Automatic robust controller synthesis

With application to a wet clutch system

R. Oskam

Master of Science Thesis

TU Delft Delft University of Technology

Delft Center for Systems and Control

Automatic robust controller synthesis

With application to a wet clutch system

Master of Science Thesis

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

R. Oskam

May 8, 2019

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

The work in this thesis was supported by Flanders Make. Their cooperation is hereby gratefully acknowledged.

Delft University of Technology
Department of
Delft Center for Systems and Control (dcsc)

The undersigned hereby certify that they have read and recommend to the Faculty of Mechanical, Maritime and Materials Engineering (3mE) for acceptance a thesis entitled

Automatic robust controller synthesis

by

R. Oskam

in partial fulfillment of the requirements for the degree of

Master of Science Systems and Control

Dated: May 8, 2019

Supervisor(s):

Dr. M. Mazo jr.

Ir. C.F. Verdier

Reader(s):

Dr. ir. M. Mazo jr.

Ir. C.F. Verdier

Prof. dr. R. Babuska

Abstract

A wet clutch is a device that transfers torque between two shafts via a hydraulic mechanism. Wet clutch control is key to achieve smooth and fast clutch engagements. Optimal control of a wet clutch is not trivial because of the complexity of the system due to nonlinearities, hybrid dynamics and changing dynamics over time due to changing temperatures and wear. Nowadays simple parametrized feedforward controllers are used in industry. The parameters of the control signal are tuned by hand and updated over time by an operator to account for the changing dynamics.

Several solutions to the wet clutch control problem exist in the literature, however the structure of the solutions is fixed beforehand and they rely on expert knowledge of the system. Another challenging aspect of wet clutch control is model uncertainty, currently robustness of performance is considered by running a finite number of experiments, but no hard guarantees can be given.

In this thesis a method is developed to automatically synthesize controllers for a wet clutch which are robust to model uncertainties. The method uses Genetic Programming (GP) to automatically synthesize controllers. Using GP controllers for a wet clutch can synthesize without fixing the structure beforehand and in a multi-objective way. Robustness is considered by optimizing the worst case performance.

To be able to give a guarantee on the worst case performance of a controller a method, that is able to formally guarantee a lower bound on the worst case performance using reachability analysis, is developed. This method is to costly to incorporate into GP and instead a use cheap estimation of the worst performance in practice.

This method was able to find robust controllers which outperformed a hand tuned baseline controller, but in order to compare this method to other methods in literature, real life experiments are needed.

Table of Contents

Acknowledgements

I would like to thank my daily supervisor Ir. C.F. Verdier and my supervisor Dr. ir. M. Mazo jr. for their assistance during the writing of this thesis.

I would like to thank the rest of Manuel's research group for their helpful input during the discussions we had.

I would like to thank Bruno Depraetere of Flanders Make for his help on the topic wet clutch control.

I would like to thank Prof. dr. R. Babuska for accepting the invitation to be on the graduation committee.

Delft, University of Technology R. Oskam
May 8, 2019

Chapter 1

Introduction

1-1 Wet clutch control

The transmission is a crucial part of any vehicle or machine where the power of the engine must be transferred to the load in a controlled manner. In case of a vehicle, a transmission is needed to transfer the power from the combustion engine to the wheels. The performance of the transmission has a significant influence on the fuel consumption and operator comfort of the system [1]. Wet clutches are mostly used for automatic transmissions in heavy duty machines like off-highway vehicles and agricultural machines [2]. They are suitable for these types of machines because of the need of a compact structure, with high efficiency and a reliable performance [1].

A wet clutch is a device that can engage and disengage two rotating shafts using a hydraulic mechanism. As can be seen in Figure 1-1, the clutch mechanism has two sets of friction plates. One set is connected to the input shaft and one set is connected to the output shaft. When no pressure is applied to these plates, they can slide freely inside each other. When pressure is applied to them, the friction between the plates will cause the two sets of plates to rotate together and hence it will cause the input and output shaft to rotate together. Pressure can be applied to the friction plates via a hydraulic piston. When the piston is extended far enough, it will start to press the two plates together. The piston can only be controlled indirectly via oil pressure. The piston is located in an oil chamber, where oil can flow in via the input valve. The electric current sent to the valve is the control input of the system. When the oil pressure in the chamber is high enough it will cause the piston to move forward, which in turn will cause the friction plates to be pressed together and finally the input and output shaft will be synchronized. When the oil pressure drops after a clutch engagement, the return spring will bring back the piston to its original position.

1-1-1 A clutch engagement

A clutch engagement starts with an engagement request from the operator during a gear shift. The engagement itself consists of two phases. The first phase is the filling phase, during this

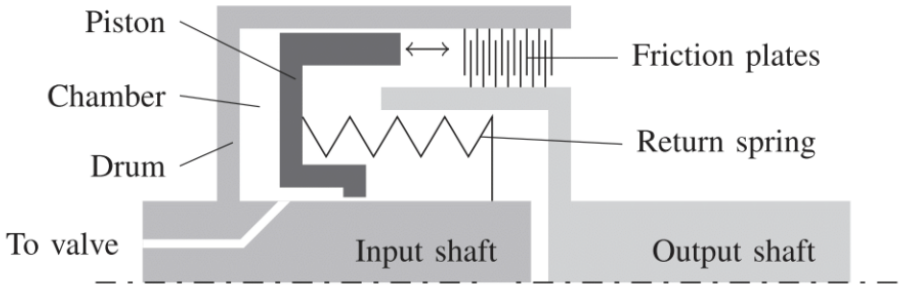


Figure 1-1: Schematic overview of a wet clutch and its main components [2].

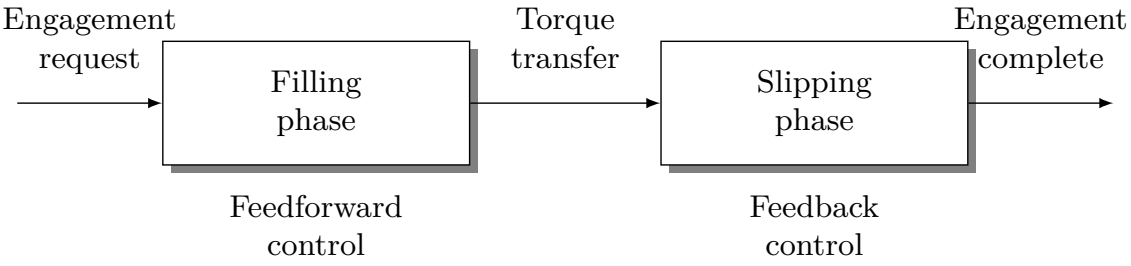


Figure 1-2: Schematic overview of a clutch engagement.

phase the oil chamber fills up with oil, the piston starts moving forward but, is not yet in contact with the friction plates and there is no torque transfer between the two shafts yet. Then, when the piston comes in contact with the friction plates and starts to press them together, the slip phase starts. Slip is defined as the angular speed difference between the input and the output shaft. When the slip is driven to zero, the shafts are synchronized and the engagement is complete. Both phases have different dynamics, making it a hybrid system. During the filling phase only feedforward control is used, while feedback control can be used in the slipping phase. See Figure 1-2 for a schematic overview of a clutch engagement.

1-1-2 Control objective

In general, the control objective for wet clutch control is to find a valve input that gives both a smooth and a fast engagement. A smooth engagement is desirable because it is strongly related to operator comfort [3]. The speed of a clutch engagement is measured by the engagement time t_{eng}, which is the time it takes from the clutch engagement request to the moment the clutch engagement is completed, i.e. the slip is driven to 0. The smoothness measure that is commonly used in literature for wet clutch control is the highest peak in the absolute value of the jerk [2]:

$$j_{\max} = \max_{t \in [0, t_{\text{eng}}]} |j(t)|. \tag{1-1}$$

Jerk is defined as the derivative of the torque of the output axis, $j(t) = \dot{\tau}(t, \xi(t))$.

Since this problem has two objectives, it is a multi-objective optimization problem. The two objectives are conflicting in nature, because there exists no controller that is optimal in both speed and smoothness. Instead, we can only find controllers which are a trade-off between the two. Typically there are two ways to tackle a multi-objective optimization problem [4]. The first approach is to convert the multi-objective optimization problem to a single-objective optimization problem. The most commonly used and most simple way to do this is by taking a linear combination of the objectives. The second approach is to solve the problem as a true multi-objective optimization problem and optimize both objectives simultaneously without expressing a preference for one objective. In this approach the optimal solution is not a single point, but the set of solutions which are optimal for each different trade-off. In general the latter option is more desirable because determining the trade-off a priori is not always trivial. Additionally the solution to the problem formulated as a single-objective optimization problem can always be reconstructed from the set of Pareto optimal solutions.

1-1-3 Challenges in wet clutch control

Several elements of the wet clutch system make controller design for it a complex task. First of all, the nonlinear dynamics in the two phases make the system a nonlinear hybrid system, for which standard control techniques do not suffice. Secondly the dynamics of the system change over time due to wear and changes in operating conditions, like oil temperature which are not modelled. Because of this it is not easy to verify that controllers perform well for all possible operating conditions.

1-2 Prior work and motivation

Nowadays in industry, parametrized feedforward controllers are used for wet clutch control. See Figure 1-3 for an example of such a control signal. The parameters of the signal are calibrated experimentally by the operator. The controller has to be recalibrated from time to time, to compensate for effects due to wear and changing operating conditions [5].

Several controllers for wet clutches have been developed in the literature in order to improve on the current mechanism of manual calibration. Because of the complexity of the system, most of them incorporate a learning mechanism in the controller design. The most notable model-based controllers in the literature are the Two-level Nonlinear Model Predictive Controller (2l-NMPC) and Two-level Iterative Learning Control (2l-ILC) [6]. They both use a control scheme consisiting of a high level and a low level controller. In the high level controller an optimal parametrized reference is learned and the low level controller is a reference tracking controller. The references are an oil pressure reference for the filling phase and a slip reference for the slipping phase. In both approaches Iterative Learning Control (ILC) is used in the high level controller to learn the optimal references. In 2l-NMPC an nonlinear MPC is used to compute the optimal tracking controller. In 2l-ILC the same learning method as in the high level controller is used in the low level controller. In [7] a reference-free iterative learning method for wet clutch control method is presented. This method is different from the previous two methods in that the controller is directly optimized over the engagement time and the maximum jerk. To be able to do this, linear models were used. Additionally other simplifications in the optimization problem were made based on insights in the system.

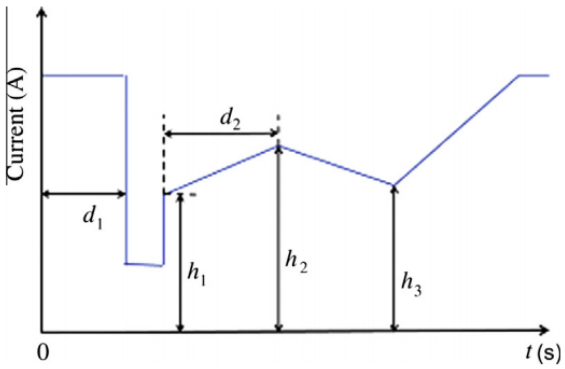


Figure 1-3: Example of a parametrized feedforward control with five tunable parameters [2].

These simplifications allowed to pose the problem as a convex optimization problem. Two other methods that have been applied to wet clutch control are Genetic Algorithms (GA) [2] and Reinforcement Learning (RL)[8]. Both of these methods work in a similar way in that they try to find the optimal parameters of a fixed-structure feedforward control signal like in Figure 1-3. They can be seen as a replacement for the operator who nowadays manually tunes these parameters. Both methods used scalar cost functions. All of the beforementioned methods were discussed and compared in [2]. In general the model-based techniques performed better and converged faster. The model-free techniques performed slightly worse, but also relied less on expert knowledge.

1-2-1 Current issues and open challenges

All discussed controllers require some form of **expert knowledge** of the system, for example in the design of optimal references or in the parametrization of the feedforward control signal. Furthermore all controllers except the iterative optimization technique (that uses only simple linear models) use either a reference or control signal with a **fixed structure**. Synthesizing controllers beyond these fixed structures could yield better results. The robustness of the performance of the controllers are verified by performing a finite number of experiments, however **no hard guarantees on the performance for different scenarios can be given**. The controllers discussed in this chapter fix the trade-off of the control objectives a priori. **No true multi-objective optimization** was used in these controllers. In [9] multi-objective GA was used for wet clutch control, however a different control objective (torque loss instead of maximum jerk) was used. The following open challenges in wet clutch control can be formulated:

- Synthesizing controllers for a wet clutch without relying expert knowledge,
- beyond fixed structures,
- in a true multi-objective manner,
- while giving guarantees on robustness of performance.

1-3 Our approach

Because we want to automatically synthesize controllers that are robust to model uncertainty, we will use feedback control in the slipping phase. In contrast to pure feedforward control which is commonly used in industry and in literature, feedback control can be respond to disturbances, because the input is computed based on the state of the system. We do not use feedback control for the filling phase because the piston position cannot be measured.

There exist several methods to automatically synthesize controllers for nonlinear hybrid systems with respect to a certain property. The two popular approaches are using abstractions and using Barrier and Lyapunov functions. In the first approach the system is transformed into a finite abstraction [10]. A drawback of this method is that controllers take the form of big look-up tables, while we are interested in finding controllers as compact expressions. Barrier and Lyapunov functions are used to prove safety and stability of systems [11]. Since the control objective of wet clutch control cannot easily be converted to such specifications, since no performance bound is known beforehand. Therefore it is not directly applicable for our purpose. In [12] Genetic Programming (GP) was used to synthesize candidate Control Lyapunov Functions (CLFs) for a system, from which a switched controller was derived. Stability and safety of the controller is guaranteed when the CLF properties of the corresponding candidate CLF are verified using an SMT solver. In this thesis we will also use GP to directly synthesize controllers for a wet clutch, instead of indirectly via CLFs. In our approach, we will consider robustness by optimizing over the worst case performance of the controller, which we will compute using reachability analysis.

GP is an evolutionary algorithm based on the principles of Darwin's theory of evolution. What distinguishes GP from other optimization and search methods is that is is able to optimize over both the function and parameter space, rather than just the parameter space. When applied to controller synthesis, this means that GP is able to find not only the optimal values of a fixed-structure controller, it can also optimize controller structures. Therefore GP can synthesize controllers directly without relying purely on expert knowledge. An additional advantage of GP is that it can perform true multi-objective optimization [4]. Instead of proving safety and stability as in [12], we will use formal verification methods to verify the robustness of performance of a found controller. Existing formal verification tools such as Flow* [13] are able to over-approximate the reachable set of complex uncertain systems. Using these over-approximations we can determine the worst case t_{eng} and $j_{\max}$ over all operating conditions. If we optimize over these worst case scenarios with GP, we will synthesize controllers that are robust for all operating conditions.

