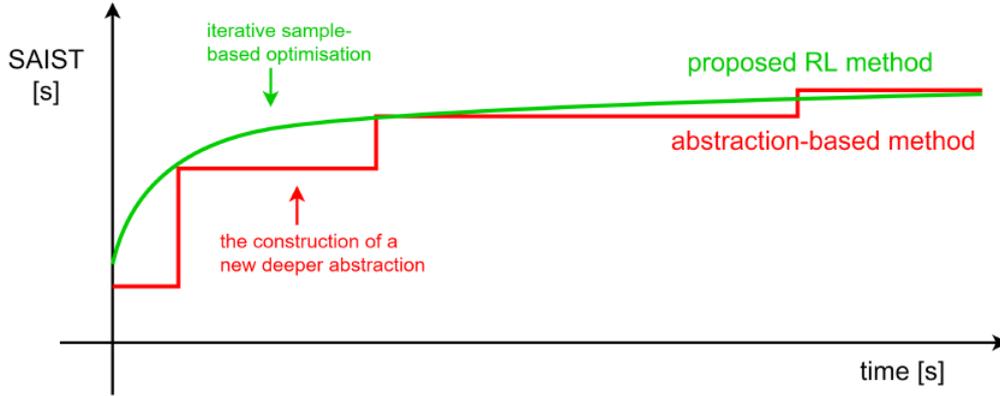


Figure 2-22: Sample-based optimisation lead to gradual improvements [12].

learning framework contains the following elements:

- An environment and agent states, S .
- The set of actions A the agent can take.
- $P_a(s, s') = \Pr(s_{t+1} = s' \mid s_t = s, a_t = a)$ is the probability of transitioning from state s_t to state s_{t+1} upon taking action a at time t .
- $R_a(s, s')$ is the reward after taking action a and thereby transitioning from state s to s' .

RL tries to find a policy π that optimises the expected reward. Two commonly used functions to evaluate the performance of a policy are the value function and Q -function. The value function is the expected reward upon following policy π from state s :

$$V^\pi(s) = \mathbb{E}[R \mid s_k = s, \pi] = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k r_{k+1} \mid s_k = s, \pi\right]$$

The Q -function is the expected reward upon taking action a from state s and subsequently following policy π .

$$Q^\pi(s, a) = \mathbb{E}[R \mid s_k = s, a_k = a, \pi] = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k r_{k+1} \mid s_k = s, a_k = a, \pi\right]$$

An optimal policy π^* gives an equal or greater reward when compared to all other possible policies. The optimal value function is given by

$$V^*(s) = \max_{\pi} V^\pi(s) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k r_{k+1} \mid s_k = s, \pi^*\right],$$

and the optimal Q -function is given by

$$Q^*(s, a) = \max_{\pi} Q^\pi(s, a) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k r_{k+1} \mid s_k = s, a_k = a, \pi^*\right].$$

Deep reinforcement learning (DRL) is a technique that combines reinforcement learning and deep learning. Often, the number of states DRL uses neural networks to approximate the value function, Q -function, and policy. An Artificial Neural Network (ANN) is a computational model inspired by the biological neural networks inside the brain. An ANN consists of an interconnection of artificial neurons. A model of an artificial is shown in Figure 2-23. Typically, an artificial neuron's parameters are its weights, bias, and activation function. The

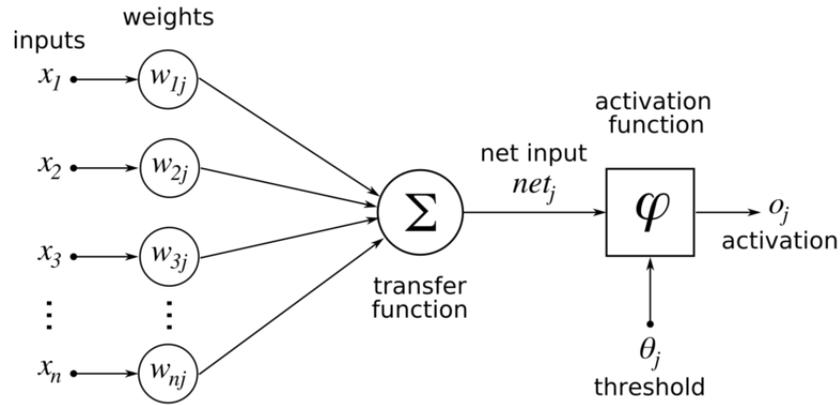


Figure 2-23: Inner workings of an artificial neuron [10]

output of this neuron is

$$o_j = \phi(W_j x + \theta_j).$$

The activation function maps the output to the range $[0, 1]$. A common activation function is the sigmoid function:

$$f(x) = \frac{1}{1 + e^{-x}}.$$

An ANN consists of a network of interconnected neurons, consisting of an input layer, an output layer, and several hidden layers between them. Using gradient descent, called back-propagation in the context of neural networks, the weights and biases of the neurons can be found such that the mean-squared error between the output of the ANN and some reference is minimised. Figure 2-24 shows an ANN with an input layer, output layer, and three hidden layers.

Learning environment

This section summarises how [12] implements a DRL tool to find a SDSS for LTI systems. The OpenAI Gym framework is a standard for formulating, developing, and testing problems to be solved by RL. The Gym framework sets up a learning environment as a class that requires the user to construct functions that relate the agent's state and action to its new state and reward upon taking the action. The framework requires a reset function and a step function. The reset function initialises or resets the environment and returns a first observation. The step function takes an action and returns the reward, new observation, and information on whether the goal is reached and further debugging information. Wrappers are a way to conveniently add functionality to an existing environment. They are commonly used to transform actions before applying them to the base environment, and observations

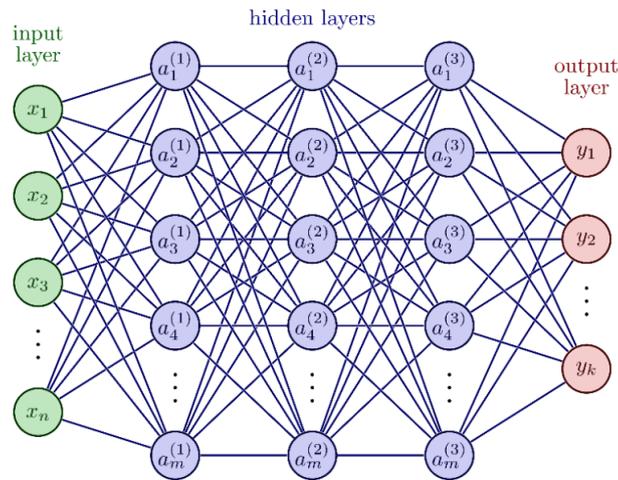


Figure 2-24: Neural network with an input layer, output layer, and three hidden layers [30].

and rewards returned by the base environment. [12] describes the following base environment and wrappers.

- **Base environment:** The base environment requires the system matrices A_d, B_d of the discrete-time plant and the controller gain K , the quadratic triggering function, sampling period h , and the minimum and maximum triggering times k_{\min} and k_{\max} .
- **Step function:** The step function calculates the next state when the control signal is held constant for k steps. The new state is then given by

$$x(t+1) = \left(A_d^k + B_d K \sum_{i=0}^{k-1} A_d^i \right) x(t).$$

The returned reward is given by kh .

- **Reset function:** The reset function resets the state to an initial point within a bounded region.
- **Deadline function:** The deadline function calculate the PETC sampling deadline for a given state.
- **Episodic wrapper:** Episodes are ended when they exceed a maximum episode length.
- **Action-penalisation wrapper:** For the agent to quickly learn not to exceed the PETC deadlines, the agent does not take the step when it selects an invalid deadline, and a negative reward is returned as a punishment.
- **Action-masking wrapper:** An action mask can further aid the agent in quickly learning valid deadlines. It does so by setting the generated action probabilities of invalid deadlines to zero and updating the probabilities. This is illustrated in Figure 2-25.

- **n-Spherical coordinates wrapper:** For quadratic triggering functions, states that lie on the same radial line have the same sampling deadline, since $\xi^T(t_k) Q \xi(t_k) > 0 \iff \lambda^2 \xi^T(t_k) Q \xi(t_k) > 0$. This fact can be used to reduce the search space. The wrapper transforms the Cartesian coordinates to n -spherical coordinates. The algorithm now only needs to look at a small environment around the origin. Furthermore, mirrored states have the same deadline; the search space can therefore be halved. The conversion from Cartesian to spherical coordinates is not unique. For example, [4] outlines a conversion from Cartesian coordinates to the so-called hyperspherical coordinate system. The approach used by the RL tool uses the coordinate system presented in [8]. The Cartesian and spherical coordinates are related as follows:

$$\begin{aligned}
 x_1 &= r \cos \varphi_1 \dots \sin \varphi_{n-1} \\
 x_2 &= r \sin \varphi_1 \dots \sin \varphi_{n-1} \\
 &\vdots \\
 x_{n-2} &= r \cos \varphi_{n-3} \sin \varphi_{n-2} \sin \varphi_{n-1} \\
 x_{n-1} &= r \cos \varphi_{n-2} \sin \varphi_{n-1} \\
 x_n &= r \cos \varphi_{n-1}
 \end{aligned}$$

- **Smoothness reward wrapper:** To avoid drastic trajectory changes upon sampling, the user can add a smoothness criterion to the reward function and make a trade-off between sampling performance and trajectory smoothness.

Algorithm choice

[12] selected Proximal Policy Optimisation (PPO) as the deep reinforcement learning algorithm of choice, as sampling efficiency is not of a priority (sampling is cheap in this simulation scenario), the ease of tuning and the possibility of running parallel workloads. The PPO implementation of Stable Baselines 2 was used for the implementation.

Neural network expansion

While a neural network with only one hidden layer can be proven to be a universal function approximator, having an appropriate number of layers can make it more easily capture the problem's complexity and speed up learning. [12] incrementally expands the neural network when learning stalls to avoid local minima, while avoiding the high computational loads that would accompany starting the initial training period with a large, fixed network.

Final evaluation

The final policy π_θ is evaluated by means of the SAIST and mean-average inter-sample time (MAIST) metrics given by

$$\text{SAIST}_{\text{approx}}(\pi_\theta) := \inf_{x \in X_M} \sum_{i=0}^N \tau_i(x, \pi_\theta),$$

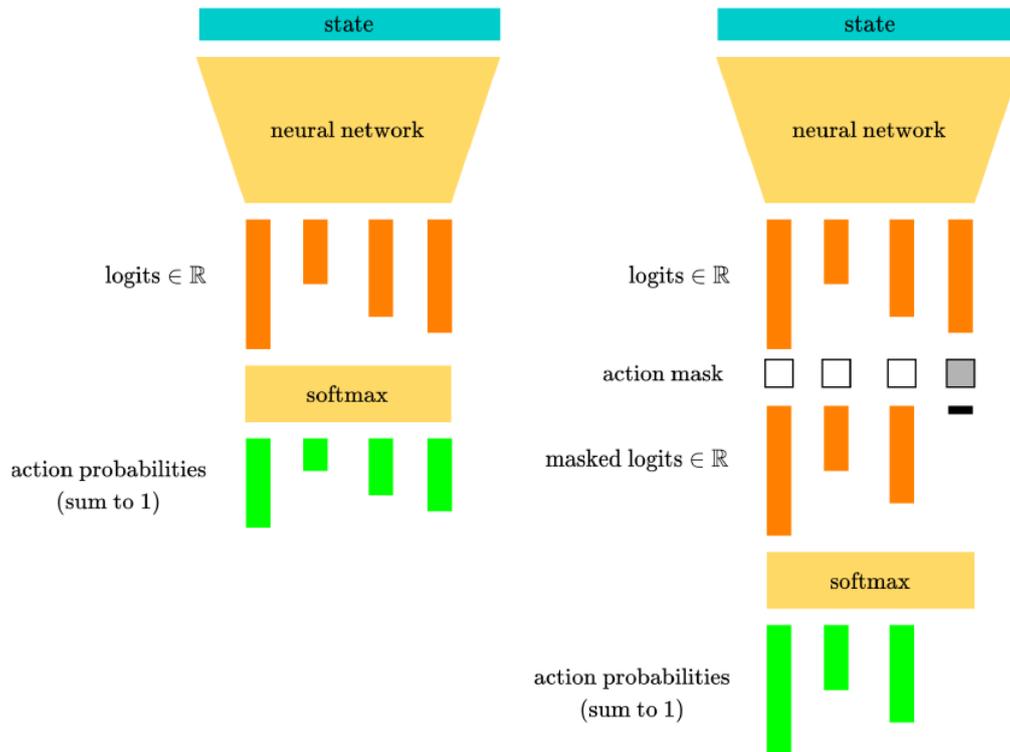


Figure 2-25: Schematic to demonstrate the working of action masking [12].

where X_M is a set of M different equidistant random points, and

$$\text{MAIST}_{\text{approx}}(\pi_\theta) := \frac{1}{M} \sum_{j=0}^M \sum_{i=0}^N \tau_i(x_j, \pi_\theta).$$

Figure 2-26 shows the result of the optimised SDSS controller for a 2D system. A great improvement of the AIST was found for this system.