1-4 Problem formulation

The wet clutch is modeled by a hybrid system of the form:

$$\dot{\xi}(t) = f_i(\xi(t), u(t, \xi(t)), l) \quad \text{for} \quad \begin{bmatrix} \xi(t) \\ u(t, \xi(t)) \end{bmatrix} \in \Omega_i \tag{1-2}$$

where $\xi(t) \in X \subseteq \mathbb{R}^n$ denotes the state, $u(t, \xi(t)) \in \mathcal{U} \subseteq \mathbb{R}^m$ denotes the input and $l \in \mathcal{L} \subseteq \mathbb{R}^k$ denotes the uncertain parameter and $\mathcal{L}$ is a bounded set. We assume that l stays constant

during a clutch engagement: $\dot{l}(t) = 0$. The initial state is denoted by $\xi_0 = \xi(0)$. For $i \in \{1, \ldots, N_\Omega\}$, $\Omega_i \in \mathbb{R}^n \times \mathbb{R}^m$ are disjoint regions ($\Omega_j \cap \Omega_k = 0$, for all $j \neq k$), where $f_i(\xi(t), u(t, \xi(t)), l)$ is continuous for $\begin{bmatrix} \xi(t) \\ u(t, \xi(t)) \end{bmatrix} \in \Omega_i$.

A trajectory of a wet clutch is defined as follows:

Definition 1. *A trajectory of a wet clutch $\xi(t)$ is a function $\xi : [0, t_{\text{eng}}] \mapsto X$, such that $\dot{\xi}(t) = f_i(\xi(t), u(t, \xi(t)), l)$ for $\begin{bmatrix} \xi(t) \\ u(t, \xi(t)) \end{bmatrix} \in \Omega_i$, $\xi(0) = \xi_0$ and $\forall t \in [0, t_{\text{eng}}]$: $l \in \mathcal{L}$ and $u(t, \xi(t)) \in \mathcal{U}$.*

The slip $s(t)$ is defined as the difference between the input shaft speed $\omega_i(t)$ and the output shaft speed $\omega_o(t)$:

$$s(t) = \omega_i(t) - \omega_o(t). \tag{1-3}$$

The engagement time t_{eng} is defined as the first time instance where the slip is sufficiently small:

$$t_{\text{eng}} = \inf\{t \in \mathbb{R}^+ \mid |s(t)| < \epsilon\}. \tag{1-4}$$

We assume that $\forall t \geq t_{\text{eng}} : |s(t)| < \epsilon$. The time instance at which the system switches from the filling phase to the slip phase and the controller switches from feedforward control to feedback control is denoted by t_{switch}. We assume that only one switching instance occurs during a clutch engagement. The control input $u(t, \xi(t))$ is a control signal that is composed of feedforward part for the filling phase and a feedback part for the slip phase:

$$u(t, \xi(t)) = \begin{cases} u_{\text{ff}}(t), & t \in [0, t_{\text{switch}}) \\ u_{\text{fb}}(\xi(t)), & t \in [t_{\text{switch}}, t_{\text{eng}}] \end{cases}. \tag{1-5}$$

We define $t_{\text{eng}}^{u,l}$ as the engagement time related to a clutch engagement generated by a control input $u(t, \xi(t))$ and a realization of $l \in \mathcal{L}$. The worst case engagement time related to a control input u and a parameter set $\mathcal{L}$ is defined as:

$$t_{\text{eng}}^{u,\mathcal{L}} = \max_{l \in \mathcal{L}} t_{\text{eng}}^{u,l}. \tag{1-6}$$

The jerk j is defined as the time derivative of the torque:

$$j(t) = \dot{\tau}(t, \xi(t)), \tag{1-7}$$

where $\tau(t, \xi(t)) = T(\dot{\xi}(t), \xi(t))$, $T : \mathbb{R}^2 \mapsto \mathbb{R}$. The maximum jerk related to a clutch engagement generated by a control input $u(t, \xi(t))$ and a realization of $l \in \mathcal{L}$ is denoted by $j_{\text{max}}^{u,l}$ and is defined as:

$$j_{\text{max}}^{u,l} = \max_{t \in [0, t_{\text{eng}}^{u,l}]} |j(t)|. \tag{1-8}$$

The worst case maximum jerk related to a control input u and a parameter set $\mathcal{L}$ is defined as:

$$j_{\text{max}}^{u,\mathcal{L}} = \max_{l \in \mathcal{L}} j_{\text{max}}^{u,l}. \tag{1-9}$$

In this way we define the performance criteria as the worst case scenario over all possible values of the uncertain parameters. We then say that a controller u is Pareto optimal if and only if $\nexists a$ such that $(j^{a,\mathcal{L}}_{\max} < j^{u}_{\max} \wedge t^{a,\mathcal{L}}_{\text{eng}} \leq t^{u}_{\text{eng}}) \vee (j^{a,\mathcal{L}}_{\max} \leq j^{u,\mathcal{L}}_{\max} \wedge t^{a,\mathcal{L}}_{\text{eng}} < t^{u,\mathcal{L}}_{\text{eng}})$. Now we define the robust wet clutch control problem that we intend to solve as:

Problem 1 (Robust wet clutch control problem)**.** *Given a system (1-2), ξ_0, $\mathcal{L}$, ϵ and $\mathcal{U}$, find the set of controllers $u(t, \xi(t))$ that are Pareto optimal.*

Problem 1 can be divided into three subproblems with increasing complexity, so that the full problem can be solved step-by-step. The first step is to solve Subproblem 1, which does not consider uncertainty and where the only task is to synthesize feedback controllers for the slipping phase while fixing the control input for the filling phase. We denote this fixed, given feedforward input by $u_{\text{ff,baseline}}(t)$.

Subproblem 1. *Given a a system (1-2), ξ_0, $\mathcal{U}$, $u_{\text{ff,baseline}}(t)$ and $\mathcal{L} : |\mathcal{L}| = 1$, find the set of feedback controllers $u_{\text{fb}}(\xi(t))$ for the slip phase that are Pareto optimal using $u_{\text{ff}}(t) = u_{\text{ff,baseline}}(t)$.*

The second step is to add the task of also also finding the feedforward control input for the filling phase, while still not considering uncertainty. We call this Subproblem 2.

Subproblem 2. *Given a system (1-2), ξ_0, $\mathcal{U}$ and $\mathcal{L} : |\mathcal{L}| = 1$, find the set of controllers $u(t, \xi(t))$ that are Pareto optimal.*

In the final step we add the element of uncertainty by taking $\mathcal{L} : |\mathcal{L}| > 1$ and thus try to solve Problem 1.

1-5 Thesis outline

The thesis is structured in the following way:

- In Chapter 2, the dynamics of the model of the wet clutch that is used throughout this thesis is introduced and explained.
- In Chapter 3, the topic of Genetic Programming is introduced. Furthermore the application of GP to wet clutch controller synthesis is explained.
- In Chapter 4, we explain how reachability analysis is used to find a formal guarantee on the performance of a controller in our method.
- In Chapter 5, the problem of finding a feedback controller for the slipping phase for a system without uncertainty is considered.
- In Chapter 6, the problem of synthesizing both a feedforward and a feedback controller for the system without uncertainty is considered.
- In Chapter 7, the full problem of finding a feedforward and a feedback controller for a system with uncertainty is considered.
- In Chapter 8, the conclusions of the thesis and the recommendations for future research are presented.

Chapter 2

The wet clutch model

In this Chapter we introduce and explain the nonlinear, hybrid model that is used throughout this thesis to simulate the wet clutch.

2-1 Setup

The model we will use is described in [14]. This model was designed to simulate an experimental clutch setup, which can be seen in Figure 2-1. In the setup a flywheel is driven by an electric motor via a torque converter and two clutches. Only the first clutch is controlled, the second one is only used to vary the loads.

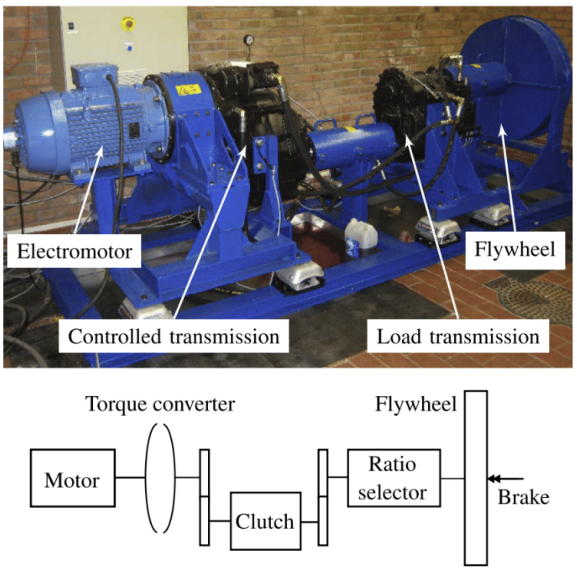


Figure 2-1: The experimental setup on which the model is based [2].

2-2 Inputs, variables and states

The input to the system is the voltage sent to the oil valve. The minimum and maximum control input are 0 and 0.2 volts respectively. An unsaturated control input $u_r(t)$ is saturated between 0 and 0.2 volts to obtain u: $u(t) = \min(0.2, \max(u_r(t), 0))$. Therefore $\mathcal{U} = [0, 0.2]$ V for this model. The state vector of the model is given as:

$$\xi_{\text{org}}(t) = \begin{bmatrix} x_1(t) & x_2'(t) & x_3(t) & x_4(t) & x_5(t) & x_6(t) \end{bmatrix}^T. \tag{2-1}$$

The physical meaning of each state can be found in Table 2-1.

State	Meaning	Unit
$x_1(t)$	Oil pressure	[Pa]
$x_2'(t)$	$\dot{x}_1(t)$	[Pa/s]
$x_3(t)$	Unsaturated piston position divided by k_t	[-]
$x_4(t)$	$\omega_i(t)$, angular velocity of the clutch input	[rad/s]
$x_5(t)$	$\omega_o(t)/g_{\text{rl}}$, angular velocity of the load	[rad/s]
$x_6(t)$	State to model transition between phases of torque transfer	[-]

Table 2-1: The physical meaning of the states in ξ_{org}.

The slip in terms of the states of the model is given as:

$$s(t) = x_4(t) - g_{\text{rl}} x_5(t). \tag{2-2}$$

2-3 Dynamics

The dynamics of the model are hybrid and the continuous dynamics in each region is nonlinear. In this Section we will go through the dynamics and highlight the hybrid elements of it. The values and meaning of the constants of the model can be found in Appendix A. The piston position $\text{pos}_1(t)$ is obtained by saturating $k_t x_3(t)$ between a lower bound of 0 and an upper bound of pos_{c}:

$$\text{pos}_1(t) = \min(\text{pos}_{\text{c}}, \max(0, k_t x_3(t))). \tag{2-3}$$

This saturation leads to three different cases for the value of $\text{pos}_1(t)$:

$$\text{pos}_1(t) = \begin{cases} 0 & \text{if} \quad k_t x_3(t) \leq 0 \\ k_t x_3(t) & \text{if} \quad 0 < k_t x_3(t) \leq \text{pos}_{\text{c}} \\ \text{pos}_{\text{c}} & \text{if} \quad k_t x_3(t) > \text{pos}_{\text{c}} \end{cases} \tag{2-4}$$

In addition to pos_1 a variable pos_2 is used in the state equations. When $x_6(t)$ becomes bigger or equal than 1, $\text{pos}_2(t)$ is set to the arbitrary value of 1.

$$\text{pos}_2(t) = \begin{cases} \text{pos}_1(t) & \text{if } x_6 < 1 \\ 1 & \text{if } x_6 \geq 1 \end{cases} \tag{2-5}$$

Let us introduce $\text{pos}_{\text{diff}}(t)$ and $\text{pos}_{\text{frac}}(t)$ which are used in the next equations.

$$\text{pos}_{\text{diff}}(t) = \text{pos}_2(t) - \text{pos}_{\text{c}} \tag{2-6}$$

$$\text{pos}_{\text{frac}}(t) = \frac{\frac{\text{pos}_1(t)}{\text{pos}_{\text{c}}} - \text{frac}_{\text{s}}}{1 - \text{frac}_{\text{s}}} \tag{2-7}$$

The clutch contact pressure $p(t)$, which is used in the dynamics of $x_4(t)$ and $x_5(t)$ is given as Equation 2-8, this introduces two different cases.

$$p(t) = \begin{cases} 10^5 & \text{if } \text{pos}_{\text{frac}}(t) \leq 0 \\ \text{pos}_{\text{frac}}(t) 10^5 x_1(t) + 10^5 & \text{if } \text{pos}_{\text{frac}}(t) > 0 \end{cases} \tag{2-8}$$

The first two differential equations model how the input voltage u influences the oil pressure x_1:

$$\dot{x}_1(t) = x_2'(t) \tag{2-9}$$

$$\dot{x}_2'(t) = -\frac{6}{s^2}x_1(t) - \frac{4}{s}x_2'(t) - \frac{6ak}{s^2} + \frac{6k}{s^2}u(t) - \frac{2k}{s}\dot{u}(t) \tag{2-10}$$

$$\tag{2-11}$$

In order to put the system in state space form, we need to get rid of the derivative of the input $\dot{u}(t)$ on the right hand side of Equation 2-10. If we bring the term containing $\dot{u}(t)$ to the left hand side, we obtain:

$$\dot{x}_2(t)' + \frac{2k}{s}\dot{u}(t) = -\frac{6}{s^2}x_1(t) - \frac{4}{s}x_2(t)' - \frac{6ak}{s^2} + \frac{6k}{s^2}u(t). \tag{2-12}$$

By introducing the variable $x_2(t) := x_2(t)' + \frac{2k}{s}u(t)$ and substituting its derivative into Equation 2-12, we obtain:

$$\dot{x}_2(t) = -\frac{6}{s^2}x_1(t) - \frac{4}{s}x_2(t)' - \frac{6ak}{s^2} + \frac{6k}{s^2}u(t). \tag{2-13}$$

We redefine the state vector as:

$$\xi(t) := \begin{bmatrix} x_1(t) & x_2(t) & x_3(t) & x_4(t) & x_5(t) & x_6(t) \end{bmatrix}^T. \tag{2-14}$$

By substituting $x_2(t)'$ by $x_2(t) - \frac{2k}{s}u(t)$, we finally get:

$$\dot{x}_1(t) = x_2(t) - \frac{2k}{s}u(t) \tag{2-15}$$

$$\dot{x}_2(t) = -\frac{6}{s^2}x_1(t) - \frac{4}{s}\left(x_2(t) - \frac{2k}{s}u(t)\right) - \frac{6ak}{s^2} + \frac{6k}{s^2}u(t). \tag{2-16}$$

The dynamics of $x_3(t)$ are given by:

$$\dot{x}_3(t) = \frac{b_2}{c_2}x_1(t) + \frac{a_2}{c_2}x_2(t)' - \frac{d_2}{c_2}x_3(t) - \frac{c_o b_2}{c_2}. \tag{2-17}$$