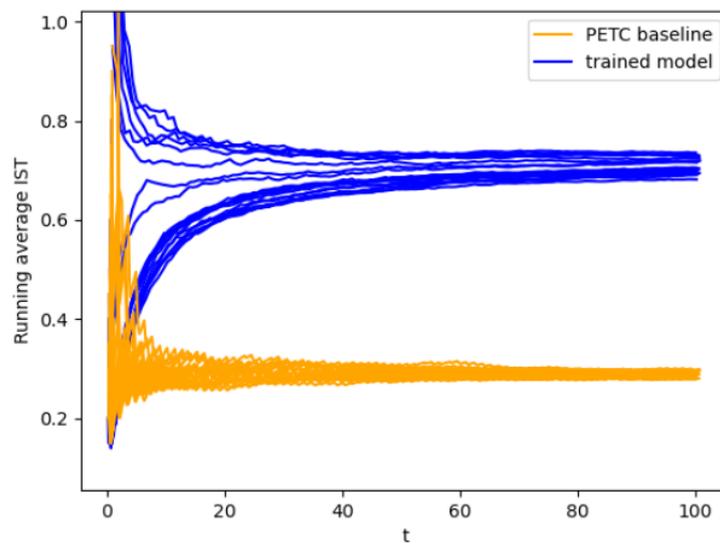


Figure 2-26: Running average IST simulation, comparing the trained model and PETC baseline.

Chapter 3

Simulations

This chapter describes the design choices for the controller and shows the results of computer simulations.

3-0-1 Controller design results

As a start, a train consisting of three wagons was used. The train parameters of Table 2-1 were used. A controller K was found using LQR that kept the actuator forces within a safety bound of 200 kN. This controller gain yielded a closed-loop system pole with an imaginary part of 2.4545. The frequency of oscillation is thus given by $f = \frac{2.4545}{2\pi} = 0.3906$ Hz. A common heuristic is to sample at ten times the highest frequency component. Rounding up the result gives a period of $h = 0.25$ s that is used to discretise the system. The following experiment was performed to assess the performance of the system.

- The references are: 100 km/h at $t = \{0 \text{ s}, 300 \text{ s}\}$, 50 km/h at $t = 100 \text{ s}$, 25 km/h at $t = 250 \text{ s}$.
- Zero-mean Gaussian noise with parameters $(\sigma_v, \sigma_{\Delta x}) = (10^{-3}, 10^{-5})$, where σ_v acts on the speed sensors and $\sigma_{\Delta x}$ acts on the distance sensors, affects the sensor readings.
- A disturbance of -20 kN at $t = 400$ s is applied.

Figure 3-1 shows the results of this experiment. The variables referred to in the legends are equivalent in subsequent experiment plots. The LQR controller parameters can be found in Table A-2 in the appendix.

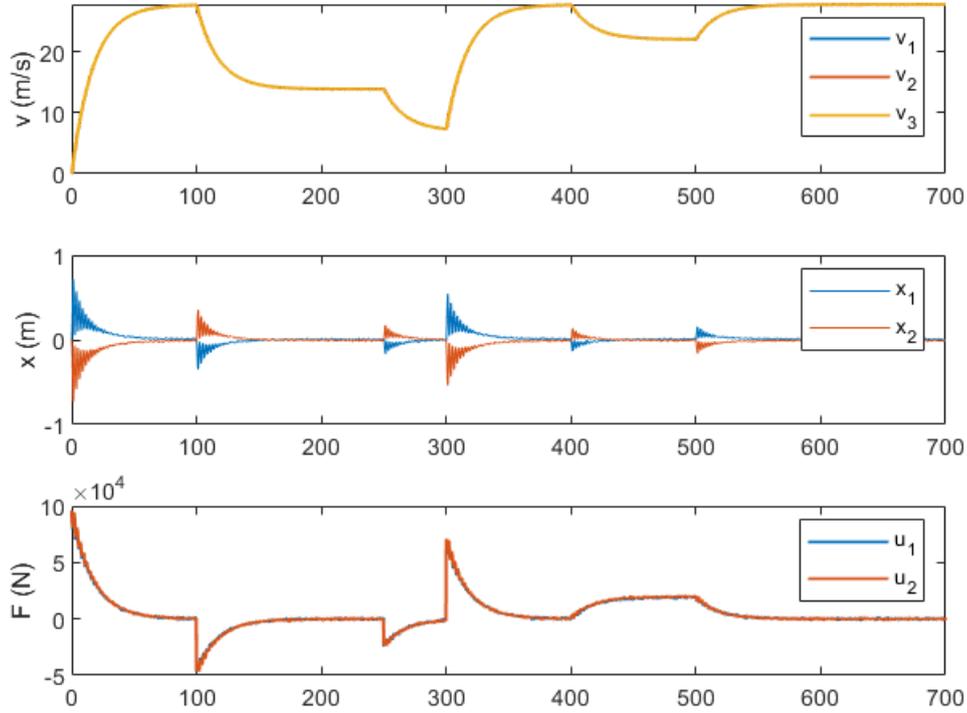


Figure 3-1: Periodic controller simulation.

For the PETC system, the triggering function is given by

$$\dot{V}(x) > -\rho x^T Q_{\text{Lyap}} x,$$

where

$$V(x) = x^T P_{\text{Lyap}} x.$$

This can be rewritten to

$$2(z^T P x) + \rho(x^T Q x) > 0$$

with

$$z = Ax + BK\hat{x}.$$

The quadratic triggering matrix Q for this triggering condition is equal to

$$Q = \begin{bmatrix} 2AP_{\text{Lyap}} + \rho Q_{\text{Lyap}} & 0 \\ 2K^T B^T P_{\text{Lyap}} & 0 \end{bmatrix}.$$

Two different PETC controllers were designed by selecting other PETC parameters. It was attempted to prove the stability of the PETC controllers by solving the LMIs in Corollary 1.1, but this was unsuccessful. Proving stability using the LMIs in Theorem 2 also did not succeed. However, the systems showed empirical stability. Figure 3-2 shows the results of PETC controller 1, which was designed to have a triggering condition that emphasises

controlling the inter-wagon distances. It can be observed that the trajectories do not differ much from Figure 3-1; the speed trajectories appear almost identical and the inter-wagon distances have a slightly noisier convergence pattern. The control parameters can be found in Table A-3 in the appendix.

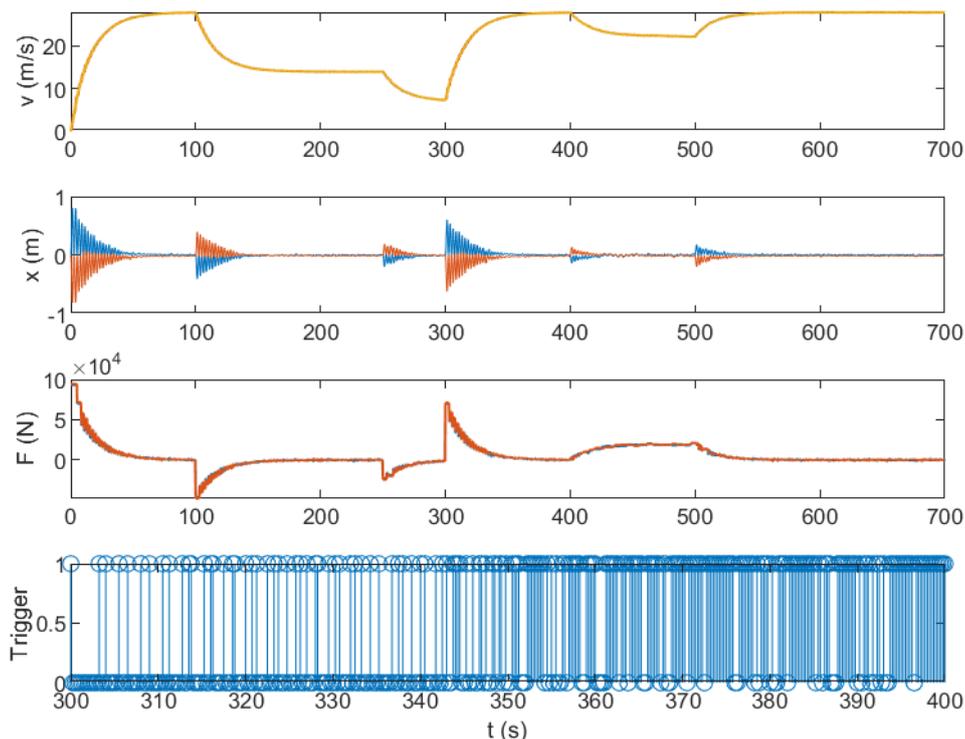


Figure 3-2: PETC controller 1 with emphasis on regulating the inter-wagon distances, $k_{\max} = 20$.

Figure 3-4 shows the simulations of PETC controller 2, which puts more emphasis on speed tracking. Without a maximum IST, a single trigger causes the train to quickly reach the desired speed setpoints, at the cost of great inter-wagon forces for a sustained amount of time. Figure 3-4 shows that enforcing a heartbeat can alleviate some of the excessive inter-wagon forces. This is because, for this system (PETC controller 3), triggering more often can improve the inter-wagon distances, but it will deteriorate speed tracking, as ideal (fastest) speed tracking can be achieved by applying maximal or minimal traction in a dead-beat control fashion.

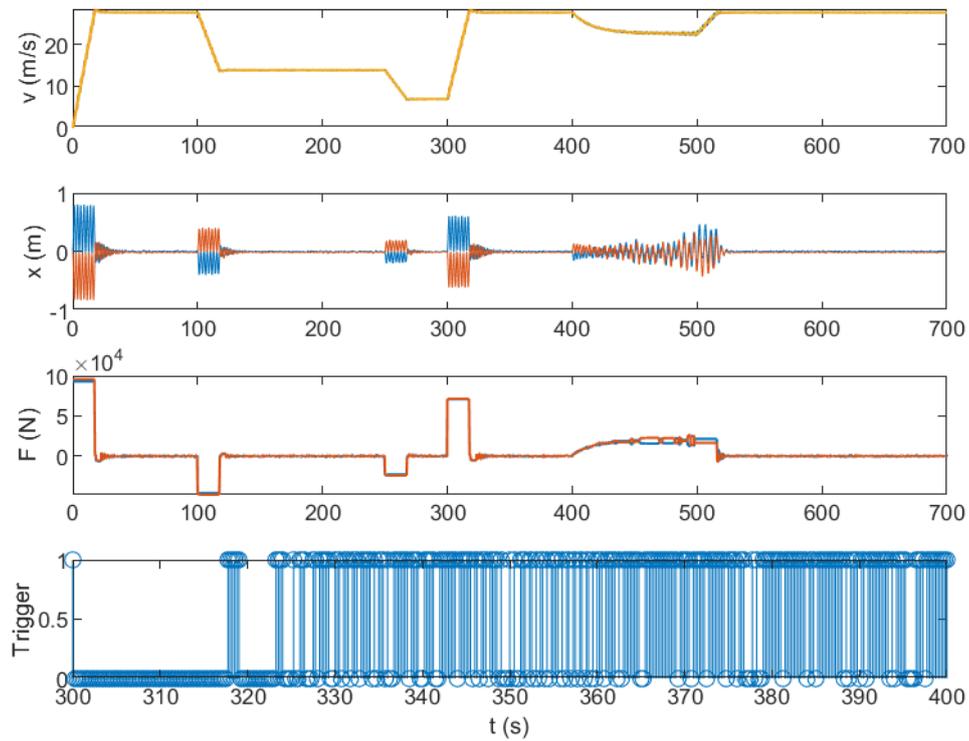


Figure 3-3: PETC controller 2 with speed emphasis, no max. IST.

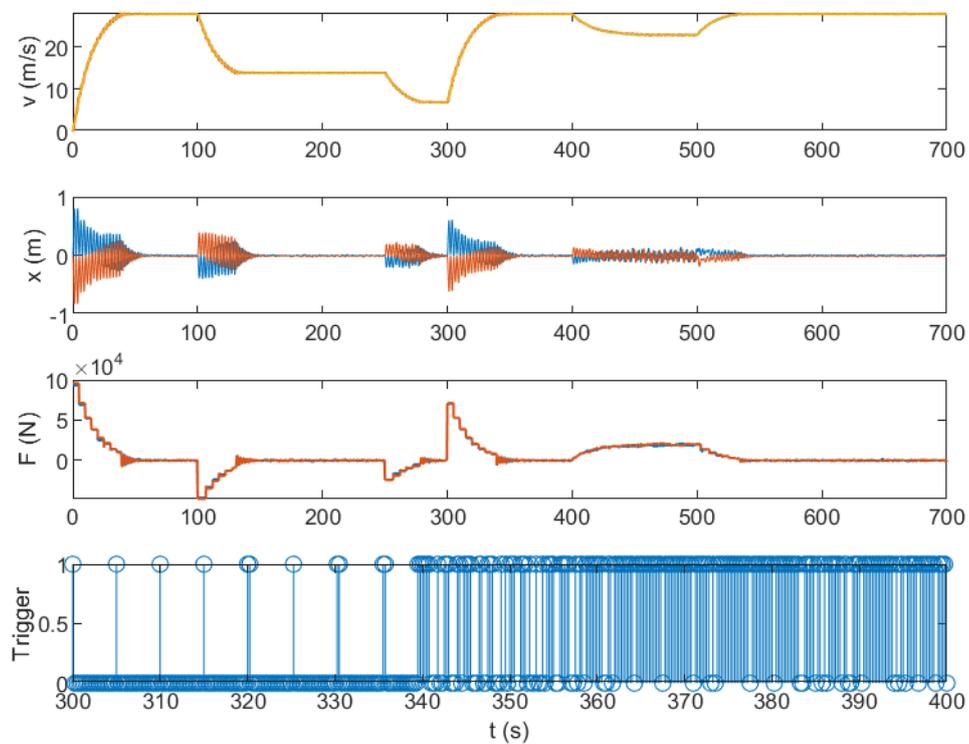


Figure 3-4: PETC controller 3 with speed emphasis, $k_{\max} = 20$.

Table 3-1 shows the number of samples and AISTs of these runs. An interesting observation is that while PETC 3 enforces a heartbeat, it leads to fewer overall samples. This shows that a greedy approach does not always yield fewer samples overall. The RL tool was used to find

Table 3-1: Periodic and PETC implementations evaluation.

	Periodic	PETC 1	PETC 2	PETC 3
Total samples	2800	1524	1595	1395
AIST (s)	0.25	0.4595	0.4391	0.5022

a more sample-efficient controller, taking the controller from Figure 3-3 as a baseline. The tool settings can be found in Appendix A-1. Figure 3-5 compares the IST performance of the PETC baseline and the trained model. It can be seen that no improvement over the PETC controller was found.

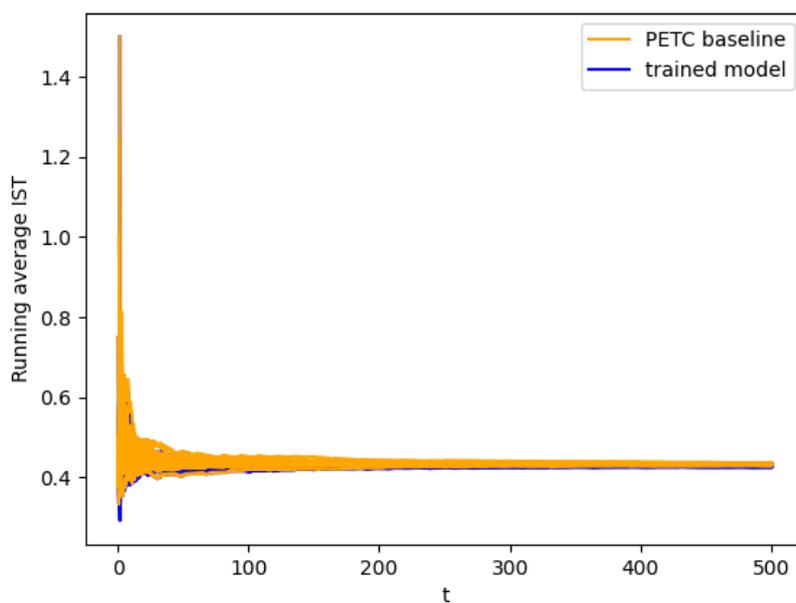


Figure 3-5: Final controller performance evaluation using 20 random initial conditions.

3-0-2 Evaluation

To understand why the trained SDSS does not improve on the performance of PETC, a metric that can be looked at is the *deadline distribution*. The deadline distribution shows which deadlines can be generated in a PETC scheme. The deadline distribution of the original PETC system is shown in Figure 3-6.

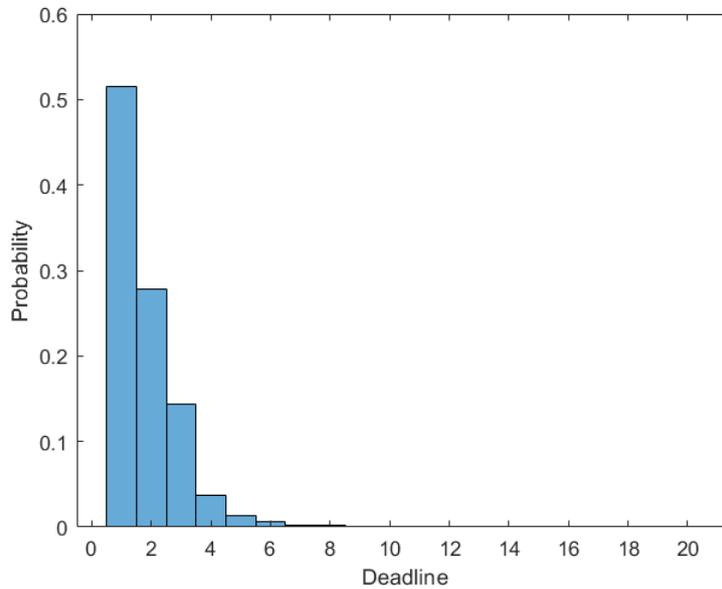


Figure 3-6: Deadline of 100,000 uniformly distributed $([-1, 1])$ random points.

It can be observed that the majority of the deadlines are small multiples of the fundamental checking period. Figure 3-7 shows the deadline distribution of the neural network SDSS, before enforcing the PETC deadline. The generated deadlines exceed that of the PETC deadlines shown in Figure 3-6. This means that the neural network deadlines will be ignored and overwritten by the PETC deadlines, i.e., no meaningful early-triggering SDSS was found.

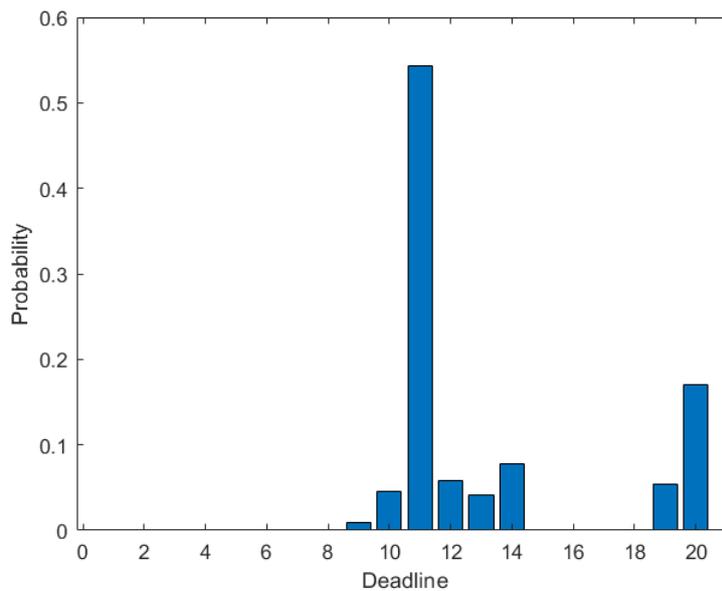


Figure 3-7: Neural network SDSS deadline distribution before enforcing the PETC deadline.

One possible explanation for why the neural network failed to train correctly is that the deadline distribution in Figure 3-6 is too unfavourable to improve the performance, as the majority of the deadlines are a small multiple of the fundamental checking period. Another possibility is that the training hyperparameters were not attuned to the problem's complexity. The simulations were run with only one CPU core on a virtual machine, which caused slow training. If better hardware were used, neural network expansion could occur faster, perhaps capturing enough system complexity to improve upon the PETC baseline. Under the current settings, neural network expansion often did not occur before meeting the expired time deadline. However, because training takes a long time and even suboptimal hyperparameters were expected to show some training improvement, it was concluded that the controller design choices made the system challenging to improve in any case. One possibility to enhance the trainability of the system is by decreasing the sampling period. This will likely improve the deadline distribution, as more triggering patterns will be possible. A new controller was designed with $h = 0.05$ s and different PETC parameters, which can be found in Table A-5 in the appendix. Figure 3-8 shows the deadline distribution of this new system. It can be observed that many more deadlines are possible.

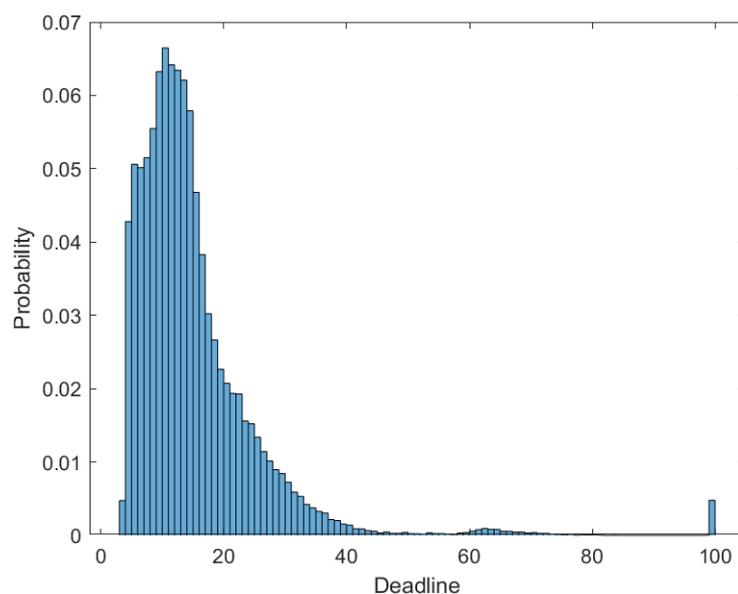


Figure 3-8: Deadline distribution of system with $h = 0.05$ s.

Figure 3-9 shows the training result for this new system. The training parameters are equivalent, except that training was performed over a time window of 10 hours instead of 5 hours. The other training parameters can again be found in Appendix A-1. It can be observed that there is a slight improvement over the PETC performance. Figure 3-10 shows the AIST and MAIST performance over time as the controller was trained.