Substituting $x_2'(t)$ by $x_2(t) - \frac{2k}{s}u(t)$ again gives us:

$$\dot{x}_3(t) = \frac{b_2}{c_2}x_1(t) + \frac{a_2}{c_2}\left(x_2(t) - \frac{2k}{s}u(t)\right) - \frac{d_2}{c_2}x_3(t) - \frac{c_o b_2}{c_2}. \tag{2-18}$$

The dynamics of the input and output shaft speeds $x_4(t)$ and $x_5(t)$ are given below. The dynamics of $x_4(t)$ depend on whether the system is slipping or locking. Locking occurs when the slip is sufficiently small: $|s(t)| < 5 \cdot 10^{-4}$.

$$\dot{x}_4(t) = \begin{cases} -\frac{b_{1v}}{J_1}x_4(t) - \frac{T_{1c}}{J_1} - x_6(t)\frac{ap(t)}{J_1} + \frac{\omega_m^2 f_{\text{st}}\left(\frac{x_4(t)}{\omega_m}\right)}{J_1 f_{\text{sk}}\left(\frac{x_4(t)}{\omega_m}\right)^2} & \text{if} \quad |s(t)| \geq 5 \cdot 10^{-4} \\ -(1 - x_6(t))\frac{\gamma}{J_1(\text{pos}_{\text{frac}})}\left(x_5(t) - \frac{1}{g_{\text{rl}}}x_4(t)\right) & \\ & \\ -\frac{b_{1v}}{J_1+\frac{J_3}{g_{\text{rl}}^2}}x_4 - \frac{T_{1c}}{J_1+\frac{J_3}{g_{\text{rl}}^2}} + \frac{\omega_m^2 f_{\text{st}}\left(\frac{x_4}{\omega_m}\right)}{\left(J_1+\frac{J_3}{g_{\text{rl}}^2}\right) f_{\text{sk}}\left(\frac{x_4}{\omega_m}\right)^2} - & \text{if} \quad |s(t)| < 5 \cdot 10^{-4} \\ \frac{b_{3v}}{g_{\text{rl}}^2 J_1 + J_3}x_4 - \frac{T_{3c}}{(g_{\text{rl}}^2 J_1 + J_3)\omega_m}x_4 & \end{cases} \tag{2-19}$$

The dynamics of $x_5(t)$ depend on whether $x_5(t)$ is above or below a threshold w_{th}. It is modeled in this way to avoid the occurrence of non-physical motion caused by numerical errors. There is an additional case for the dynamics of x_5 when locking occurs.

$$\dot{x}_5(t) = \begin{cases} -\frac{b_{3v}}{J_3}x_5(t) - \frac{T_{3c}}{J_3} + x_6(t)\frac{g_{\text{rl}}\alpha p(t)}{J_3} & \text{if} \quad |x_5(t)| \geq w_{\text{th}} \\ +(1 - x_6(t))\frac{g_{\text{rl}}\gamma}{J_3(\text{pos}_{\text{frac}})}\left(x_5(t) - \frac{1}{g_{\text{rl}}}x_4(t)\right) & \wedge |s(t)| \geq 5 \cdot 10^{-4} \\ & \\ \frac{1}{w_{\text{th}}}\left(-\frac{b_{3v}w_{\text{th}}}{J_3} - \frac{T_{3c}}{J_3}\right)x_5(t) + x_6(t)\frac{g_{\text{rl}}\alpha p(t)}{J_3} & \text{if} \quad |x_5(t)| < w_{\text{th}} \\ +(1 - x_6(t))\frac{g_{\text{rl}}\gamma}{J_3(\text{pos}_{\text{frac}})}(x_5(t) - \frac{1}{g_{\text{rl}}}x_4(t)) & \wedge |s(t)| \geq 5 \cdot 10^{-4} \\ & \\ 1/g_{\text{rl}}\left(-\frac{b_{1v}}{J1+\frac{J3}{g_{\text{rl}}^2}}x_4(t) - \frac{T1c}{J1+\frac{J3}{g_{\text{rl}}^2}} + \frac{\omega_m^2 f_{\text{st}}(\frac{x_4(t)}{\omega_m})}{(J1+\frac{J3}{g_{\text{rl}}^2})f_{\text{sk}}(\frac{x_4(t)}{\omega_m})^2}\right) & \text{if} \quad |s(t)| < 5 \cdot 10^{-4} \\ -1/g_{\text{rl}}\left(\frac{b_{3v}}{g_{\text{rl}}^2 J1+J3}x_4 - \frac{T3c}{(g_{\text{rl}}^2 J1+J3)\omega_m}x_4(t)\right) & \end{cases} \tag{2-20}$$

The state variable $x_6(t)$ is used to govern the transition between the first and second phase of torque transfer in Equations 2-19 and 2-20. The hybrid dynamics of $x_6(t)$ are given by:

$$\dot{x}_6(t) = \begin{cases} 0 & \text{if} \quad \text{pos}(t) < \text{frac}_{\text{s}}\text{pos}_{\text{c}} \vee x_6(t) > 1 \\ \beta(x_6(t) + c_t) & \text{if} \quad \text{pos}(t) \geq \text{frac}_{\text{s}}\text{pos}_{\text{c}} \wedge x_6 \leq 1. \end{cases} \tag{2-21}$$

The functions $f_{\text{st}} : \mathbb{R} \mapsto \mathbb{R}$ and $f_{\text{sk}} : \mathbb{R} \mapsto \mathbb{R}$ are given in Equations 2-22 and 2-23 respectively.

$$f_{\text{st}}(x) = -0.546x^2 - 0.605x + 1.93 \tag{2-22}$$

$$f_{\text{sk}}(x) = \exp(5.72x - 0.377) + 11.8 \tag{2-23}$$

The torque sensor that is installed on the experimental setup, measures the torque $\tau(t) = T(\xi(t), \dot{\xi}(t))$ as:

$$T(\dot{\xi}(t), \xi(t)) = \begin{cases} J_3\dot{x}_5(t) + b_{3v}x_5(t) + \frac{T_{3cx_5}(t)}{w_{\text{th}}} & \text{if} \quad x_5(t) < w_{\text{th}} \\ J_3\dot{x}_5(t) + b_{3v}x_5(t) + T_{3c} & \text{if} \quad x_5(t) \geq w_{\text{th}} \end{cases}. \tag{2-24}$$

2-4 Simulating the system

Before the system can be simulated, the parameter $p_{\max}$, which is used to compute α, must be computed. It is equal to $p(t_{\text{switch}})$, the contact pressure at $t = t_{\text{switch}}$ and is computed using a simulation of the filling phase. Its value only depends on the controller used in the filling phase. Although its value will vary for different values of l, we only consider the value generated by simulating the system for the nominal values of l for simplicity.

In this thesis we consider a single scenario where the motor turns constantly at 125.6637 rad/s (1200 RPM) and we have a low inertia load which starts at a standstill. For this scenario, the initial state ξ_0 is equal to:

$$\xi_0 = \xi(0) = \begin{bmatrix} -1.472 \\ 2.2431 \cdot 10^{-7} + \frac{2k}{s}u(0) \\ -1.9306 \cdot 10^{3} \\ 120.1139 \\ 8.5069 \cdot 10^{-7} \\ 0 \end{bmatrix} \tag{2-25}$$

2-5 Computing performance criteria

The engagement time is computed from the simulated trajectory by finding the first time instance where $|x_4(t) - g_{\text{rl}}x_5(t)| < 0.05$ rad/s, so we have:

$$\epsilon = 0.05 \text{ rad/s}. \tag{2-26}$$

The jerk $j(t) = \dot{\tau}(t) = \dot{T}(\dot{\xi}(t), \xi(t))$ is a function of $\ddot{x}_5(t)$, since $\dot{x}_5(t)$ is not differentiable at points in time where $\begin{bmatrix} \xi(t) \\ u(t) \end{bmatrix}$ is on the boundary of a region Ω_i in general, $j(t)$ is not properly defined at all times. To overcome this issue, we redefine the time interval of $j_{\max}$:

$$\lambda = [0, t_{\text{eng}}] \setminus \{t : \begin{bmatrix} \xi(t) \\ u(t) \end{bmatrix} \in \partial\Omega_i\}, \tag{2-27}$$

wherein $\partial\Omega_i$ denotes the boundary of Ω_i. $j_{\max}$ is redefined as:

$$j_{\max} = \max_{t \in \lambda} |j(t)|. \tag{2-28}$$

2-6 Uncertainty model

The mismatch between this model and reality mainly comes from unmodelled dynamics. From experiments it is known that this uncertainty can be captured by taking the parameters frac_{s} and c_o as intervals around their nominal value. frac_{s} is the fraction of pos_{c} at which the clutch plate contact pressure $p(t)$ becomes equal to the oil pressure $x_1(t)$ and c_o is a constant which

is used in the dynamics of $x_3(t)$. Full uncertainty is considered by taking these variables as the following intervals respectively:

$$c_o \in \mathcal{C}, \mathcal{C} := [1.2c_o^{\text{nom}}, 0.8c_o^{\text{nom}}] \tag{2-29}$$

$$\text{frac}_\text{s} \in \mathcal{F}, \mathcal{F} := [0.8\text{frac}_\text{s}^{\text{nom}}, 1.2\text{frac}_\text{s}^{\text{nom}}]. \tag{2-30}$$

Therefore, for this model, we have:

$$l = \begin{bmatrix} c_o \\ \text{frac}_\text{s} \end{bmatrix}, \tag{2-31}$$

and when we consider uncertainty we have:

$$\mathcal{L} = \mathcal{C} \times \mathcal{F}. \tag{2-32}$$

Chapter 3

Genetic Programming

In this Chapter we give a basic introduction to Evolutionary Algorithms (EAs) in Section 3-1 and explain the elements of GP in Sections 3-2 to 3-6. In Section 3-8 we explain which steps must be taken before we can use GP to synthesize controllers for a wet clutch.

3-1 Evolutionary Algorithms

EAs are a collection of algorithms which try to solve problems by simulating natural evolution. This is usually done by generating a random population of solutions or individuals and by repeatedly forming new generations by applying genetic operators on good solutions from the previous generations. These genetic operators typically combine elements of multiple good individuals or apply random mutations on them to create new individuals. This process of creating generations is repeated until a stopping criterion is met [15]. The general EA procedure is summarized in Figure 3-1.

The most popular EA is Genetic Algorithm (GA), which was introduced by Holland in the 1970s [16]. GA encodes individuals as (usually fixed-length) character strings, much alike chromosomes in natural organisms. Genetic Programming was introduced by Koza in the 1990s [17] as an extension of GA. It was developed as a way to automatically generate computer programs. Instead of fixing the structure of the solution beforehand, individuals are encoded as trees of which both the nodes and the structure are allowed to evolve. This property is useful when the structure of the optimal solution is not known beforehand.

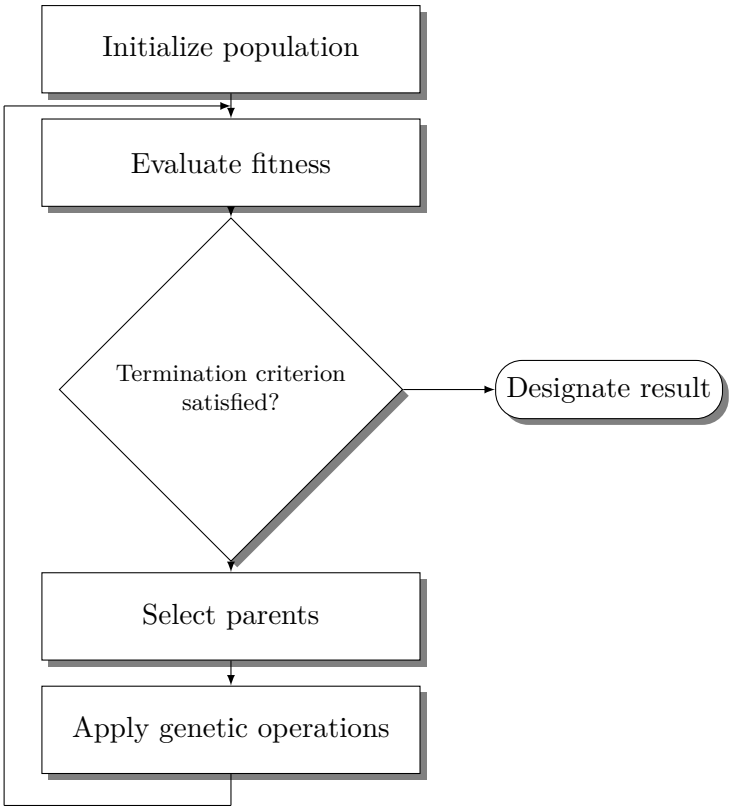


Figure 3-1: Schematic overview of the EA procedure.

3-2 Representation of individuals

In classical GP, expressions are represented by trees. In this way the structure of an expression can be altered by altering the structure of the tree. In GP, individuals consist of a combination of functions and terminals. Terminals can be either variables or constants and thus take no arguments. The functions can be any arithmetic, mathematical or logical function and take terminals or outputs from other functions as arguments. In Figure 3-2 the expression $(a^b + \sin(b)) \times (c + d)$ is represented as a tree. The set of functions in this tree is equal to $\{\times, \hat{}, +, \sin\}$, the set of terminals is equal to $\{a, b, c, d\}$. The leaves of the tree consist of terminals and the nodes of the tree consist of functions.

An individual that is generated from a set of terminals and a set of functions may not be syntactically valid. For example when one takes 0 as the denominator in a division, it may lead to an expression which is not mathematically defined. The problem of constructing a terminal set and a function set such that any generated individual is valid is called the problem of closure by Koza [17].

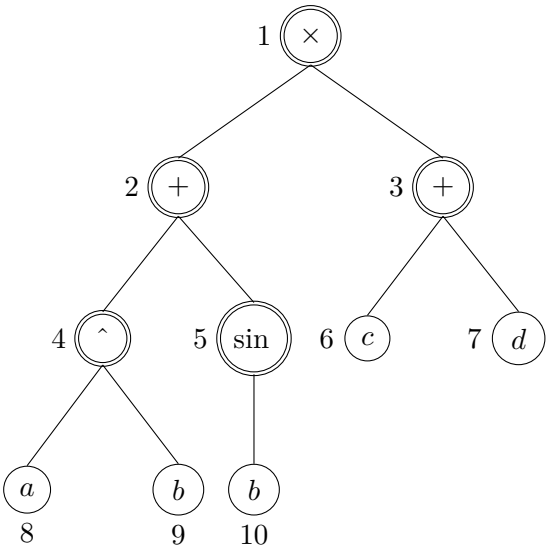


Figure 3-2: The expression $(a^b + \sin(b)) \times (c + d)$ represented as a tree, non-terminals are drawn as double circles and terminals are drawn as single circles.