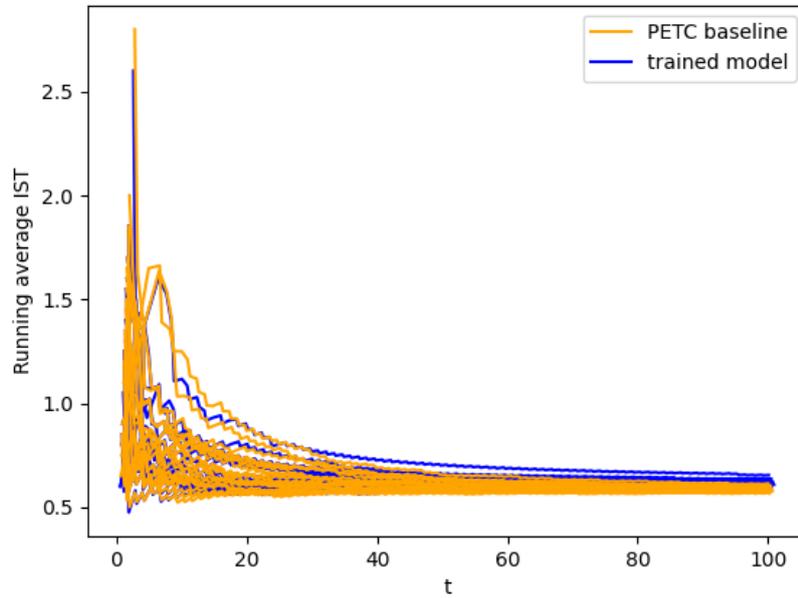


Figure 3-9: Final performance evaluation of new controller ($h = 0.05$ s) using 20 random initial conditions.

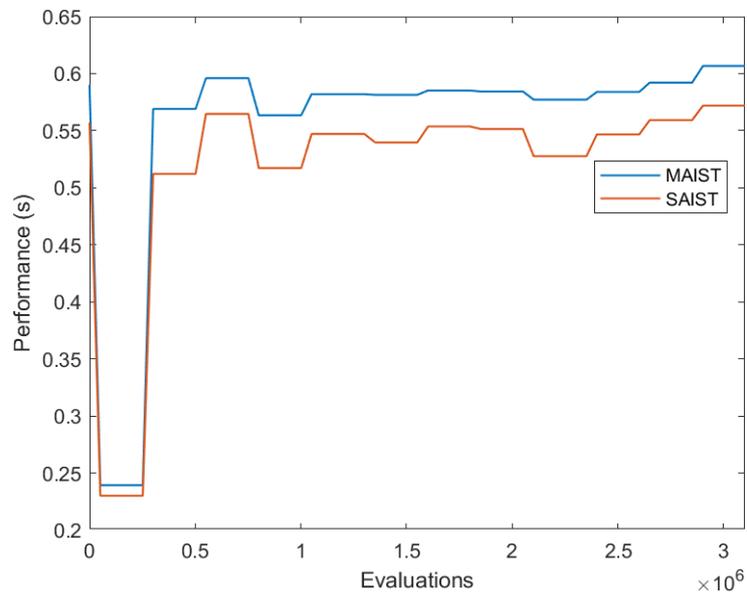


Figure 3-10: Training progress of new system ($h = 0.05$ s).

Figure 3-11 and Figure 3-12 show the deadlines around small deviations of the inter-wagon distances around the equilibrium, while fixing the velocities at some arbitrarily chosen points around the origin. It can be observed that there are certain points in the state-space where the result of the deadline calculation is very sensitive to minor deviations, which could arise

from noise.

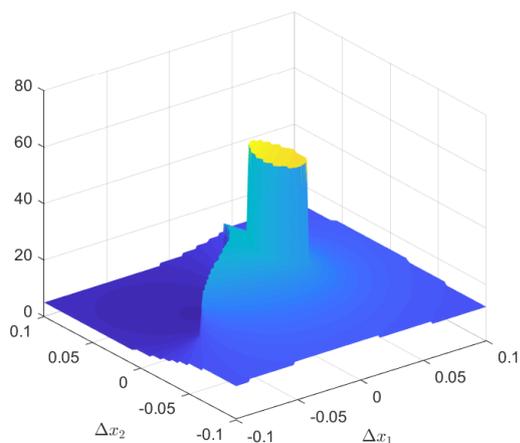


Figure 3-11: Deadline plot for $x_0 = [-0.05 \ 0.01 \ -0.05 \ \Delta x_1 \ \Delta x_2]^T$ with $(\Delta x_1, \Delta x_2) \in [-0.1, 0.1]$.

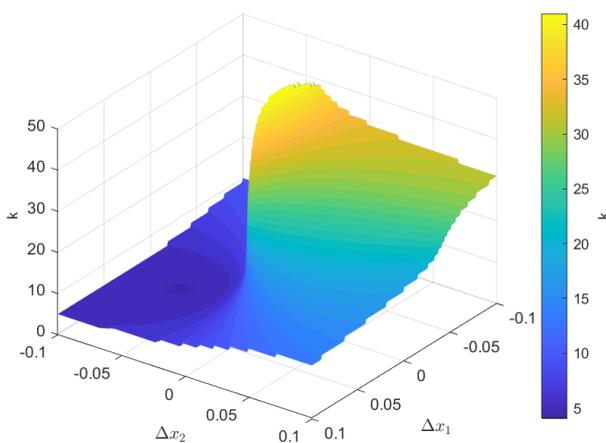


Figure 3-12: Deadline plot for $x_0 = [0.1 \ 0.04 \ -0.02 \ \Delta x_1 \ \Delta x_2]^T$ with $(\Delta x_1, \Delta x_2) \in [-0.1, 0.1]$.

Since the controller was found by training on the linearised system near equilibrium not subject to noise, the performance may suffer under real-world conditions. To study this, the early-triggering controller was compared to the controller that uses the PETC deadline every time. Figure 3-13 shows the AIST while running several iterations of the Virtual Serial experiment. It can be observed that the system is spending a significant amount of time switching between different equilibrium points and being subject to disturbances. It can be observed that the AIST performance of the early-triggering controller is worse than the reference controller. A likely explanation for this is that the system spends much time in regions of the state-space with suboptimal triggering patterns that it was not optimised for. A reference tracking experiment without disturbances but with noise enabled was performed to better study the long-term performance around the equilibrium. Figure 3-14 shows no observable performance gain.

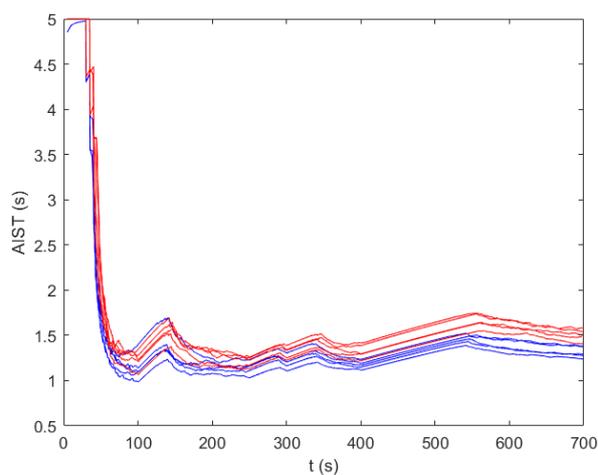


Figure 3-13: AIST during Virtual Serial full experiment.

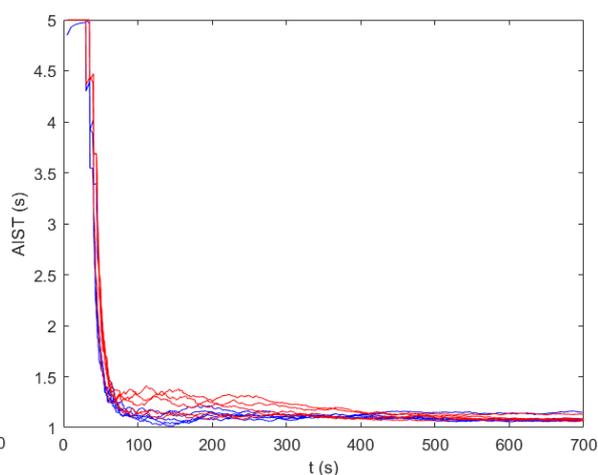


Figure 3-14: AIST during Virtual Serial reference tracking experiment with noise.

From these experiments, we can conclude that the optimised controller does not outperform the PETC controller when subject to real conditions. Further simulations running the SDSS controller in different experiments, comparing the computer and real-time machine runs, will be performed in Chapter 5.

HIL Design & Implementation

One of the contributions of this thesis is the implementation of a HIL environment to perform a real-time simulation of the system. The design of this setup, which comprises hardware choices, communication protocols, and the Simulink simulation setup, will be explained in this chapter.

4-1 Hardware

4-1-1 Firefly

The hardware development platform used in this thesis is the Zolertia Firefly platform, an IoT development platform based on the CC2538 System-on-Chip by Texas Instruments. This thesis makes use of the board's 802.15.4 radio support. The Firefly will be used for wireless communication between the controller and train, to send state and controller updates. Figure 4-1 shows the Zolertia Firefly board.



Figure 4-1: Zolertia Firefly board [25].

Contiki-NG

The software for the Firefly boards will be built around the Contiki-NG operating system (OS). Contiki-NG is an OS suited for resource-constrained IoT devices. The OS runs on various platforms, including the Firefly.

4-1-2 Speedgoat

Speedgoat is a company that develops real-time systems for real-time testing [32]. The hardware developed by Speedgoat is designed to interact with Simulink and Simulink Real-Time. The DCSC lab owns three Baseline real-time target machines [33]. Figure 4-2 shows the Speedgoat Baseline real-time target machine type that the university owns. Two Speedgoats will be used: one for running the train model and one for running the controller.



Figure 4-2: Speedgoat Baseline real-time target machine [33].

4-2 Communication

Data formatting

The states read from the model and the control signal and deadline computed by the controller need to be transmitted. A byte string can be constructed by appending the hexadecimal IEEE 754 double representation of the array elements to form an ASCII string with characters 0-F. As an example, if the state is given by $\begin{bmatrix} 1 & -2 & 3 \end{bmatrix}^T$, then the IEEE 754 representation of the individual elements is given by $\begin{bmatrix} 3FF0\dots0 & C0\dots0 & 40080\dots0 \end{bmatrix}^T$. The dots represent zeros up to the word length. The byte string is formed by appending these elements and is thus given by 3FF0...0C0...040080...0. The controller needs to send both a control signal and a deadline. Since the deadline is an integer, it need not be converted to the IEEE 754 format but can be appended as a digit string. Note that if a fixed data size is required, as is used here, the deadline field needs to have a fixed length and may need padding if the deadline takes up fewer characters than the largest possibility. For example, if the maximum deadline is 20, the deadline field takes up two characters; a deadline of, e.g., 5 would need padding to form 05 (other non-terminating characters could be used instead of 0). Note that more efficient representations are possible, but these will not be explored in this thesis.

Networking

RPL (Routing Protocol for Low-Power and Lossy Networks) is a routing protocol that lends itself well to low-power wireless networks subject to packet losses; however, it is a link layer agnostic protocol. RPL creates a directed acyclic graph (DAG) topology. The RPL standardisation is described in RFC 6550 [2]. Contiki-NG provides example code of a RPL client and server that uses UDP (User Datagram Protocol) as its communication protocol for the transport layer. UDP is suitable in applications where fast data transfer is required and the occasional packet loss is acceptable. The Contiki-NG example code has been adapted for the purposes of this thesis. The Firefly has four functions:

- **Receive wireless data:** The Firefly on the controller side receives a packet sent by the Firefly on the model side, or vice versa.
- **Send data to serial:** The packet is sent to the serial port. The Firefly is connected to a USB port.
- **Receive data from serial:** The packet is received from the serial port.
- **Send wireless data:** The Firefly on the controller side sends a packet to the Firefly on the model side, or vice versa.

4-3 System architecture

This section aims to provide an overview of the system architecture. It aims to provide a graphic Figure 4-3 shows a schematic overview of the system architecture. Two Speedgoats are used: one for simulating the train model and one for running the controller. The Firefly boards are connected to the Speedgoat through USB. The Fireflies communicate through UDP.

4-4 Physical setup

Figure 4-4 and Figure 4-5 show the physical setup in the lab. Figure 4-4 shows the front side of the Speedgoats, and Figure 4-5 shows the back side of the Speedgoats with the Firefly boards connected to the USB ports. The blue cables connect the Host link Ethernet port in Figure 4-5 to the Ethernet switch shown in Figure 4-6. Another Ethernet cable is connected from this switch to the host computer to retrieve the Speedgoat data after an experiment.

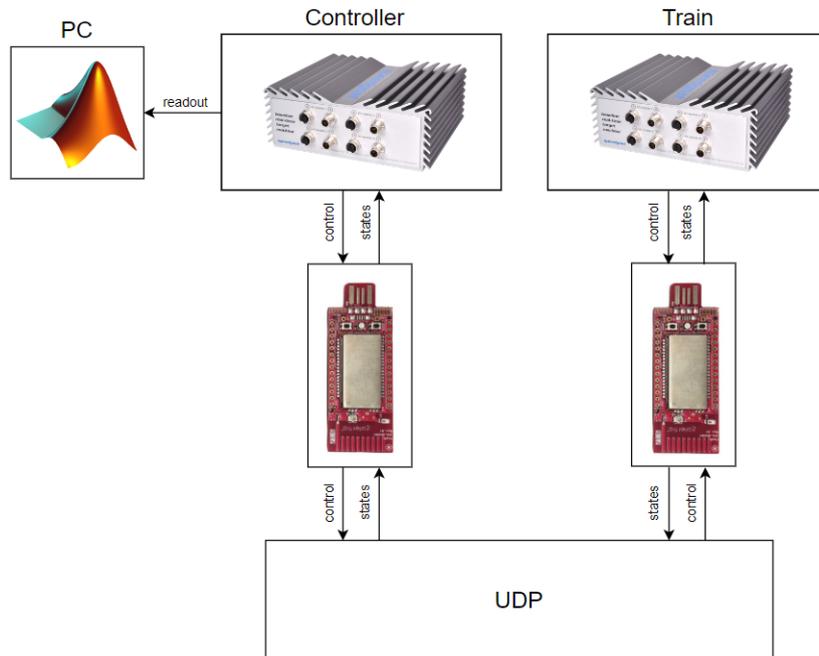


Figure 4-3: HIL setup with Speedgoats running the controller and train model, communicating through USB with the Zolertia Firefly boards that exchange information over UDP.



Figure 4-4: Speedgoats front view.



Figure 4-5: Speedgoats back view.

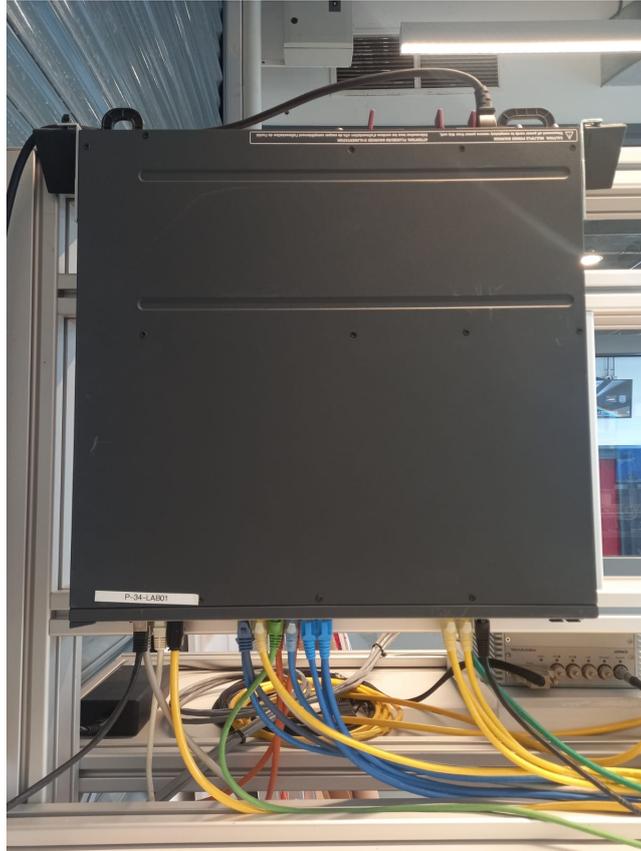


Figure 4-6: Ethernet switch in the DCSC lab.

4-5 Experimental setup

4-5-1 Experiment configuration

The experiments will be performed on four levels, progressively approximating a real system. The four levels are the following:

- **Simulink computer simulation:** The whole system runs on the computer and no real-time elements are added.
- **Virtual serial simulation:** The train simulation and controller run on the computer and communication between them is done through virtual serial ports.
- **Computer Firefly simulation:** The train simulation and controller run on the computer and communication is done with the Firefly boards, which are connected to the computer through USB.
- **Speedgoat simulation:** The train simulation is run on one Speedgoat, and the controller is run on another Speedgoat. Communication is done through the Firefly boards, which are connected to the USB ports of the Speedgoat.

Part of the experiment is to learn how the system behaves as it is exposed to situations that would occur in a real-life setting, such as a changing reference, disturbances, and sensor noise. Therefore, the simulation allows the user to configure these settings:

- **Reference setting:** The user can configure a reference profile. The reference profile is made up of step-wise references that are defined for a chosen time interval. The references must be part of a valid reference set: each reference must have a corresponding linearised system. The three references that are used in this thesis are 25 km/h, 50 km/h, and 100 km/h.
- **Disturbance setting:** The user can enable disturbances and configure a disturbance profile. The disturbance profile is made up of step-wise disturbances that are defined for a chosen time interval.
- **Noise setting:** The user can enable noise that acts on the "sensors" (i.e., the states that are read out) and configure the noise variances for the states. The noise type is zero-mean Gaussian white noise.

With these options, the following experiments will be performed:

- The experiment is run on four levels, progressively mimicking a real-time system.
- Noise and disturbances can be enabled or disabled independently, yielding four options.
- The references are: 100 km/h at $t = \{0\text{ s}, 300\text{ s}\}$, 50 km/h at $t = 100\text{ s}$, 25 km/h at $t = 250\text{ s}$.
- If noise is enabled, zero-mean Gaussian noise with parameters $(\sigma_v, \sigma_{\Delta x}) = (10^{-3}, 10^{-5})$, where σ_v acts on the speed sensors and $\sigma_{\Delta x}$ acts on the distance sensors, affects the sensor readings.
- When disturbances are enabled, a disturbance of -20 kN at $t = 400\text{ s}$ is applied.

4-5-2 Simulink simulation environment

The Simulink setup is divided into four parts: Model, Controller, Experiment Generator, and Logging. For the computer simulations, this setup resembles Figure 4-8. While each computer experiment (the first three in the list) has a unique Simulink file, the general architecture is the same. The Model part is responsible for reading the received serial data sent by the controller and extracting the control signal and deadline from the byte string in `Receive control`. This control signal is applied to the non-linear model, and the subsequent state is sent by the `Transmit states` block when the received deadline triggers. The Controller receives the byte string sent by the Model and extracts the states. The states are sent to the controller, which finds the deadline and controller signal. Finally, `Transmit control` formats the output of the controller block as a byte string and sends it through serial. The `Experiment generator` is responsible for providing the reference, disturbance, and noise signals, and the linearised system matrices corresponding to the current reference. The first three affect the model output, and the controller uses the system matrices to find the maximum deadline.

Finally, the Logging section collects and logs all the signals of interest for later inspection and processing.

The neural network policy network was exported using the Open Neural Network Exchange (ONNX) format, which can be imported into MATLAB/Simulink. Figure 4-7 shows the imported neural network, including preprocessing and postprocessing functions.

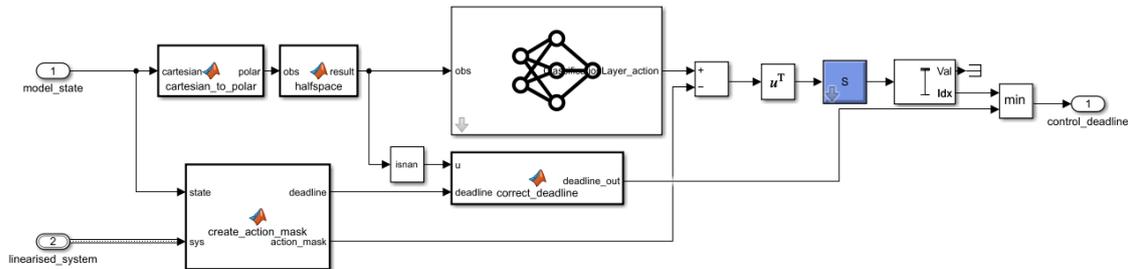


Figure 4-7: Neural network controller with preprocessing and postprocessing functions.

For the Speedgoat experiment, the Simulink model is divided into two models that will each run on their own Speedgoat. Figure 4-9 and Figure 4-10 show the Simulink model for the Controller and Model.

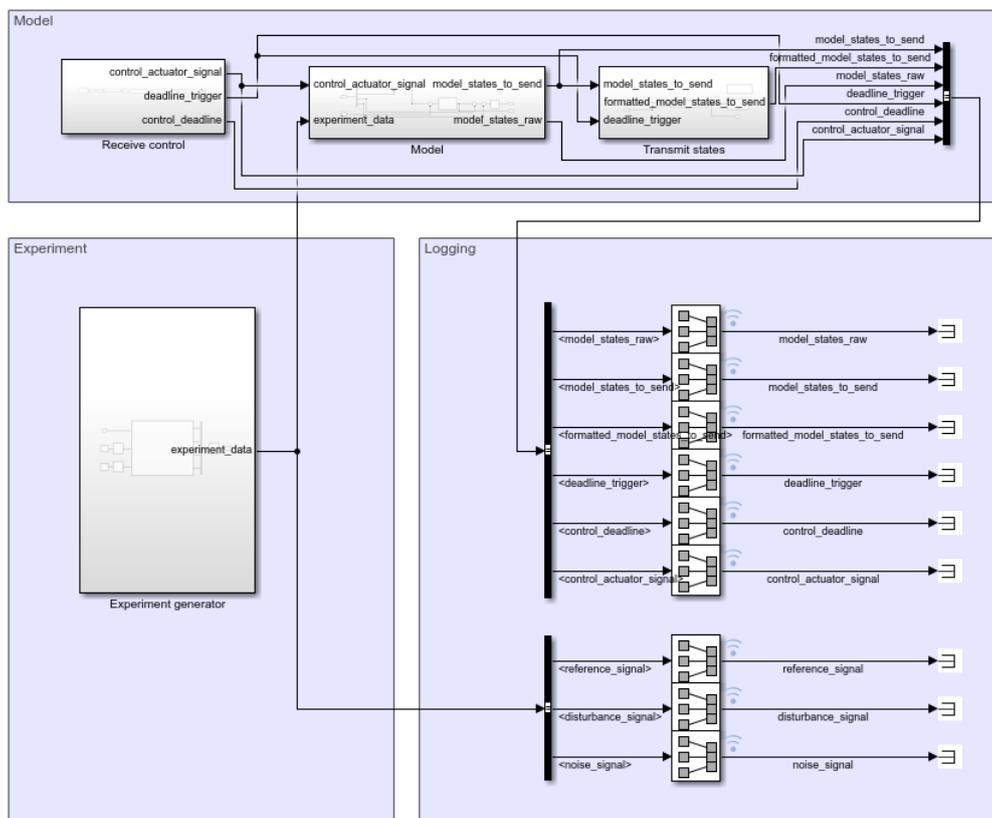


Figure 4-9: Simulink setup of the Speedgoat model side.