3-2-1 Grammar-guided Genetic Programming

A popular way to restrict the search space of GP is by using grammar. A grammar is a set of rules that define which sentences in a language are syntactically correct. Such a mechanism can also be used to define which expressions are allowed in GP. In this way the problem of closure reduces to constructing a grammar that always lead to valid individuals. Additionally, restricting the search space can be used to incorporate expert knowledge of structures of good solutions. GP combined with a grammar is called Grammar-guided Genetic Programming (GGGP). The use of a Context-Free Grammar (CFG) in combination with GP was proposed by Wigham [18]. A CFG is defined by a four-tuple (N, Σ, P, S), where N is the non-terminal alphabet, Σ is the terminal alphabet, P is the set of production rules and S is the designated start symbol [18]. Production rules are rules that transform non-terminals into non-terminals or terminals. We denote a production rule that transforms a into either b or c by: a $\rightarrow$ b|c. The characteristic element of a CFG as opposed to other types of grammars is that the left hand side of the production rules contain only one non-terminal and therefore the context of the non-terminal in relation to other symbols surrounding it does not influence the derivation of that non-terminal. A CFG G_1 that is able to produce the expression from Figure 3-2 could be given as $G_1 = (N_1, \Sigma_1, P_1, S_1)$, with:

$$\begin{aligned} N_1 =& \{\text{Prod}, \text{Sum}, \text{Expr}, \text{Var}\} \\ \Sigma_1 =& \{a, b, c, d, \times, +, \hat{}, \sin, (,)\} \\ P_1 =& \{\text{Prod} \rightarrow \text{Sum} \times \text{Sum}, \text{Sum} \rightarrow (\text{Expr} + \text{Expr}), \text{Expr} \rightarrow \text{Var}\hat{}\text{Var} | \sin(\text{Var}) | \text{Var}, \\ & \text{Var} \rightarrow a|b|c|d\} \\ S_1 =& \{\text{Prod}\}. \end{aligned} \tag{3-1}$$

An individual is generated from a grammar by starting with a terminal or non-terminal from S. Then each of the non-terminals that remain in the current expression is transformed by

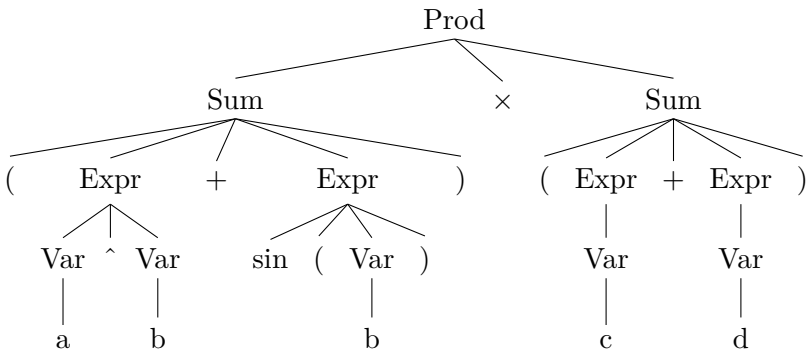


Figure 3-3: The expression $(a^b + \sin(b)) \times (c + d)$ represented as a derivation tree, derived from G_1.

selecting a random production rule from P. The derivation is completed when only terminals remain in the expression. Figure 3-3 shows the derivation tree for the expression using G_1.

3-3 Initialisation

The initial population is formed by deriving n_p random individuals from the grammar, where n_p is the fixed amount of individuals in a population. In order to exclude the possibility of creating infinitely big trees, a maximum tree depth is often desired. When the used grammar contains recursive production rules, i.e. when a non-terminal can eventually produce an expression containing the same non-terminal, a derivation tree can become infinitely large. Wigham [18] proposed to restrict the tree depth by labeling each production rule with minimum number of derivation steps to create only terminals. In [19], a maximum tree depth is ensured by separating the production rules into recursive and non-recursive rules. We can limit the depth of a generated tree by selecting only non-recursive rules when a certain tree depth is reached. Using this method we redefine a grammar to be (N, Σ, P, P^*, S), where P^* is the set of non-recursive production rules.

3-4 Fitness

In EAs, fitness is a measure of the optimality of a solution. The fitness function should be a function which assigns a high fitness to good solutions and a low fitness to bad solutions. After the initial population is formed, it is assessed by its fitness so that fit individuals have a higher chance of being selected for reproduction. For single-objective optimization the selection of a fitness function is usually trivial. For the multi-objective optimization the fitness function is usually more complex. The most popular technique for multi-objective optimization is to have a ranking based on Pareto dominance as a fitness measure. This ranking, in combination with niching techniques to encourage diversity of solutions, is mostly used to handle multi-objective optimization in GP. A popular multi-objective optimization algorithm is Non-dominated Sorting in Genetic Algorithm II (NSGA-II) [20]. NSGA-II makes use of a non-dominated sorting mechanism introduced by Goldberg in [21]. This sorting mechanism first computes the set of non-dominated individuals in the population and assigns

these solutions, which are in the first non-dominated front, rank 1. Then, the next non-dominated front is computed by repeating this computation, excluding the individuals of rank 1. The new non-dominated front is assigned rank 2. This process is repeated until all individuals are ranked. Then, to preserve diversity, a final ranking is computed by sorting the population first by the non-domination rank and as a tie-breaker a crowding distance is used. The crowding distance is a measure of area an individual represents in the objective function space. The higher the crowding distance, the greater the distance to the nearest neighbors in the population. An individual with equal non-domination rank is assigned a higher final rank if it has a higher crowding distance.

3-5 Selection

The working principle of GP is the selection of good solutions from the previous generation (parents) to form new solutions with better solutions (offspring). The two main techniques to select parents are roulette wheel selection and tournament selection. Roulette wheel selection assigns to each individual a probability of being selected as a parent that is proportional its fitness. This is only useful when the fitness is proportional to the optimality of the solution, which is not the case when we make use of a Pareto ranking, therefore we will use tournament selection instead.

3-5-1 Tournament selection

In tournament selection, parents are selected by picking n_t individuals randomly to participate in a tournament, where n_t is the tournament size. The winner of a tournament is the individual with the highest fitness, or in the case of multi-objective optimization it is the individual with (final) rank 1. The winner of a tournament is selected as a parent. The tournament size determines the selection pressure of the algorithm. The higher the selection the pressure, the more better individuals are favored. Increasing n_t increases the selection pressure. A high selection pressure improves the speed of convergence of the algorithm by exploiting the current best solutions, but it also increases the chance of converging prematurely because of a lack of exploitation [22].

3-5-2 Elitism

Elitism is the practice of preserving the best n_e individuals of the previous generation into the next generation, where n_e is called the elitism number. By using elitism it is guaranteed that the best solution of the next generation is at least as good as the best solution of the previous generation. Empirical results have shown that elitism is an important factor in the performance of Multi Objective Evolutionary Algorithms [23].

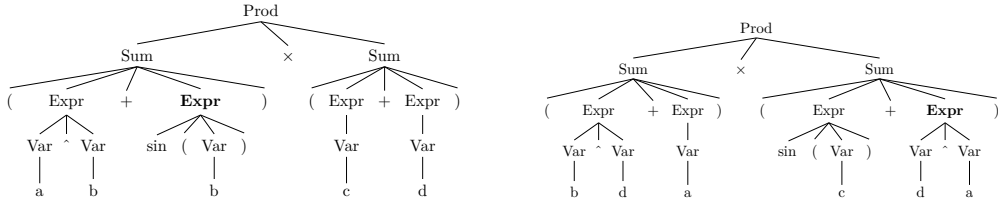
3-6 Genetic operations

When the parents are selected, genetic operations are applied to them in order to create offspring. The two most commonly used genetic operations in GP are crossover and mutation.

Both of these genetic operations find their inspiration in natural evolution.

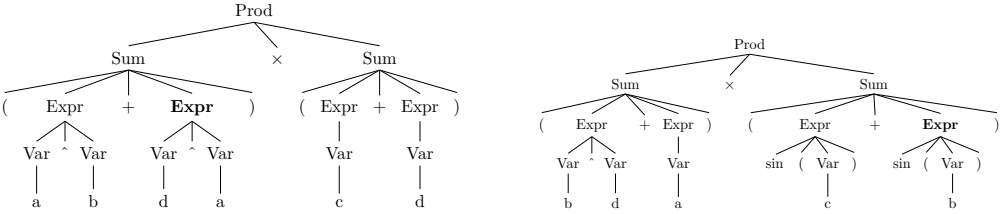
3-6-1 Crossover

The crossover operator creates two offspring from two parents by swapping subtrees of each parent. It is inspired by chromosomal crossover in sexual recombination. In GGGP, the non-terminal at the root of both subtrees to be swapped must be equal, so that any offspring is syntactically valid. In Figure 3-4 two individuals generated from the grammar G_1 that act as parents in this example are shown. In Figure 3-5 two offspring generated by applying crossover to these parents are shown, two Expr subtrees were swapped to create two new expressions.



(a) Parent 1: $(a^b + \sin(b))(c + d)$. **(b)** Parent 2: $(b^d + a)(\sin(c)d^a)$.

Figure 3-4: Two example parents generated from G_1.



(a) Offspring 1: $(a^b + d^a)(c + d)$. **(b)** Offspring 2: $(b^d + a)(\sin(c) + \sin(b))$.

Figure 3-5: Two example offspring generated by applying crossover to the parents in Figure 3-4. The Expr $\sin(b)$ of Parent 1 was swapped with the Expr d^a of Parent 2.

3-6-2 Mutation

The mutation operator takes one parent, selects a non-terminal to be mutated and randomly rederives it to create one offspring. The inspiration for this genetic operator comes from the random mutations that occur in natural genes. Figure 3-6 shows how mutation was used to generate an offspring from the expression from Figure 3-3.

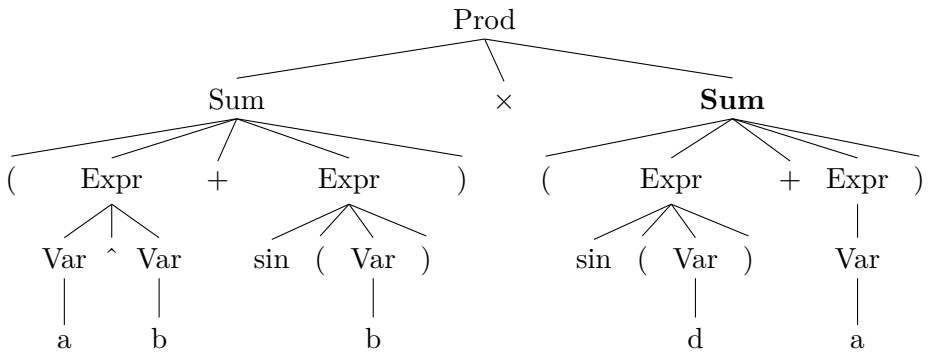


Figure 3-6: An example of an offspring generated by applying mutation to the expression in Figure 3-3. The Sum to the right (shown in bold) was rederived randomly, resulting in the expression $(a^b + \sin(b))(\sin(d) + a)$.

3-7 GP parameters

In GP several parameters need to be defined by the user, generally it is not obvious what the optimal values are for them.

3-7-1 Population size and amount of generations

The population size n_p is the most important parameter for controlling a GP run [24]. A big population size means that a great diversity of solutions can be maintained, so that a larger part of the Pareto optimal set of solutions can be found. The higher the amount of generations, the more the population is allowed to evolve. The amount of fitness evaluations is equal to the product of the population size and the amount of generations. For a given computation time of the algorithm a trade-off must be made between population size and amount of generations (assuming computation time of the fitness evaluation is dominant over the computation time of the other computations in the algorithm).

3-7-2 Genetic operators probabilities

Genetic operators are applied at a user defined rate. It is not entirely clear how to choose them in order to favor either exploration or exploitation [25]. Both crossover and mutation have an element of random exploration of the search space. Mutation is able to introduce expressions which are not present in the initial population. While crossover only combines existing solutions the selection of nodes is random and can create individuals which have a completely different fitness than its parents. Traditionally a crossover rate of >90% and a mutation rate of 0 was suggested [17], but currently a 50-50 mixture of crossover and mutation is considered to perform well [24].

3-8 Applying GP to wet clutch controller synthesis

Our approach is to use GP to synthesize pairs of feedforward and feedback controllers for a wet clutch. To be able to do this, we have to make several design choices.

3-8-1 Grammar

First of all we need to select a grammar from which controller can be generated. Since a feedforward controller takes different a input than feedback controller, different grammars are need for each controller type. The feedforward controller takes only time as an input and the feedback controller takes the states variables which are available for control: the oil pressure $x_1(t$ and the angular velocities of the input and output shafts $x_4(t)$ and $x_5(t)$.

3-8-2 Fitness evaluation and selection

In wet clutch control the objective is to minimize both $j_{\max}$ and t_{eng}, however the fitness function should be maximized to optimize performance. In order to have an infinite $j_{\max}$ and t_{eng} correspond to a fitness value of $\begin{bmatrix}0 & 0\end{bmatrix}$ and a $j_{\max}$ and t_{eng} of 0 correspond to $\begin{bmatrix}1 & 1\end{bmatrix}$, we choose the fitness function as in Equation 3-2.

$$\text{fitness}(u) = \begin{bmatrix}\frac{1}{1+t^u_{\text{eng}}} & \frac{1}{1+j^u_{\max}}\end{bmatrix} \tag{3-2}$$

Since we are interested in finding the Pareto optimal set of controllers, the obvious choice is to use non-dominated sorting and tournament selection. To evaluate the fitness function we use either simulations or reachability analysis. Since fitness evaluation takes a considerable amount of time compared to all other computations, the amount of fitness evaluations is the limiting factor of the speed the algorithm.

3-8-3 Termination criterion

The termination criterion is usually chosen so that the algorithm terminates when a certain fitness is achieved or when a maximum generation count $\text{gen}_{\max}$ is reached. For multi-objective optimization it is not straightforward to choose a fitness threshold and we will use only a $\text{gen}_{\max}$ termination criterion.

3-8-4 Designate result

After the termination criterion is reached the result of the algorithm must be designated. The goal of Problem 1 is to find the set of Pareto optimal controllers, therefore we designate the result of a run of GP as the non-dominated front of all generations combined.

Chapter 4

Evaluating worst case performance

In order to measure the robustness of the performance of a found controller, we want to be able to find an upper bound of the maximum jerk and engagement time associated to them. Since the model we consider has uncertain parameters which lie inside an infinite set, an infinite amount of simulations would be needed to find a formal guarantee on the lower bound on the performance. Instead we will use reachability analysis to achieve this.