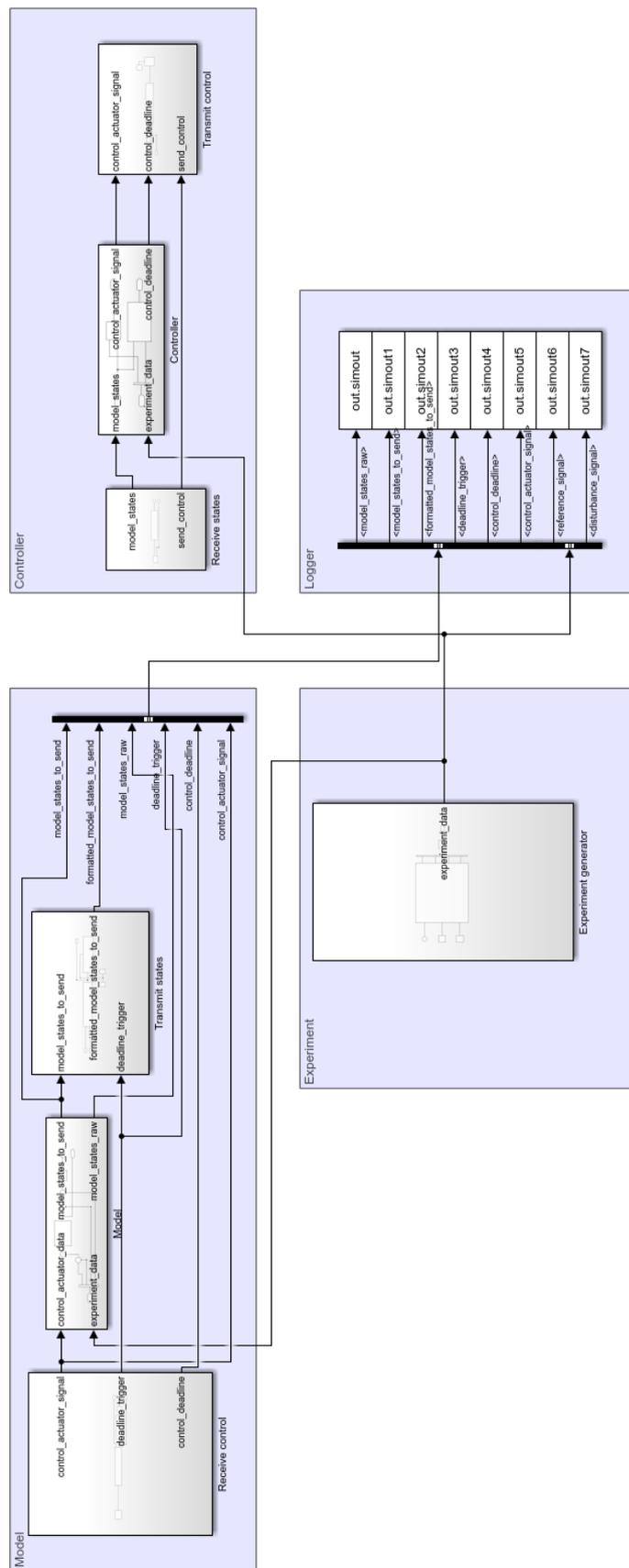


Figure 4-8: Simulink setup for Firefly experiment.

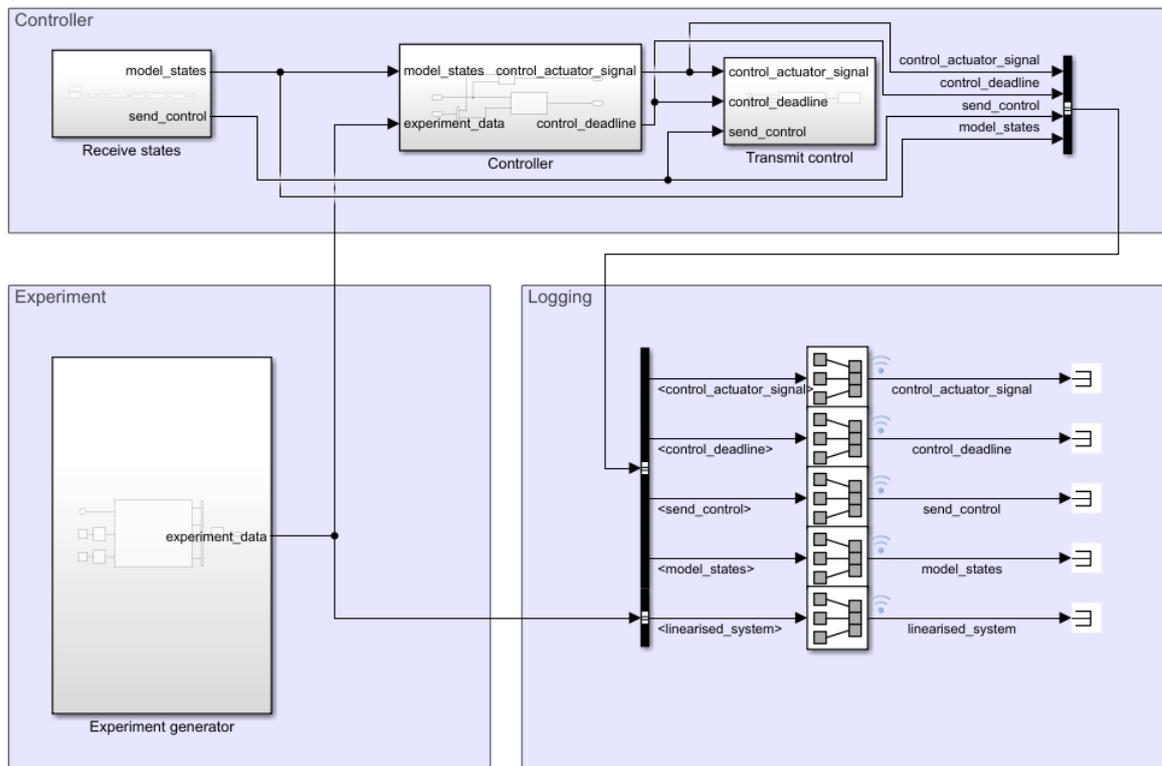


Figure 4-10: Simulink setup of the Speedgoat controller side.

Chapter 5

HIL Experiments

5-0-1 Experiment results

Experiments were performed to analyse the performance of the controller in the different experimental scenarios. Because running the controller at $h = 0.05$ s was not physically feasible due to communication delays (which will be studied in this chapter), the controller with $h = 0.25$ s of Figure 3-4 will be used to performance the physical experiments. Figure 5-1 shows an impression of a typical run on the Speedgoat. It can be observed that the system correctly performs reference tracking when no disturbance is present. Note that the plots showing the triggers and deadlines only show the data for the interval $t = [300 \text{ s}, 600 \text{ s}]$, as the most interesting data would otherwise be difficult to present. At $t = 400$ s, the controller is unable to cancel the disturbance. This is expected, as the controller would need to be augmented with integral action as in Equation 2-9 to cancel disturbances. However, the disturbance caused excessive oscillations in the inter-wagon distances, leading to large in-train forces.

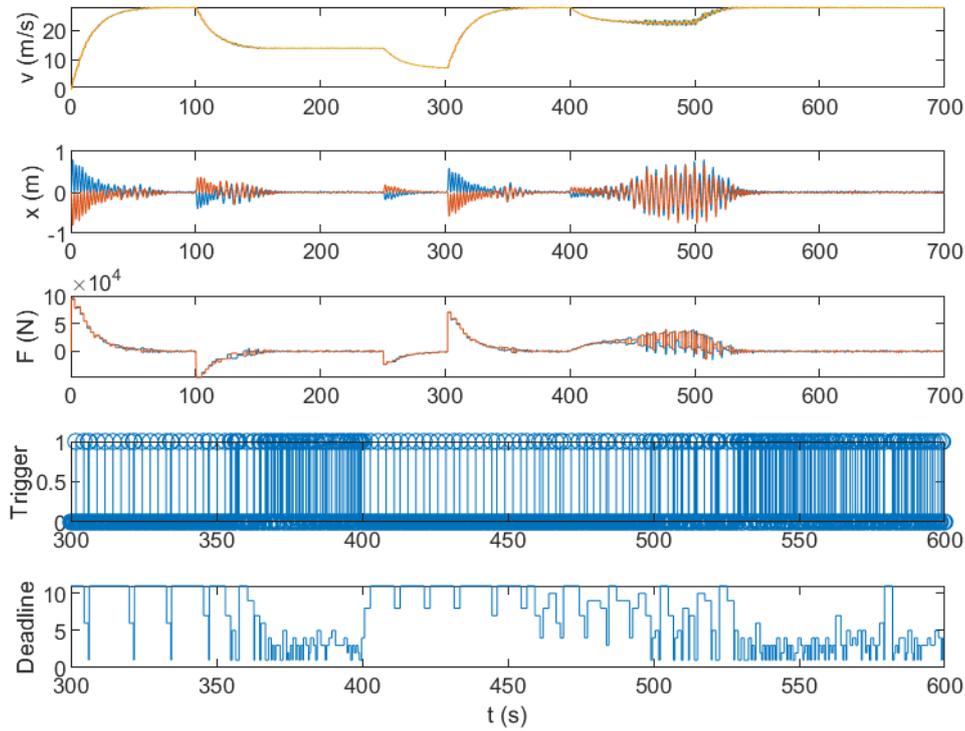


Figure 5-1: Speedgoat simulation result showing reference tracking, actuation effort, triggers, and deadlines.

To assess the controller's performance, all experiments as outlined in Section 4-5 were performed. This gave rise to a total of sixteen experiments. The following tables summarise the performance of the controller in terms of the total number of samples, the number of late packets, packet loss, and AIST.

Table 5-1: Total number of samples.

	Computer	Virtual Serial	Firefly	Speedgoat
\overline{ND}	1339	1270	1249	1246
\overline{ND}	993	977	950	944
\overline{ND}	864	894	995	930
ND	780	792	858	774

Table 5-2: Number of late samples.

	Computer	Virtual Serial	Firefly	Speedgoat
\overline{ND}	0	3	58	1
\overline{ND}	0	0	53	0
\overline{ND}	0	0	155	0
ND	0	0	64	0

Table 5-3: Number of lost samples.

	Computer	Virtual Serial	Firefly	Speedgoat
\overline{ND}	0	10	12	1
\overline{ND}	0	1	13	1
\overline{ND}	0	18	3	1
ND	0	1	7	1

Table 5-4: AIST for different experiments.

	Computer	Virtual Serial	Firefly	Speedgoat
\overline{ND}	0.5228	0.5516	0.5597	0.5622
\overline{ND}	0.7047	0.7172	0.7376	0.7420
\overline{ND}	0.8084	0.7839	0.7035	0.7535
ND	0.8953	0.8840	0.8142	0.9056

The most salient observation from these tables is that using the Firefly boards with the (non-real-time) computer led to many late packets. The Virtual Serial and Firefly experiments also suffer from many packet losses. The Firefly \overline{ND} experiment had an especially high number of late samples, but this was likely due to a fluke in the computer load, and not related to these noise and disturbance settings. It can also be observed that changing the settings has a significant effect on the AIST. An explanation of this is that different settings can push the system to spend a lot of time in some regions of the deadline distribution. Counter-intuitively, adding noise improved the AIST. Under noisy conditions, this system thus spends more time in favourable regions. Figure 5-2 and Figure 5-3 show the round-trip times of the Firefly and Speedgoat experiments (type ND). It can be observed that the distribution of the Firefly round-trip times has a longer tail, and that many packets exceed the $h = 0.25$ s receive deadline. On the other hand, the round-trip times of the Speedgoat appear to follow a normal distribution. Another observation is that the minimum round-trip time equals 0.05 s. This is because the Contiki-OS scheduler needs some time before the UDP process receives attention. In some cases, this is more than the time required for processing. The Contiki-OS kernel clock could be altered so that the scheduler checks for events at a higher rate. A downside of this is that it can increase overhead and power consumption.

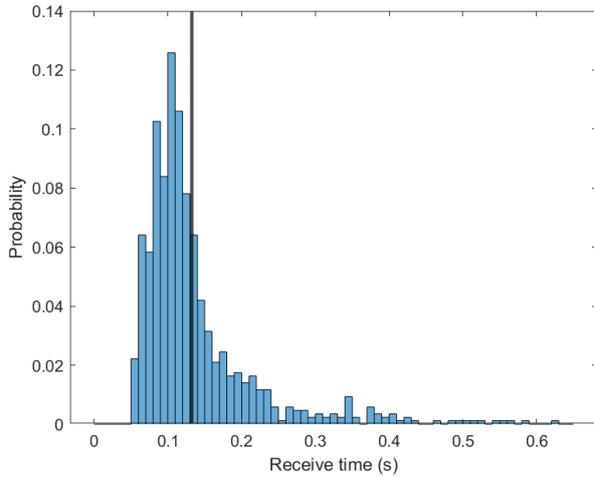


Figure 5-2: Round-trip times of Firefly experiment on computer. Mean = 0.1324 s.