4-1 Reachability analysis

Reachability analysis tries to compute the set states which are is reachable by a system in a certain time interval or at a certain point in time. Because every behaviour of the system is included in it, reachable sets can be used to formally verify properties of the system. Most research in the field of reachability analysis was motivated by the verification of safety properties [26], i.e. verifying whether a system avoids a set of unsafe states. Since computing the exact reachable set is undecidable for hybrid systems in general [27], most techniques for hybrid systems aim at the computation of over-approximations of the reachable set. Over-approximations of the reachable set can still be used to verify safety properties on systems. This principle also applies for the problem of finding a lower bound on the performance of a controller for a wet clutch. If we find an over-approximation of the reachable set for a clutch engagement and then compute the worst case performance for that over-approximation, the performance of all other possible trajectories is guaranteed to be at least as good as the worst case performance.

4-1-1 Reachable set

The reachable set $\mathcal{R}(t)$ of a system $\dot{\xi}(t) = f(\xi(t), u(t), l)$, at a certain point in time r can be defined similarly as in [28]:

$$\mathcal{R}(r) = \left\{ \xi(r) = \int_0^r f(\xi(t), u(t), l)dt \mid \xi(0) \in X_0, l \in \mathcal{L}, \quad \forall t \in [0, r] : u(t) \in \mathcal{U}, \right\} \tag{4-1}$$

Where X_0 is the initial set of states, $\mathcal{U}$ is the set of possible inputs and $\mathcal{L}$ is the set of possible parameters. The reachable set for a time interval $[0, r]$ is defined as the union of all reachable sets of all points in time of that interval [28]:

$$\mathcal{R}([0,r]) = \bigcup_{t\in[0,r]} \mathcal{R}(t) \tag{4-2}$$

4-1-2 Hybrid reachability

In order to compute $\mathcal{R}$ for a hybrid system, it is usually necessary to model the system as a hybrid automaton. We define a hybrid automaton as in [29]:

Definition 2. *A hybrid automaton* HA *is a 9-tuple* $(Z, z_0, X, X_0, \text{inv}, \text{Tr}, g, \text{jump}, f)$ *where*

- $Z \subseteq \mathbb{N}$ *is a finite set of locations with an initial location* $z_0 \in Z$,
- $X \subseteq \mathbb{R}^n, \xi(t) \in X$ *is the continuous state space,*
- $X_0 \subseteq X, \xi(0) \in X_0$ *is the initial continuous set,*
- inv *is the invariant function* $\text{inv} : Z \mapsto 2^X$ *that assigns an invariant* $\text{inv}(z) \subseteq X$ *to each location* $z \in Z$,
- Tr *is the set of discrete transitions* $\text{Tr} \subseteq Z \times Z$,
- g *is the guard function* $g : \text{Tr} \mapsto 2^X$ *that assigns a guard set* $g(\text{tr}) \subseteq X$ *to each transition* $\text{tr} = (z_1, z_2) \in \text{Tr}$,
- j *is the jump function* $j : \text{Tr} \times X \mapsto 2^X$ *that assigns a jump set* $j(\text{tr}, \xi) \subseteq X$ *to each pair* $\text{tr} \in \text{Tr}$ *and* $\xi \in g(\text{tr})$,
- $f : Z \mapsto (X \mapsto \mathbb{R}^n)$ *is the flow function that assigns a continuous vector field* $f(z)$ *to each location* $z \in Z$. *The continuous evolution in* z *is determined by the ODE* $\dot{\xi} = f(z, \xi(t))$.

The evolution of a hybrid automaton can be described as follows [30]: a hybrid automation starts at $t = 0$ at the location $z(0) = z_0$ and an initial state $\xi(0) \in X_0$, the evolution of the continuous state is governed by the flow function assigned to the current location. A transition can be taken when the continuous state is within the corresponding guard set, it has to be taken if the state would otherwise leave the invariant. The system state is updated according to the jump function when a transition is taken from one location to the next. In [29] the reachable set of a hybrid automaton is given as:

Definition 3. *Let* $S = \bigcup_{z\in Z} \bigcup_{x\in \text{inv}(z)(z,x)}$ *denote the set of hybrid states* (z, x) *of a hybrid automaton HA. Then, each possible run of HA is a sequence* $\sigma = \{s_0, s_1, \dots\}$, *if and only if:*

- *the initial hybrid state is* $s_0 = (z_0, \xi_0)$ *with* $\xi_0 \in X_0$,
- *each pair of consecutive states* $(s_i, s_{i+1}) \in \sigma$ *with* $s_i = (z_i, \xi_i)$ *and* $s_{i+1} = (z_{i+1}, \xi_{i+1})$ *satisfies:*

- *either (discrete transition)* $(z_i, z_{i+1}) \in \mathrm{Tr}, x_i \in g((z_i, z_{i+1}))$ *and* $\xi_{i+1} \in j((z_i, z_{i+1}), x_i)$*;*
- *or (continuous evolution)* $z_i = z_{i+1}$ *and there exists* $\chi : [0, \tau] \mapsto X, \tau \in \mathbb{R}^{>0}$ *such that* $\xi_i = \chi(0), \dot{\chi}(t) = f(z_i, \chi(t)), \chi(t) \in inv(z_i)$ *for* $t \in [0, \tau]$ *and* $\xi_{i+1} = \chi(\tau)$*.*

If Σ *is the set of all possible runs of HA, the reachable set is defined by* $R = \{s | \exists \sigma \in \Sigma : s \in \sigma\} \subseteq S$*, i.e. R contains all hybrid states that are elements of at least one run* σ*.*

If we are able to find an over-approximation of R for the wet clutch model, an input and an uncertain parameters set, we have an over-approximation of all behaviours the system and we can compute the worst case performance of the system.

4-2 Wet clutch model as a hybrid automaton

First, we have to express the model as introduced in Chapter 2 as a hybrid automaton. From analyzing simulations of the system, 7 modes which were always visited in the same order can be identified. For completeness, all locations and transitions that could theoretically be visited should be model, but for simplicity we only implement the locations and the transition order which was found for these simulations. We name a mode for which $z = i$, mode_i and we have $Z = \{1, 2, \ldots, 7\}$.

The system starts in mode_1 (for which $z = 1$). In this mode the piston has not start moving yet and $x_3(t) < 0$, meaning that the piston position $\mathrm{pos}_1(t) = 0$. Therefore the invariant related to this mode is active when $x_3 < 0$. To determine the flow at this mode, we need to look which case for each of the parts of the dynamics is true at this point. Since $x_6(t) < 1$, $\mathrm{pos}_{\mathrm{frac}}(t)(t) \leq 0$, $|x_4(t) - g_{\mathrm{rl}} x_5(t)| \geq 5 \cdot 10^{-4}$, $|x_5(t)| < w_{\mathrm{th}}$ and $\mathrm{pos}_1(t) < \mathrm{frac}_{\mathrm{s}}\mathrm{pos}_{\mathrm{c}} \vee x_6(t) > 1$, the flow of mode_1 is given as in Equations 4-3 and 4-4.

$$\dot{\xi}(t) = f_1(\xi(t)) = \begin{cases} \dot{x}_1(t) & = x_2(t) - \frac{2k}{s} u_{\mathrm{ff}}(t) \\ \dot{x}_2(t) & = -\frac{6}{s^2} x_1(t) - \frac{4}{s}(x_2(t) - \frac{2k}{s} u_{\mathrm{ff}}(t)) - \frac{6ak}{s^2} + \frac{6k}{s^2} u_{\mathrm{ff}}(t) \\ \dot{x}_3(t) & = \frac{b_2}{c_2} x_1(t) + \frac{a_2}{c_2}(x_2(t) - \frac{2k}{s} u_{\mathrm{ff}}(t)) - \frac{d_2}{c_2} x_3(t) - \frac{c_o b2}{c_2} \\ \dot{x}_4(t) & = -\frac{b_{1v}}{J_1} x_4(t) - \frac{T_{1c}}{J_1} - \frac{\gamma}{J_1 \mathrm{pos}_{\mathrm{diff}}(t)} \left(x_5(t) - \frac{1}{g_{\mathrm{rl}}} x_4(t) \right) + \frac{\omega_m^2 f_{st}(\frac{x_4(t)}{\omega_m})}{J_1 f_{sk}(\frac{x_4(t)}{\omega_m})} \\ \dot{x}_5(t) & = \frac{1}{w_{\mathrm{th}}} \left(-\frac{b_{3v} w_{\mathrm{th}}}{J_3} - \frac{T_{3c}}{J_3} \right) x_5(t) + \frac{g_{\mathrm{rl}} \gamma}{J_3 \mathrm{pos}_{\mathrm{diff}}(t)} \left(x_5(t) - \frac{1}{g_{\mathrm{rl}}} x_4(t) \right) \\ \dot{x}_6(t) & = 0 \end{cases} \tag{4-3}$$

$$\begin{aligned} p(t) &= 10^5 \\ \mathrm{pos}_1(t) &= 0 \\ \mathrm{pos}_2(t) &= \mathrm{pos}_1 \end{aligned} \tag{4-4}$$

The system transitions to mode_2 when $x_3(t) > 0$, so the guard of transition 1: becomes active when $x_3 > 0$. When this transition is taken, the piston starts moving and we have $\mathrm{pos}_1 = k_t x_3(t)$. The flow of mode2 is given by Equations 4-5 and 4-6.

$$\dot{\xi}(t) = f_2(\xi(t)) = f_1(\xi(t)) \tag{4-5}$$

$$\begin{aligned} p(t) &= 10^5 \\ \text{pos}_1(t) &= k_t x_3(t) \\ \text{pos}_2(t) &= \text{pos}_1(t) \end{aligned} \tag{4-6}$$

The system transitions to mode$_3$ when the guard of transition 2: $k_t x_3(t) \geq \frac{\text{pos}_c}{\text{frac}_s}$ becomes active, therefore we choose the invariant of mode$_2$ as $k_t x_3(t) < \frac{\text{pos}_c}{\text{frac}_s}$. When this transition is taken, the contact pressure $p(t)$ starts to depend on $\text{pos}_1(t)$ and $x_1(t)$, also $x_6(t)$ starts to increase. The flow of mode$_3$ is given by Equations 4-7 and 4-8.

$$\dot{\xi}(t) = f_3(\xi(t)) = \begin{cases} \dot{x}_1(t) & = x_2(t) - \frac{2k}{s} u_{\text{ff}}(t) \\ \dot{x}_2(t) & = -\frac{6}{s^2} x_1(t) - \frac{4}{s}(x_2(t) - \frac{2k}{s} u_{\text{ff}}(t)) - \frac{6ak}{s^2} + \frac{6k}{s^2} u_{\text{ff}}(t) \\ \dot{x}_3(t) & = \frac{b_2}{c_2} x_1(t) + \frac{a_2}{c_2}(x_2(t) - \frac{2k}{s} u_{\text{ff}}(t)) - \frac{d_2}{c_2} x_3(t) - \frac{c_o b2}{c_2} \\ \dot{x}_4(t) & = -\frac{b_{1v}}{J_1} x_4(t) - \frac{T_{1c}}{J_1} - (1 - x_6(t)) \frac{\gamma}{J_1 \text{pos}_{\text{diff}}(t)} \left(x_5(t) - \frac{1}{g_{\text{rl}}} x_4(t) \right) \\ & \quad -x_6(t) \frac{\alpha p(t)}{J_1} + \frac{\omega_m^2 f_{st}(\frac{x_4(t)}{\omega_m})}{J_1 f_{sk}(\frac{x_4(t)}{\omega_m})} \\ \dot{x}_5(t) & = \frac{1}{w_{\text{th}}} \left(-\frac{b_{3v} w_{\text{th}}}{J_3} - \frac{T_{3c}}{J_3} \right) x_5(t) + (1 - x_6(t)) \frac{g_{\text{rl}} \gamma}{J_3 \text{pos}_{\text{diff}}(t)} \left(x_5(t) - \frac{1}{g_{\text{rl}}} x_4(t) \right) \\ & \quad + x_6(t) \frac{g_{\text{rl}} \alpha p(t)}{J_3} \\ \dot{x}_6(t) & = \beta(x_6(t) + c_t) \end{cases} \tag{4-7}$$

$$\begin{aligned} p(t) &= \text{pos}_{\text{frac}}(t) 10^5 x_1(t) + 10^5 \\ \text{pos}_1(t) &= k_t x_3(t) \\ \text{pos}_2(t) &= \text{pos}_1(t) \end{aligned} \tag{4-8}$$

The system transitions to mode$_4$ when the guard of transition 3: $x_5(t) \geq w_{\text{th}}$ becomes active, the invariant of mode$_3$ is active when $x_5(t) < w_{\text{th}}$. When this transition is taken the dynamics of x_5 change because the speed of the output axis exceeds the threshold w_{th}. The flow of mode$_4$ is given by Equations 4-9 and 4-10.

$$\dot{\xi}(t) = f_4(\xi(t)) = \begin{cases} \dot{x}_1(t) & = x_2(t) - \frac{2k}{s} u_{\text{ff}}(t) \\ \dot{x}_2(t) & = -\frac{6}{s^2} x_1(t) - \frac{4}{s}(x_2(t) - \frac{2k}{s} u_{\text{ff}}(t)) - \frac{6ak}{s^2} + \frac{6k}{s^2} u_{\text{ff}}(t) \\ \dot{x}_3(t) & = \frac{b_2}{c_2} x_1(t) + \frac{a_2}{c_2}(x_2(t) - \frac{2k}{s} u_{\text{ff}}(t)) - \frac{d_2}{c_2} x_3(t) - \frac{c_o b2}{c_2} \\ \dot{x}_4(t) & = -\frac{b_{1v}}{J_1} x_4(t) - \frac{T_{1c}}{J_1} - (1 - x_6(t)) \frac{\gamma}{J_1 \text{pos}_{\text{diff}}(t)} \left(x_5(t) - \frac{1}{g_{\text{rl}}} x_4(t) \right) \\ & \quad -x_6(t) \frac{\alpha p(t)}{J_1} + \frac{\omega_m^2 f_{st}(\frac{x_4(t)}{\omega_m})}{J_1 f_{sk}(\frac{x_4(t)}{\omega_m})} \\ \dot{x}_5(t) & = -\frac{b_{3v}}{J_3} x_5(t) - \frac{T_{3c}}{J_3} + (1 - x_6(t)) \frac{g_{\text{rl}} \gamma}{J_3 \text{pos}_{\text{diff}}(t)} \left(x_5(t) - \frac{1}{g_{\text{rl}}} x_4(t) \right) \\ & \quad + x_6(t) \frac{g_{\text{rl}} \alpha p(t)}{J_3} \\ \dot{x}_6(t) & = \beta(x_6(t) + c_t) \end{cases} \tag{4-9}$$

$$\begin{aligned} p(t) &= \text{pos}_{\text{frac}}(t) 10^5 x_1(t) + 10^5 \\ \text{pos}_1(t) &= k_t x_3(t) \\ \text{pos}_2(t) &= \text{pos}_1(t) \end{aligned} \tag{4-10}$$

The system transitions to mode$_5$ when the guard of transition 4: $x_6(t) \geq 1$ becomes active, the invariant of mode$_4$ is active when $x_6(t) < 1$. This transition is taken when $x_6(t)$ reaches 1 and the transition from the first phase to the second phase of torque transfer is completed. The flow of mode$_5$ is given by Equations 4-11 and 4-12.

$$\dot{\xi}(t) = f_5(\xi(t)) = \begin{cases} \dot{x}_1(t) & = x_2(t) - \frac{2k}{s} u_{\text{ff}}(t) \\ \dot{x}_2(t) & = -\frac{6}{s^2} x_1(t) - \frac{4}{s}(x_2(t) - \frac{2k}{s} u_{\text{ff}}(t)) - \frac{6ak}{s^2} + \frac{6k}{s^2} u_{\text{ff}}(t) \\ \dot{x}_3(t) & = \frac{b_2}{c_2} x_1(t) + \frac{a_2}{c_2}(x_2(t) - \frac{2k}{s} u_{\text{ff}}(t)) - \frac{d_2}{c_2} x_3(t) - \frac{c_o b2}{c_2} \\ \dot{x}_4(t) & = -\frac{b_{1v}}{J_1} x_4(t) - \frac{T_{1c}}{J_1} - \frac{\alpha p(t)}{J_1} + \frac{\omega_m^2 f_{st}\left(\frac{x_4(t)}{\omega_m}\right)}{J_1 f_{sk}\left(\frac{x_4(t)}{\omega_m}\right)^2} \\ \dot{x}_5(t) & = -\frac{b_{3v}}{J_3} x_5(t) - \frac{T_{3c}}{J_3} + \frac{g_{\text{rl}} \alpha p(t)}{J_3} \\ \dot{x}_6(t) & = 0 \end{cases} \tag{4-11}$$

$$\begin{aligned} p(t) &= \text{pos}_{\text{frac}}(t) 10^5 x_1(t) + 10^5 \\ \text{pos}_1(t) &= k_t x_3(t) \\ \text{pos}_2(t) &= 1 \end{aligned} \tag{4-12}$$

The system transitions to mode$_6$ when the guard of transition 5: $k_t x_3(t) \geq \text{pos}_{\text{c}}$ becomes active, the invariant of mode$_5$ is active when $k_t x_3(t) < \text{pos}_{\text{c}}$. From this point the piston is fully extended and we switch to feedback control. The flow of mode$_6$ is given by Equations 4-13 and 4-14.

$$\dot{\xi}(t) = f_6(\xi(t)) = \begin{cases} \dot{x}_1(t) & = x_2(t) - \frac{2k}{s} u_{\text{fb}}(\xi(t)) \\ \dot{x}_2(t) & = -\frac{6}{s^2} x_1(t) - \frac{4}{s}(x_2(t) - \frac{2k}{s} u_{\text{fb}}(\xi(t))) - \frac{6ak}{s^2} + \frac{6k}{s^2} u_{\text{fb}}(\xi(t)) \\ \dot{x}_3(t) & = \frac{b_2}{c_2} x_1(t) + \frac{a_2}{c_2}(x_2(t) - \frac{2k}{s} u_{\text{fb}}(\xi(t))) - \frac{d_2}{c_2} x_3(t) - \frac{c_o b2}{c_2} \\ \dot{x}_4(t) & = -\frac{b_{1v}}{J_1} x_4(t) - \frac{T_{1c}}{J_1} - \frac{\alpha p(t)}{J_1} + \frac{\omega_m^2 f_{st}\left(\frac{x_4(t)}{\omega_m}\right)}{J_1 f_{sk}\left(\frac{x_4(t)}{\omega_m}\right)^2} \\ \dot{x}_5(t) & = -\frac{b_{3v}}{J_3} x_5(t) - \frac{T_{3c}}{J_3} + \frac{g_{\text{rl}} \alpha p(t)}{J_3} \\ \dot{x}_6(t) & = 0 \end{cases} \tag{4-13}$$

$$\begin{aligned} p &= \text{pos}_{\text{frac}}(t) 10^5 x_1(t) + 10^5 \\ \text{pos}_1(t) &= \text{pos}_{\text{c}} \\ \text{pos}_2(t) &= 1 \end{aligned} \tag{4-14}$$

When the guard of transition 6: $x_5(t) - \frac{x_4(t)}{g_{\text{rl}}} >= 5 \cdot 10^{-4}$ becomes active, the system enters the final mode, mode$_7$. In the model, no transition to another mode can be taken because the behaviour of the system is not of interest for the performance criteria, therefore the invariant set for mode$_7$ is equal to the state space X. The flow of mode$_7$ is given by Equations 4-15

and 4-16.

$$\dot{\xi}(t) = f_7(\xi(t)) = \begin{cases} \dot{x}_1(t) & = x_2(t) - \frac{2k}{s} u_{\text{fb}}(\xi(t)) \\ \dot{x}_2(t) & = -\frac{6}{s^2} x_1(t) - \frac{4}{s}(x_2(t) - \frac{2k}{s} u_{\text{fb}}(\xi(t))) - \frac{6ak}{s^2} + \frac{6k}{s^2} u_{\text{fb}}(\xi(t)) \\ \dot{x}_3(t) & = \frac{b_2}{c_2} x_1 + \frac{a_2}{c_2}(x_2(t) - \frac{2k}{s} u_{\text{fb}}(\xi(t))) - \frac{d_2}{c_2} x_3(t) - \frac{c_o b2}{c_2} \\ \dot{x}_4(t) & = -\frac{b_{1v}}{J1 + \frac{J3}{g_{\text{rl}}^2}} x_4(t) - \frac{T1c}{J1 + \frac{J3}{g_{\text{rl}}^2}} + \frac{\omega_m^2 f_{st}(\frac{x_4(t)}{\omega_m})}{(J1 + \frac{J3}{g_{\text{rl}}^2}) f_{sk}(\frac{x_4(t)}{\omega_m})^2} - \frac{b_{3v}}{g_{\text{rl}}^2 J1 + J3} x_4(t) \\ & \quad - \frac{T3c}{(g_{\text{rl}}^2 J1 + J3)\omega_m} x_4(t) \\ \dot{x}_5(t) & = \frac{1}{g_{\text{rl}}} \left(-\frac{b_{1v}}{J1 + \frac{J3}{g_{\text{rl}}^2}} x_4(t) - \frac{T1c}{J1 + \frac{J3}{g_{\text{rl}}^2}} + \frac{\omega_m^2 f_{st}(\frac{x_4(t)}{\omega_m})}{(J1 + \frac{J3}{g_{\text{rl}}^2}) f_{sk}(\frac{x_4(t)}{\omega_m})^2} \right) \\ & \quad - \frac{1}{g_{\text{rl}}} \left(\frac{b_{3v}}{g_{\text{rl}}^2 J1 + J3} x_4(t) - \frac{T3c}{(g_{\text{rl}}^2 J1 + J3)\omega_m} x_4(t) \right) \\ \dot{x}_6(t) & = 0 \end{cases} \tag{4-15}$$

$$\begin{aligned} p(t) &= \text{pos}_{\text{frac}}(t) 10^5 x_1(t) + 10^5 \\ \text{pos}_1(t) &= \text{pos}_{\text{c}} \\ \text{pos}_2(t) &= 1 \end{aligned} \tag{4-16}$$

Since $\xi(t)$ does not jump during any transition, every reset map is the identity map. See Figure 4-1 for a schematic overview of this hybrid automaton.

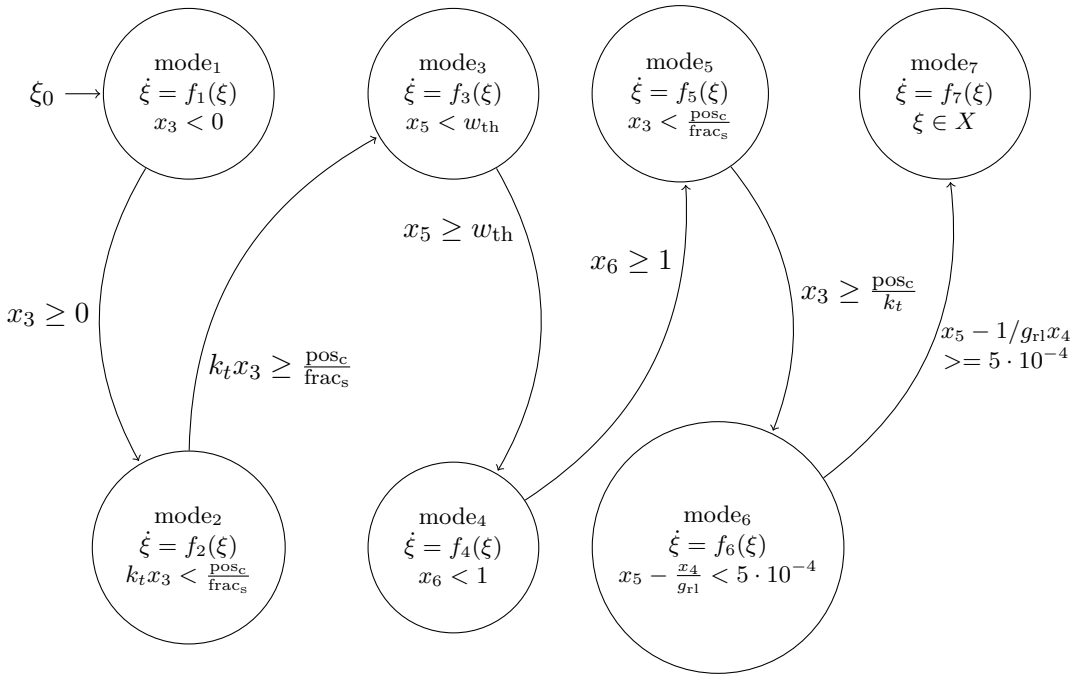


Figure 4-1: Schematic representation of the wet clutch model as a hybrid automaton. Transitions from one mode to another are drawn as arrows with the related guard written next to them. Each circle represents a node, the name of the mode, the dynamics of the mode and the invariant are listed inside them.

Several tools to compute R for this hybrid automaton exist. Most notably CORA [30], FLOW* [13] and ARIADNE[31]. We chose to use FLOW* because we found it to be the most user-friendly tool.

4-3 Flow* implementation

FLOW* can compute the reachable set for a specified time interval of system modeled by a hybrid automaton as a finite set of Taylor Models (TMs). The definition of a TM as given in [32] is:

Definition 4. *A Taylor Model (TM) is denoted by a pair (p, I) such that p is a polynomial over a set of variables x ranging in an interval domain D, and I is the interval remainder. Given a TM (p, I) and a function f which are over the same domain D, we say that f is over-approximated by (p, I) if $f(x) \in p(x) + I, \forall x \in D$.*

Because the uncertainties in our model are non-time-varying and FLOW* assumes all uncertain variables to be time-varying, we embed the uncertain variables into the hybrid automaton as states with derivatives equal to zero. We set the initial set of them equal to the interval their value may attain. Additionally a clock is needed to compute the feedforward control input. The augmented state vector $\xi_{\text{aug}} \in X_{\text{aug}} = \mathbb{R}^{n_{\text{aug}}}$ we use in FLOW* is equal to

$$\xi_{\text{aug}}(t) = \begin{bmatrix} \xi(t) \\ l \\ t \end{bmatrix} = \begin{bmatrix} \xi(t) \\ \text{frac}_{\text{s}} \\ c_o \\ t \end{bmatrix}, \tag{4-17}$$

the dynamics for the augmented state in mode i is equal to

$$\dot{\xi}_{\text{aug}}(t) = f_{\text{aug},i}(\xi_{\text{aug}}(t)) = \begin{bmatrix} f_i(\xi(t)) \\ 0 \\ 0 \\ 1 \end{bmatrix}. \tag{4-18}$$

FLOW* takes a model file as an input and outputs an output file containing the set of TMs and the computation path which indicates the time intervals at which transitions take place. The parameters in FLOW* that need to be specified by the user are the time interval $[0, \Delta]$ for which the reachable set is computed, the TM order, the time step size, an estimate for the remainder interval, a cutoff threshold and a precision. The cutoff threshold ε is used to simplify TMs: each polynomial that is within $[-\varepsilon, \varepsilon]$ is stored in the remainder interval. The remainder estimate φ is used as an initial estimate for the remainder interval in a TM computation. In our experience, a choice of a remainder estimate of 10^{-5}, a cutoff threshold of 10^{-6}, is the best for this specific problem. For these settings we chose an adaptive time step size of $[10^{-6}, 0.01]$ seconds and a TM order of 4. For our model an adaptive step size is useful because for some parts of the reachable set computation, a very small step size was needed, while for other parts a significantly bigger step size could be used.

4-4 Extracting t_{eng} from the output file

Included in the output file from FLOW∗ is a computation path representing the time intervals at which each transition is taken. A transition from mode$x$ to mode$y$ is outputted as

```
1 modex ( ni , [timin , timax] ) -> modey,
```

with ni denoting the ith transition taken and timin and timax denoting the lower and upper bound of the time, measured from the previous transition, at which transition i is taken respectively. Since the engagement time is equal to the first time instance where $|x_4(t) - g_{\text{rl}}x_5(t)| < \epsilon$, we can extract the worst case engagement time from the computation path by splitting mode$_6$ into two modes with equal dynamics: mode$_{6,1}$ and mode$_{6,2}$. By replacing mode$_7$ (which is not needed to compute the worst case performance) with mode$_{6,2}$ and picking the appropriate invariants and guards for mode as shown in Figure 4-2, the engagement time is equal to the time at which mode$_{6,2}$ is reached.