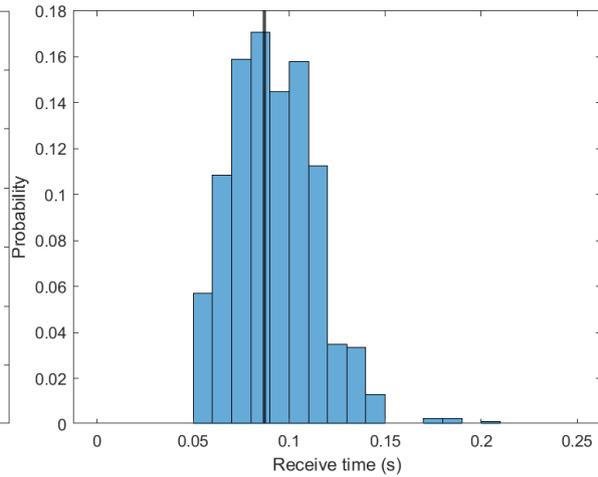


Figure 5-3: Round-trip times of Speedgoat experiment. Mean = 0.0870 s.

To further assess the impact of not using a real-time machine, a Virtual Serial and Firefly experiment was run under two qualitative conditions: low and high processor load. To create a higher load, various background processes (such as watching videos) were set up to run during the simulation. Note that this is meant to give a qualitative impression, as no formal load metric was measured. Table 5-5 and Table 5-6 show the result of this experiment. It can be seen that the number of packet losses increased when the computational load was high in both types of experiments. In the case of the Firefly experiment, the number of late samples also increased. Compared to a real-time machine, delays are more variable and the likelihood that deadlines are not obeyed increases. It was observed that this did not significantly affect the performance of the control loop; however, in a larger train, this could decrease the efficiency and cause higher in-train forces, reduced robustness to disturbances, and greater material wear and tear.

Table 5-5: Load experiment using Virtual Serial.

	Low load	High load
Total	681	825
Late	0	0
Lost	13	79

Table 5-6: Load experiment using Firefly communication.

	Low load	High load
Total	898	852
Late	24	48
Lost	5	23

Conclusions and Future Work

6-1 Overview

This thesis has attempted to construct a controller for a train system that minimises the number of samples required to attain a certain control performance. This kind of controller could make it possible to reduce the energy consumption of packet communication. In a heavy-haul train, the wagons are equipped with sensors powered either by batteries or an energy harvesting mechanism. In the scenario that motivates this thesis, it is important to reduce energy consumption to reduce the maintenance costs (replacing the batteries) or, when energy-harvesting sensors are used, achieve reliable sensor functioning (avoiding sensor downtime when energy consumption is excessive). First, different train models were considered. A non-linear train model was chosen that incorporates the inter-wagon dynamics, but ignores limitations such as actuator saturation and delays. This model was subsequently linearised and used to design a controller. In order to reduce the number of communication instances, aperiodic control mechanisms were employed. PETC and STC are two common methods to achieve this. To further optimise the number of control instants required to obtain a specified performance, the method of early triggering was used. Due to problems of scale, a sample-based approach was used. In this way, a neural network is trained to find a state-dependent sampling mechanism that outperforms a reference PETC controller in terms of inter-sample times while guaranteeing the same performance. It was found that using a controller with a sample time based on common heuristics (i.e., sample ten times the highest frequency) led to a controller of which no better early-triggering mechanism could be found. While this could partially be due to suboptimal training hyperparameters, a likely explanatory factor is the deadline landscape of the system: which sampling deadlines does the system generate, and could sampling earlier lead to improvements? To evaluate the controller's performance, four kinds of simulations of a progressively more real-time nature were used: computer simulations, simulations using virtual serial ports, simulations with wireless communications using Firefly boards, and a HIL setup using Speedgoat real-time target machines and Firefly communication. During experiments with the Speedgoat, a round-trip time of about 0.1 s was found. The simulations on the computer showed more variable communication delays, due to various

background processes that demand processor attention and no guarantee that deadlines are met, the signature of real-time. This demonstrates the value of using a real-time machine for simulations. To improve the chance of finding an early-triggering mechanism that improves the baseline performance, a controller with a better deadline distribution, a metric used as a proxy of potential, was found. In order to achieve this, the system's sampling period was decreased from 0.25 s to 0.05 s and other control parameters were used. Since this sampling speed could not be achieved with the real-time simulation, computer simulations were used to assess the performance of this new controller. First, a SDSS controller that marginally improves upon the PETC controller was found. However, using this controller under real conditions subject to noise and disturbances showed similar or worse performance compared to the baseline.

6-2 Future work

The three most significant problems that merit further study are robustness, scalability, and improving the accuracy of the HIL setup to reflect as much as possible a real system. These problems will now be briefly described, and suggestions for their resolution will be made.

6-2-1 Robustness

It was found that using the new controller in real-time conditions did not yield the expected performance improvement. Figure 3-11 and Figure 3-12 showed that some points in the deadline landscape are sensitive to minor deviations. The training of the neural network controller was done under the conditions of a linear system not subject to noise. To improve the robustness of the controller, noise must be considered during the training process. One way to do this is by considering the predicted deadline as a random variable, and training the system on its expected value. By considering a collection of random points around the state, the expected deadline can be found.

6-2-2 Scalability

In this thesis, a train of three wagons was considered. Clearly, this is not a heavy-haul train. Figure 3-9 shows the performance of the controller with $h = 0.05$ s. While the early-triggering controller shows marginal improvements over the reference PETC, it is expected that performance increases will be more modest as the system dimensions grow. Therefore, it is important to address the trainability of the system. One way to do this is to reduce the search space. The system was trained using random deviations from the equilibrium conditions, instead of focusing on the regions of the state-space the system dynamics are likely to visit. By only training the system on states encountered in real scenarios, the search space can be reduced. Furthermore, better hardware should be used to run the optimisation, as it is a computationally taxing task. Since a heavy-haul train can consist of many wagons, a complete model would likely be too large to optimise a controller for. Therefore, one possibility to improve the system's scalability is to use a hierarchical controller. In such a controller, the dynamics of multiple wagons (under the control of an optimised controller of a smaller order) would be abstracted away and considered a primitive to connect to other

wagons. A subsequent problem concerns the communication stack of the controller. During the experiments, all state information was sent by one Firefly. In reality, each wagon would send its own data, and the controller would thus have to receive state information from multiple sources.

6-2-3 Improving the HIL setup

To further assess the potential of the control techniques of this thesis for the control of heavy-haul trains, the HIL setup must be improved to more closely approximate a real system. A first important step towards that end is to allow each wagon to send data from its own transmitter, instead of having one transmitter, the Firefly board in this case, read and send out all states. While techniques based on ETC show theoretical improvements over periodic control in terms of sampling efficiency, one crucial question is how to integrate them into communication networks. To this end, a Wireless Control Bus (WCB) was developed to function as a network stack suited for ETC [37]. It is a novel protocol that supports multi-hop communication while minimising network overhead when control traffic is absent and ensuring swift collection and dissemination of sensor readings and actuator commands when called for. Figure 6-1 shows the HIL setup if WCB were used. The train could be modelled as a multi-hop network, corresponding to the situation where wagons sufficiently far away from each other cannot communicate directly.

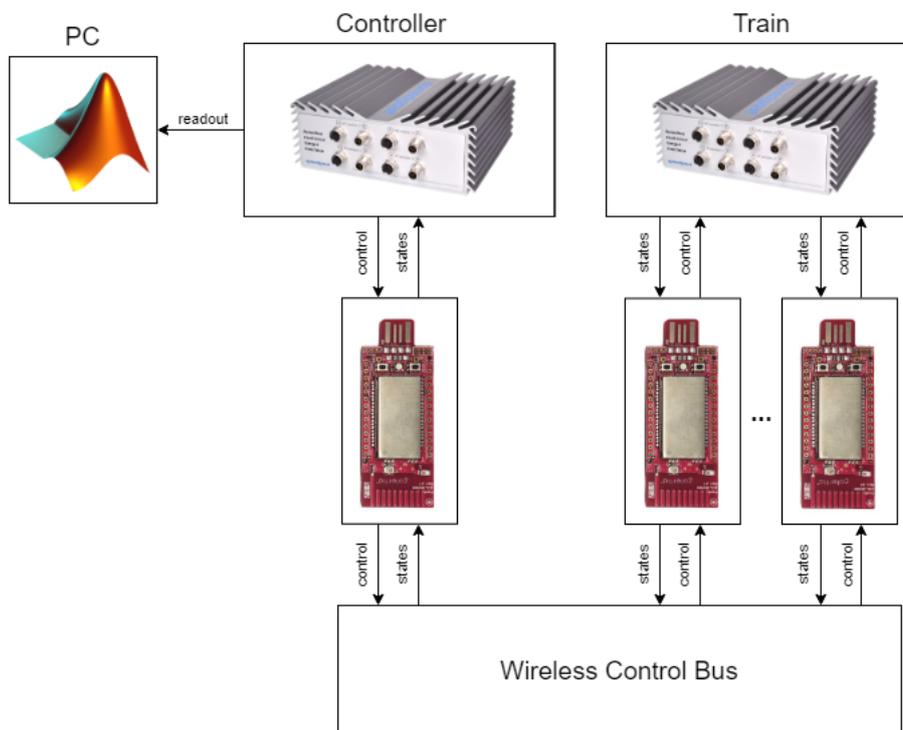


Figure 6-1: Improved HIL setup with Speedgoats running the controller and train model, communicating through USB with the Zolertia Firefly boards that exchange information over WCB.

To further reap the benefits of using aperiodic techniques, the radios of the sensor nodes should be turned off whenever possible to significantly reduce energy consumption. Figure 6-2 demonstrates the potential of using ETC to allow the radios to turn off when listening is not required. The example concerns a periodic controller with four control instances, where only the first and last are required to satisfy the desired control performance. When an STC controller, such as the SDSS controller presented in this thesis, is used, the radios can be turned off until the predicted STC deadline is reached. If these suggestions are implemented, a better evaluation of the potential of the presented techniques for the control of heavy-haul trains will be possible.

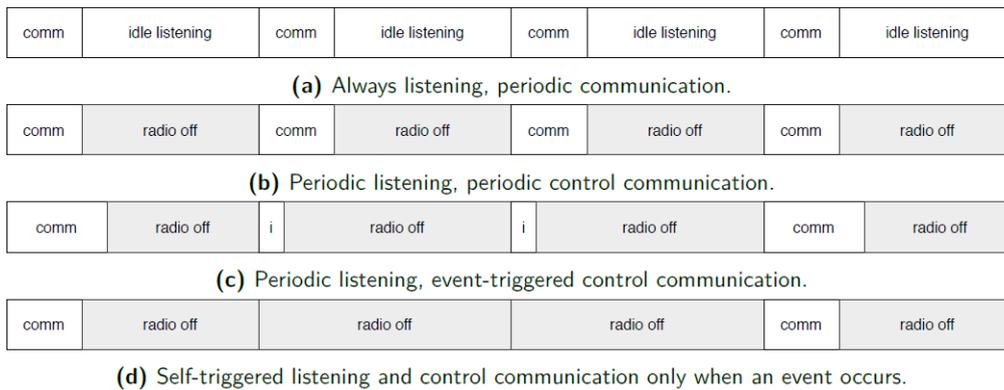


Figure 6-2: Radio savings made possible by ETC and STC [5].

Appendix A

Appendix

A-1 RL tool settings

The following tables show the settings of the RL tool for the different experiments.