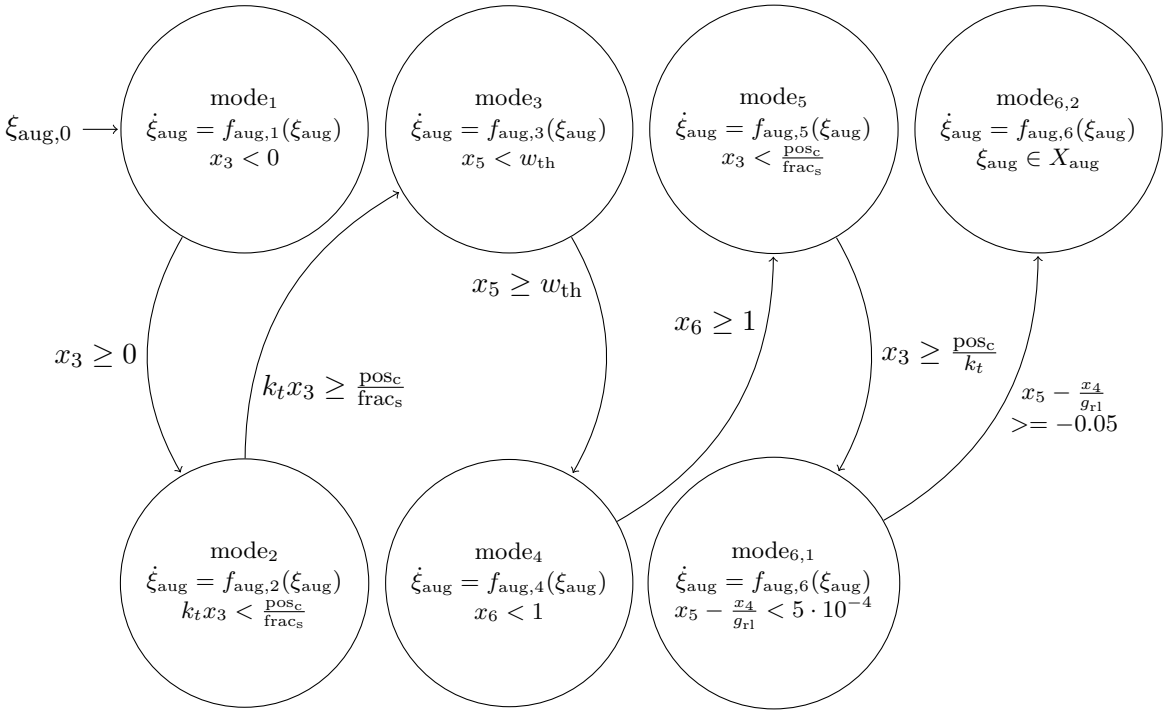


Figure 4-2: Modified hybrid automaton.

Since, each time interval in the computation path is the time that has passed since the last transition was taken, the worst case engagement time $t_{\text{eng}}^{u,\mathcal{L}}$ is obtained by taking the sum of the maximum value of the time for each transition:

$$t_{\text{eng}}^{u,\mathcal{L}} = \Sigma_{i\in\{1,\dots 6\}} \text{t}i\text{max} \tag{4-19}$$

These values are outputted in a FLOW∗ computation path as shown in Listing 4.1.

Listing 4.1: A FLOW* computation path.

```
1 mode1 ( n1 , [t1min , t1max] ) -> mode2 ( n3 , [t2min , t2max] ) ->
2 mode3 ( n3 , [t3min , t3max] ) -> mode4 ( n4 , [t4min , t4max] ) ->
3 mode5 ( n5 , [t5min , t5max] ) -> mode61 ( n6 , [t6min , t6max] ) ->
    mode62;
```

4-5 Computing $j_{\max}$ from the reachable set

As opposed to the worst case engagement time, the worst case jerk cannot be directly extracted from the output file and we cannot simply numerically differentiate TMs as we did when computing the jerk in a MATLAB simulation. Instead we will compute the worst case jerk by finding a direct mapping from the states to the jerk. The torque is given as in Equation 2-24 and since $j(t) = \dot{\tau}(t) = \dot{T}(\dot{\xi}(t), \xi(t))$, we have

$$\begin{aligned} j(t) &= \dot{T}(\dot{\xi}(t), \xi(t)) \\ &= \begin{cases} J_3\ddot{x}_5(t) + b_{3v}\dot{x}_5(t) + \frac{T_{3c\dot{x}_5(t)}}{w_{\text{th}}} & \text{if} \quad x_5(t) < w_{\text{th}} \\ J_3\ddot{x}_5(t) + b_{3v}\dot{x}_5(t) & \text{if} \quad x_5(t) > w_{\text{th}}. \end{cases} \end{aligned} \tag{4-20}$$

Since $x_5 < w_{\text{th}}$ for mode$_1$, mode$_2$ and mode$_3$ and $x_5 > w_{\text{th}}$ for the other modes we have

$$T_1(\dot{\xi}(t), \xi(t)) = T_2(\dot{\xi}(t), \xi(t)) = T_3(\dot{\xi}(t), \xi(t)) = J_3\ddot{x}_5(t) + b_{3v}\dot{x}_5(t) + \frac{T_{3c\dot{x}_5(t)}}{w_{\text{th}}} \tag{4-21}$$

$$T_4(\dot{\xi}(t), \xi(t)) = T_5(\dot{\xi}(t), \xi(t)) = T_6(\dot{\xi}(t), \xi(t)) = T_7(\dot{\xi}(t), \xi(t)) = J_3\ddot{x}_5(t) + b_{3v}\dot{x}_5(t), \tag{4-22}$$

where $T_i(\dot{\xi}(t), \xi(t))$ is the torque function for mode i. We can then find the jerk map $j_{z,\xi_{\text{aug}}}(z, \xi_{\text{aug}})$ per mode: $j_{z,\xi_{\text{aug}}} : Z \times X_{\text{aug}} \mapsto \mathbb{R}$, as

$$\begin{aligned} j_{i,\xi_{\text{aug}}}(i, \xi_{\text{aug}}(t)) &= \frac{dT_i(\dot{\xi}_{\text{aug}}(t), \xi_{\text{aug}}(t))}{dt} \\ &= \frac{dT_i}{d\xi_{\text{aug}}}\frac{d\xi_{\text{aug}}}{dt} + \frac{dT_i}{d\dot{\xi}_{\text{aug}}}\frac{d\dot{\xi}_{\text{aug}}}{dt} \\ &= \frac{dT_i}{d\xi_{\text{aug}}} f_{\text{aug},i}(\xi_{\text{aug}}(t)) + \frac{dT_i}{d\dot{\xi}_{\text{aug}}}\left(\frac{df_{\text{aug},i}(\xi_{\text{aug}})}{dt}\right) \\ &= \frac{dT_i}{d\xi_{\text{aug}}} f_{\text{aug},i}(\xi_{\text{aug}}(t)) + \frac{dT_i}{d\dot{\xi}_{\text{aug}}}\left(\frac{df_{\text{aug},i}(\xi_{\text{aug}})}{d\xi_{\text{aug}}}\frac{d\xi_{\text{aug}}}{dt}\right) \\ &= \frac{dT_i}{d\xi_{\text{aug}}} f_{\text{aug},i}(\xi_{\text{aug}}(t)) + \frac{dT_i(\dot{\xi}_{\text{aug}}, \xi_{\text{aug}})}{d\dot{\xi}_{\text{aug}}}\left(\frac{df_{\text{aug},i}(\xi_{\text{aug}})}{d\xi_{\text{aug}}} f_{\text{aug},i}(\xi_{\text{aug}}(t))\right). \end{aligned} \tag{4-23}$$

This function be computed using symbolic differentiation in MATHEMATICA, the resulting jerk maps can be found in Appendix B.

The over-approximation of the reachable set for an input u, which we will call $\hat{R}_u$, is outputted by FLOW* as a list of triples $(\hat{\zeta}_i, \nu_i, z_i)$ for $i \in \{1, \ldots, y\}$. Where $\hat{\zeta}_i$ is a TM over-approximating

$\xi_{\text{aug}}(t)$ for the time interval ν_i, $\forall t \in \nu_i : \xi_{\text{aug}}(t) \in \hat{\zeta}_i$. And z_i is the mode of the system in ν_i, $\forall t \in \nu_i : z = z_i$. The over-approximation of the reachable set $\hat{\mathcal{R}}_u \supseteq R_u$ is given as:

$$\hat{R}_u = \bigcup_{i \in \{1,\ldots,y\}} \hat{\zeta}_i.$$

Using interval arithmetic in MATHEMATICA, we first convert each $\hat{\zeta}_i$ to state intervals $\hat{\mu}_i \in \mathbb{I}$, such that $\forall t \in \nu_i : \xi_{\text{aug}}(t) \in \hat{\mu}_i \supseteq \hat{\zeta}_i$. By feeding each $\hat{\mu}_i$ in combination with the corresponding mode through $j_{z,\xi_{\text{aug}}}$, we obtain a jerk interval $\psi_i = j_{z,\xi_{\text{aug}}}(z_i, \hat{\mu}_i)$ such that $\forall t \in \nu_i : j(t) \in \psi_i$. In this case the jerk map $j_{z,\xi_{\text{aug}}}$ maps state intervals to jerk intervals: $j_{z,\xi_{\text{aug}}} : Z \times \mathbb{I}^{n_{\text{aug}}} \mapsto \mathbb{I}$.

$$\psi_i = \left\{ j_{z,\xi_{\text{aug}}}(z_i, x) | x \in \hat{\mu}_i \right\} \tag{4-24}$$

Using interval arithmetic we can find an over-approximation of this jerk interval: $\hat{\psi}_i \supseteq \psi_i$. See Figure 4-3 for a schematic overview of how we compute jerk intervals for an input pair $(u_{\text{ff}}, u_{\text{fb}})$. The over-approximation of the worst case jerk is then given by:

$$j_{\max}^{u,\mathcal{L}} = \max_i \max \hat{\psi}_i. \tag{4-25}$$

This result is an over-approximation of the actual worst case jerk because the reachable set computation as well as the conversion from TM to jerk interval using interval arithmetic is conservative in nature.

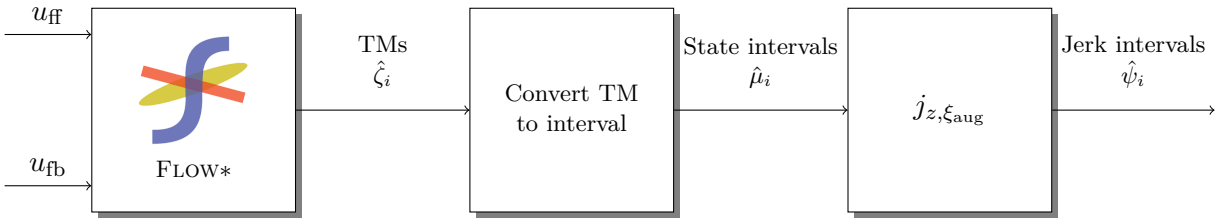


Figure 4-3: Schematic overview of how we compute jerk intervals for an input pair $(u_{\text{ff}}, u_{\text{fb}})$.

4-6 Example

As an example we will consider a simple controller $u(t)$, with $u_{\text{ff}}(t) = 0.1$ and $u_{\text{fb}}(\xi(t)) = 0.063$. We use this simple piece-wise linear controller, because more complex controllers gave issues in the computation of the reachable set. For this controller we will try to compute $t_{\text{eng}}^{u,\mathcal{L}}$ and $j_{\max}^{u,\mathcal{L}}$ for $\mathcal{L} = [1.2c_o^{\text{nom}}, 0.8c_o^{\text{nom}}] \times [0.985\text{frac}_{\text{s}}^{\text{nom}}, 1.015\text{frac}_{\text{s}}^{\text{nom}}]$. The found reachable set as well using the model file in Appendix C, as the trajectories for a simulation with the nominal parameters for each state can be seen in Figure 4-4. From the computation path we extract that $t_{\text{eng}}^{u,l} \in [0.9974, 1.1832]$ seconds, so we find $t_{\text{eng}}^{u,\mathcal{L}} = 1.1832$ seconds, the nominal engagement time for this controller is equal to 1.091 seconds. The worst case maximum jerk that was found is $j_{\max}^{u,\mathcal{L}} = 9.7 \cdot 10^5$, while for the nominal simulation $j_{\max} = 5.174 \cdot 10^3$.

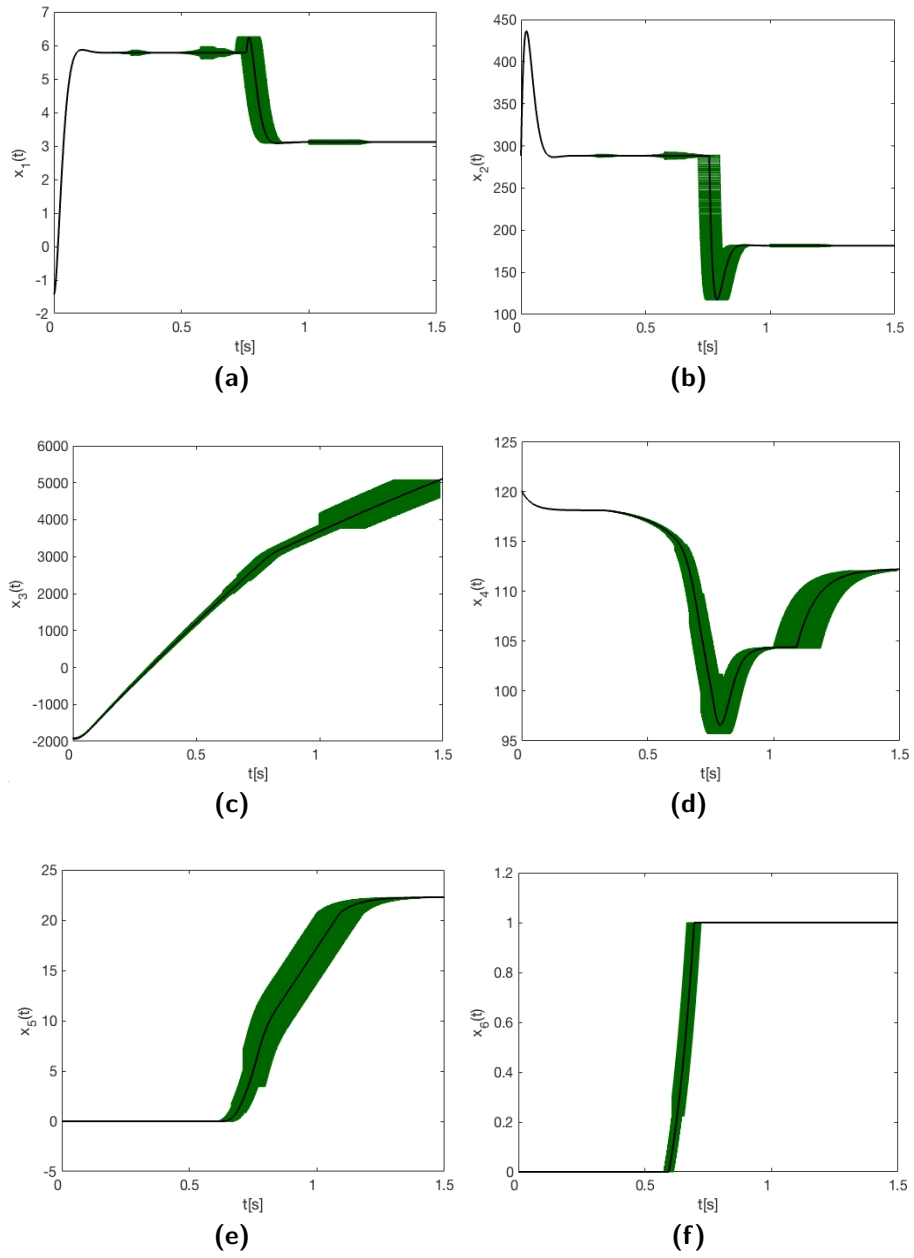


Figure 4-4: The reachable set found in FLOW∗ for each state is shown in green, the trajectory for a single simulation with nominal parameters is shown in black.