Table A-1: RL experiment settings to find SDSS.

gamma	1
horizon_start	10
initial_lr	0.003
FIXED_RUNTIME	True
runtime	18000
TOTAL_TIMESTEPS	20000000000
CHECKPOINT_SAVE	False
checkpoint_steps	X
EVALUATE	True
eval_freq	5000000
n_timesteps_eval	1500
n_init_states_eval	750
SAVE_BEST_MODEL	True
stop_on_convergence	True
patience	30

start_check	0
tol	0
VECTORIZE	True
n_envs	2048
n_procs	1
SPHERICAL	True
halfspace	True
ACTION_MASK	True
SMOOTHNESS	False
beta	X
noptepochs	10
buffer_size	262144
minibatch_size	n_envs
ent_coef	0.01
vf_coef	0

max_grad_norm	0.5
lam	0.97
cliprange	0.2
LR_SCHEDULE	False
lr_sched_timesteps	20e6
final_lr	0.00001
HORIZON_SCHEDULE	True
horizon_freq	5e6
horizon_factor	2
horizon_max	200
EXPAND	True
lr_reduction_factor	3
lr_min	0.0001
horizon_increase_factor	2

VERBOSE	1
LOAD_MODEL	False
load_dir	user setting
TRAIN	True
TRAINED_MODEL_SAVE_FOLDER	user setting
TRAINED_MODEL_SAVE_NAME	user setting
PLAY	True
PLOT	True
n_timesteps	2000
n_init_states	1000
r	0.75
plotting_stepsize	X
n_traj_steps	X
n_traces	20
n_sim_steps	2000

EXPAND_MAX_TIMESTEPS	6000000
STOP_EXPANDING	False
n_stops	3
IMITATE	True
"n_timesteps"	20000
"batch_size"	64
"min_n_epochs"	100
"max_n_epochs"	300
"patience"	3
STANDARD_NETWORK_LIST	True
n_layers	14
width	1024
net_arch_list	[dict(vf=[64,64],pi=[64,64])]
trainable_layers_list	None
D2RL	False

A-2 Controller parameters

The following tables show the control parameters that were used during experiments.

Table A-2: Periodic LQR controller parameters.

n	3
Q	$\begin{bmatrix} 1000I_n & 0 \\ 0 & 100000I_{n-1} \end{bmatrix}$
R	0.0001
h	0.25
K	Solve LQR equation

Table A-3: Parameters of PETC controller with emphasis on controlling the inter-wagon distances.

n	3
Q	$\begin{bmatrix} 1000I_n & 0 \\ 0 & 100000I_{n-1} \end{bmatrix}$
R	0.0001
ρ	0.9
h	0.25
k_{\min}	1
k_{\max}	20
Q_{lyap}	$\begin{bmatrix} 0.001I_n & 0 \\ 0 & 0.8I_{n-1} \end{bmatrix}$
P_{lyap}	Solve Lyapunov equation
K	Solve LQR equation

Table A-4: Parameters of PETC controller with more speed emphasis.

n	3
Q	$\begin{bmatrix} 1000I_n & 0 \\ 0 & 100000I_{n-1} \end{bmatrix}$
R	0.0001
ρ	0.999
h	0.25
k_{\min}	1
k_{\max}	20
Q_{lyap}	$\begin{bmatrix} 0.01I_n & 0 \\ 0 & 0.1I_{n-1} \end{bmatrix}$
P_{lyap}	Solve Lyapunov equation
K	Solve LQR equation

Table A-5: Parameters of PETC controller with improved deadline distribution.

n	3
Q	$\begin{bmatrix} 1000I_n & 0 \\ 0 & 100000I_{n-1} \end{bmatrix}$
R	0.0001
ρ	0.3
h	0.05
k_{\min}	1
k_{\max}	100
Q_{lyap}	$\begin{bmatrix} 0.1I_n & 0 \\ 0 & 0.4I_{n-1} \end{bmatrix}$
P_{lyap}	Solve Lyapunov equation
K	Solve LQR equation

Bibliography

- [1] Dieky Adzkiya and Manuel Mazo Jr. Scheduling of event-triggered networked control systems using timed game automata. *CoRR*, abs/1610.03729, 2016.
- [2] Roger Alexander, Anders Brandt, JP Vasseur, Jonathan Hui, Kris Pister, Pascal Thubert, P Levis, Rene Struik, Richard Kelsey, and Tim Winter. RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. RFC 6550, March 2012.
- [3] David Arnold. Bhp iron ore. <https://www.flickr.com/photos/129177245@N05/18829393880>. Accessed on 2023-04-17.
- [4] Leslie E. Blumenson. A derivation of n-dimensional spherical coordinates. *American Mathematical Monthly*, 67:63, 1960.
- [5] Bas Boot. Reducing wireless control communication for a water irrigation system. Master's thesis, Delft University of Technology, 2021.
- [6] Dominicus P. Borgers, Victor S. Dolk, and Maurice W.P.M.H. Heemels. Dynamic periodic event-triggered control for linear systems. In *Proceedings of the 20th International Conference on Hybrid Systems: Computation and Control*, HSCC '17, page 179–186, New York, NY, USA, 2017. Association for Computing Machinery.
- [7] Krishnendu Chatterjee, Laurent Doyen, and Thomas A. Henzinger. Quantitative languages. *ACM Trans. Comput. Logic*, 11(4), jul 2010.
- [8] Jacky Chong. On the generalization of polar coordinates for n-dimensions [duplicate]. <https://math.stackexchange.com/questions/1958882/on-the-generalization-of-polar-coordinates-for-n-dimensions>. Accessed on 2023-03-22.
- [9] Ming-Shan Chou. Optimal cruise control of heavy-haul trains equipped with electronic controlled pneumatic brake systems. Master's thesis, University of Pretoria, 2007.
- [10] Wikimedia Commons. Diagram of an artificial neuron. https://commons.wikimedia.org/wiki/File:ArtificialNeuronModel_english.png, 2005. Accessed on 2023-03-13.

- [11] William J Davis. *The tractive resistance of electric locomotives and cars*. General Electric, 1926.
- [12] Robbert de Ruijter. Deep reinforcement learning for the synthesis of self-triggered sampling strategies. Master's thesis, Delft University of Technology, 2022.
- [13] Christophe Fiter, Laurentiu Hetel, Wilfrid Perruquetti, and Jean-Pierre Richard. A state dependent sampling for linear state feedback. *Automatica*, 48(8):1860–1867, 2012.
- [14] Adam Franco, Adam Hausmann, and Joseph Nazareth. Next generation distributed power: Activating the future of freight rail through enhanced communications.
- [15] GAO-17-122. Train braking: Dot's rulemaking on electronically controlled pneumatic brakes could benefit from additional data and transparency. <https://www.gao.gov/products/gao-17-122>. Accessed on 2023-04-04.
- [16] V.K. Garg and R.V. Dukkipati. *Dynamics of Railway Vehicle Systems*. Toronto: Academic Press, 1984.
- [17] Esther Geerts. The different energy harvesting technologies used in railways. <https://www.railtech.com/digitalisation/2022/04/26/the-different-energy-harvesting-technologies-used-in-railways/>. Accessed on 2023-04-04.
- [18] Antoine Girard. Dynamic triggering mechanisms for event-triggered control. *IEEE Transactions on Automatic Control*, 60(7):1992–1997, 2015.
- [19] Gabriel Gleizer and Manuel Mazo Jr. Computing the sampling performance of event-triggered control. pages 1–7, 05 2021.
- [20] Gabriel de A. Gleizer, Khushraj Madnani, and Manuel Mazo. Self-triggered control for near-maximal average inter-sample time. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 1308–1313, 2021.
- [21] Aleksandar Haber. Pole placement with integral control action to eliminate steady-state error (state-space control design). <https://aleksandarhaber.com/pole-placement-with-integral-control-action-to-eliminate-steady-state-error-state-space-control-design/>. Accessed on 2023-04-24.
- [22] Maurice Heemels. Event-triggered and self-triggered control. <https://heemels.tue.nl/research/event-triggered-and-self-triggered-control>. Accessed on 2023-04-19.
- [23] W. P. M. H. Heemels, M. C. F. Donkers, and Andrew R. Teel. Periodic event-triggered control for linear systems. *IEEE Transactions on Automatic Control*, 58(4):847–861, 2013.
- [24] Phil Howlett. Optimal strategies for the control of a train. *Automatica*, 32(4):519–532, 1996.
- [25] FIT IoT-LAB. Zolertia firefly. <https://iot-lab.github.io/docs/boards/zolertia-firefly/>. Accessed on 2023-03-13.

-
- [26] Arman Sharifi Kolarijani and Manuel Mazo Jr. Traffic characterization of LTI event-triggered control systems: a formal approach. *CoRR*, abs/1503.05816, 2015.
- [27] R.C. Kull. Wabtec ecp system update. In *Proceedings of the 2001 IEEE/ASME Joint Railroad Conference (Cat. No.01CH37235)*, pages 129–134, 2001.
- [28] M. Mazo Jr., A. Sharifi-Kolarijani, D. Adzkiya, and C. Hop. *Abstracted Models for Scheduling of Event-Triggered Control Data Traffic*, pages 197–217. Springer International Publishing, Cham, 2018.
- [29] Akshay Nanavati. Freight train. <https://unsplash.com/photos/8FevU1xdZC0>. Accessed on 2023-04-24.
- [30] Izaak Neutelings. Neural networks. https://tikz.net/neural_networks/. Accessed on 2023-03-22.
- [31] Reza Serajian, Saeed Mohammadi, and Asghar Nasr. Influence of train length on in-train longitudinal forces during brake application. *Vehicle System Dynamics*, 57(2):192–206, 2019.
- [32] Speedgoat. About speedgoat. <https://www.speedgoat.com/company/about-speedgoat>. Accessed on 2023-03-13.
- [33] Speedgoat. Baseline real-time target machine. <https://www.speedgoat.com/products-services/real-time-target-machines/baseline-real-time-target-machine>. Accessed on 2023-03-22.
- [34] Shuai Su, Liange Han, and Shukai Li. Finite-time event-triggered consensus control for high-speed train with gradient resistance. *Journal of the Franklin Institute*, 359(2):1144–1175, 2022.
- [35] Aleksandra Szymanek. Periodic event-triggered control with a relaxed triggering condition. Master’s thesis, Delft University of Technology, 2019.
- [36] Paulo Tabuada. *Verification and Control of Hybrid Systems: A Symbolic Approach*. Springer, 06 2009.
- [37] Matteo Trobinger, Gabriel De Albuquerque Gleizer, Timofei Istomin, Manuel Mazo, Amy L. Murphy, and Gian Pietro Picco. The wireless control bus: Enabling efficient multi-hop event-triggered control with concurrent transmissions. *ACM Transactions on Cyber-Physical Systems*, 6(1), 2022.
- [38] Ciann-Dong Yang and Yun-Ping Sun. Mixed h2/h cruise controller design for high speed train. *International Journal of Control*, 74(9):905–920, 2001.
- [39] X. Zhuan and Xiaohua Xia. Optimal scheduling and control of heavy haul trains equipped with electronically controlled pneumatic braking systems. *IEEE Transactions on Control Systems Technology*, 15(6):1159–1166, 2007.
- [40] Xiangtao Zhuan. Optimal handling and fault-tolerant speed regulation of heavy haul trains. 2007.

Glossary

List of Acronyms

3mE	Mechanical, Maritime and Materials Engineering
DCSC	Delft Center for Systems and Control
EEMCS	Faculty of Electrical Engineering, Mathematics and Computer Science
ECP	Electronically Controlled Pneumatic
iDP	Independent Distributed Power
PPD	Push-Pull Driving
DD	Distributed Driving
GES	Global Exponential Stability
PWL	Piecewise Linear
EMU	Electric Multiple Unit
NCS	Networked Control System
IoT	Internet of Things
IST	Inter-Sample Time
MIST	Minimum Inter-Sample Time
ETC	Event-Triggered Control
PETC	Periodic Event-Triggered Control
CETC	Continuous Event-Triggered Control
STC	Self-Triggered Control
SDSS	State-Dependent Sampling Strategy
AIST	Average Inter-Sample Time
SAIST	Smallest Average Inter-Sample Time
MAIST	Mean-Average Inter-Sample Time
RL	Reinforcement Learning
DRL	Deep Reinforcement Learning

ANN	Artificial Neural Network
LTI	Linear Time-Invariant
PPO	Proximal Policy Optimisation
LQR	Linear Quadratic Regulator
LMI	Linear Matrix Inequality
OS	Operating System
ONNX	Open Neural Network Exchange
HIL	Hardware-in-the-Loop
CPU	Central Processing Unit
UDP	Universal Data Protocol
RPL	Routing Protocol for Low-Power and Lossy Networks
DAG	Directed Acyclic Graph
WCB	Wireless Control Bus