Chapter 5

Synthesizing feedback controllers without considering uncertainty

In this Chapter we consider Subproblem 1. The task here is to find a feedback controller, given a fixed feedforward controller for the filling phase. For the filling phase we take the baseline feedforward controller $u_{\text{ff,baseline}}(t)$ shown in in Figure 5-1. The performance $(t_{\text{eng,baseline}}, j_{\text{max,baseline}})$ which is achieved by using this controller for both the filling and the slipping phase is equal to $(1.036, 680.979)$. This feedforward controller is a hand-tuned controller which was designed for model validation experiments by FlandersMake. It serves as a baseline controller in this thesis. We chose to use this input because no other controllers were available to use for this model. The simulation results in this thesis cannot be compared with the experimental results in [2], because the torque signal was too noisy to directly differentiate and therefore a different, filtered, jerk measure was used in practice. We use the model from Section 2 with the nominal parameters, so there is no uncertainty in the model, hence we take $\mathcal{L}$ to be the singleton:

$$\mathcal{L} = \begin{bmatrix} c_o^{\text{nom}} \\ \text{frac}_{\text{s}}{}^{\text{nom}} \end{bmatrix}. \tag{5-1}$$

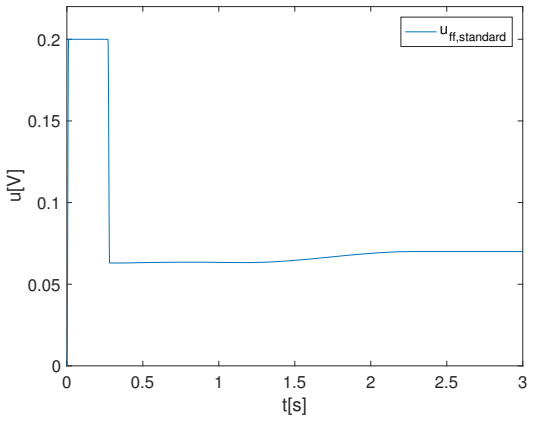


Figure 5-1: Baseline feedforward control input $u_{\text{ff,baseline}}(t)$.

5-1 Implementation

For the implementation of the GP algorithm we use a combination of MATLAB [33] and MATHEMATICA [34]. A MATLAB m-file to simulate the system was developed by Flanders-Make. By substituting a feedback controller into the differential equations, we can simulate the system for a given feedback controller. From the computed trajectories the performance related to the controller can be computed. We simulate the system for 3 seconds and when the clutch does not engage within this time or becomes unstable, an engagement time of 5 seconds and a maximum jerk of 10^7 rad/s^3 is assigned to the controller:

$$\text{fitness}(u) = \begin{cases} \left[\frac{1}{1+t_{\text{eng}^u}} \frac{1}{1+j_{\max^u}}\right] & \text{if } t_{\text{eng}^u} \leq 3 \text{ s} \\ \left[\frac{1}{1+5} \frac{1}{1+10^7}\right] & \text{if } t_{\text{eng}^u} > 3 \text{ s} \end{cases}. \tag{5-2}$$

GP as described in Chapter 3 was implemented in MATHEMATICA. We use MATLINK [35] to communicate the controllers from MATHEMATICA to MATLAB and to communicate the obtained t_{eng} and $j_{\max}$ from MATLAB to MATHEMATICA. See Figure 5-2 for an overview of the used architecture.

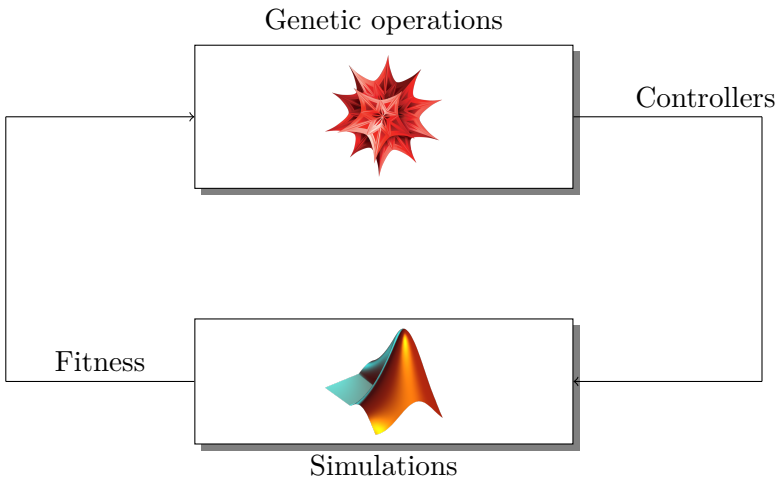


Figure 5-2: General architecture used for controller synthesis using simulations. Controllers are evolved by applying genetic operations to the previous generation in MATHEMATICA. The new controllers are used in MATLAB simulations to determine their performance. The performance is used in turn to generate a new generation of controllers in MATHEMATICA.

5-2 Parameters and grammar

We chose the parameters to generate the results of Section 5-3 as shown in Table 5-1.

Name	Value	Meaning
$\text{gen}_{\max}$	10	Amount of generations
n_p	40	Population size
d_{max}	7	Tree depth at which only recursive production rules are used
n_e	4	Elitism number
n_t	10	Tournament size
$\text{Pr}_{\text{crossover}}$	50%	Crossover probability
$\text{Pr}_{\text{mutation}}$	50 %	Mutation probability

Table 5-1: The meaning and value of the variables used in the experiment.

These parameters are not optimized in any sort of rigorous fashion and are by no means optimal. The chosen grammar $G_{\text{fb}} = (N_{\text{fb}}, \Sigma_{\text{fb}}, P_{\text{fb}}, P^*_{\text{fb}}, S_{\text{fb}})$ is given by:

$$N_{\text{fb}} = \{\text{S}, \text{Pol}, \text{Mon}, \text{Var}, \text{Constant}\} \tag{5-3}$$

$$\Sigma_{\text{fb}} = \{(,), \cdot, +, -, x_1, x_4, x_5, u_{\text{switch}}, x_{1,\text{switch}}, x_{4,\text{switch}}, x_{5,\text{switch}}, g_{\text{rl}}\} \tag{5-4}$$

$$P^*_{\text{fb}} = \{\text{Pol} \to \text{Constant} \cdot \text{Mon}, \text{Mon} \to \text{Var}, \text{Var} \to \{x_1 - x_{1,\text{switch}},$$
$$(x_4 - gr_{rl}x_5) - (x_{4,\text{switch}} - g_{\text{rl}}x_{5,\text{switch}}), x_4 - x_{4,\text{switch}}, x_5 - x_{5,\text{switch}}\}, \tag{5-5}$$
$$\text{Constant} \to\sim [-0.01, 0.01]\} \tag{5-6}$$

$$P_{\text{fb}} = \{P^*_{\text{fb}}, \text{Pol} \to (\text{Pol} + \text{Pol}), \text{Mon} \to \text{Mon} \cdot \text{Var}\} \tag{5-7}$$

$$S_{\text{fb}} = \{u_{\text{switch}} + \text{Pol}\}. \tag{5-8}$$

Here, $u_{\text{switch}}, x_{1,\text{switch}}, x_{4,\text{switch}}$ and $x_{5,\text{switch}}$ denote $u(t_{\text{switch}}), x_1(t_{\text{switch}}), x_4(t_{\text{switch}})$ and $x_5(t_{\text{switch}})$ respectively. By choosing the grammar in this way, it is guaranteed that $u(t)$ is continuous at $t = t_{\text{switch}}$, which is desirable because a discontinuity in the controller leads to spikes in the jerk. This grammar ensures that the feedback controller will be a polynomial of the available state variables and $s(t) = x_4(t) - g_{\text{rl}x_5(t)}$. $s(t)$ is included in the grammar because the goal is to drive this quantity to 0 in the slipping phase, therefore it is logical to incorporate this as expert knowledge in the grammar. A polynomial is a logical initial choice for the resulting function, because any analytic function can be approximated by a polynomial.

5-3 Results

The result of a run with these settings is shown in Figure 5-3. We observe that the non-dominated front that was found roughly intersects with the performance of u_{baseline}. For individuals that are close to u_{baseline} in terms of performance, $j_{\max}$ slowly increases for faster values of t_{eng}. But for individuals with a $t_{\text{eng}} < 0.85$ s, the maximum jerk deteriorates much more for an improvement in the engagement time. This could be explained by the fact that the performance is expected to be closer to the actual Pareto optimum around $t_{\text{eng,baseline}}$ and $j_{\max,\text{baseline}}$, because fixed feedforward control input was designed to perform well for an engagement time of around 1 second.

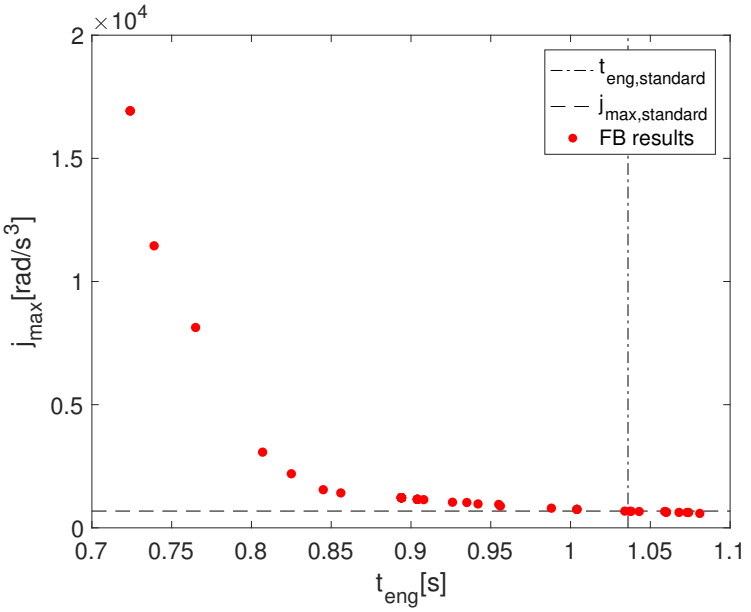


Figure 5-3: Results of the experiment. Each dot represents the performance of the final result. The standard performance $(\text{t}_{\text{eng,baseline}}, \text{j}_{\max,\text{baseline}})$ is shown as the dotted lines.

One found controller $u_{\text{fb},1}(\xi(t)) = 0.06338 - 0.00015763(x_4 - 105.8337)$ was Pareto dominant with respect to $\text{u}_{\text{baseline}}$, with a performance of $(1.034, 679.017)$. It slightly improved upon u_{baseline} in both control objectives. The control input of this controller is compared to the baseline input in Figure 5-4. It can be seen that the control input is very similar to u_{baseline} and managed to outperform it by applying a slightly higher control input. Another controller

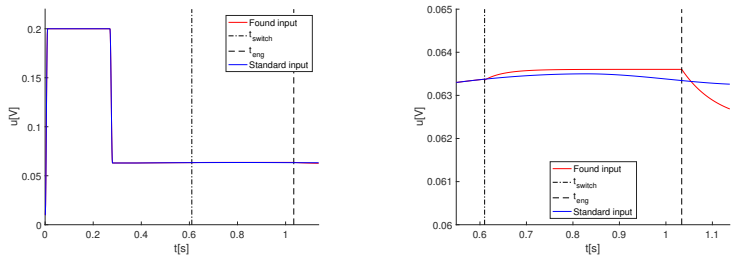


Figure 5-4: The input for $u_{\text{fb},1}(\xi) = 0.0633558 - 0.000157629(x_1 - 10.5.834)$ is plotted in red and the u_{baseline} is plotted in blue. The dotted lines indicate t_{switch} and t_{eng}. For this controller $j_{\max} =$ rad/s^3 and $t_{\text{eng}} =$s.

from the resulting non-dominated front is $u_{\text{fb},2}(\xi) = 0.0633558 - 0.000157629(x_1 - 10.5.834)$. The control input for this controller is compared in Figure 5-5. This controller smoothly increases from 0.063 to about 0.12 V to achieve a t_{eng} of 0.894 s and a $j_{\max}$ of 1221.68 rad/s^3, significantly improving the speed of the engagement, but also significantly increasing $j_{\max}$.

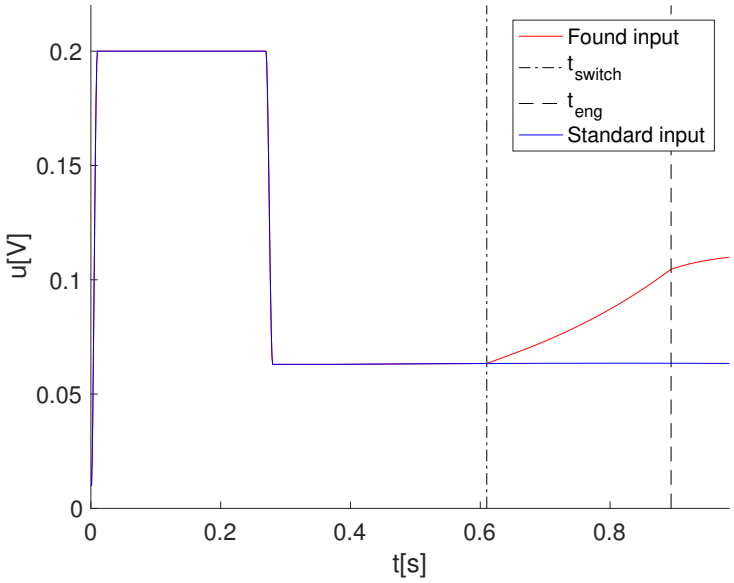


Figure 5-5: The input for $u_{\text{fb},2}(\xi) = 0.0633558 + 0.0026824(x_5 - 4.03066)$ is plotted in red and the u_{baseline} is plotted in blue. The dotted lines indicate t_{switch} and t_{eng}. For this controller $j_{\max} = 1221.68$ rad/s^3 and $t_{\text{eng}} = 0.894$s.

Chapter 6

Synthesizing feedforward and feedback controllers without considering uncertainty

In this Chapter we consider Subproblem 2 and add the task of synthesizing the feedforward controller ourselves to Subproblem 1.

6-1 Feedforward control grammar

Since we need to synthesize the full controller $u(t)$, we also need a grammar G_{ff} to synthesize feedforward controllers. From literature [2] it is known that a high initial input current is needed in the filling phase to ensure a good performance. The baseline controller in Figure 5-1 starts with the maximum input of 0.2 V for about 0.25 s, followed by a transition to an input value of about 0.063 V. Because the transition between input values should not be abrupt to avoid jerk spikes, we will use the logistic function H_s, which is a smooth approximation of the step signal:

$$H_s(t, t_0, k) = \frac{1}{1 + \exp(-2k(t - t_0))}, \tag{6-1}$$

wherein t_0 is the step time and k is a measure of steepness of the transition: $k = 2\dot{H}_s(t_0, t_0, k)$. The shape of the curve of this function for different values of k can be seen in Figure 6-1.

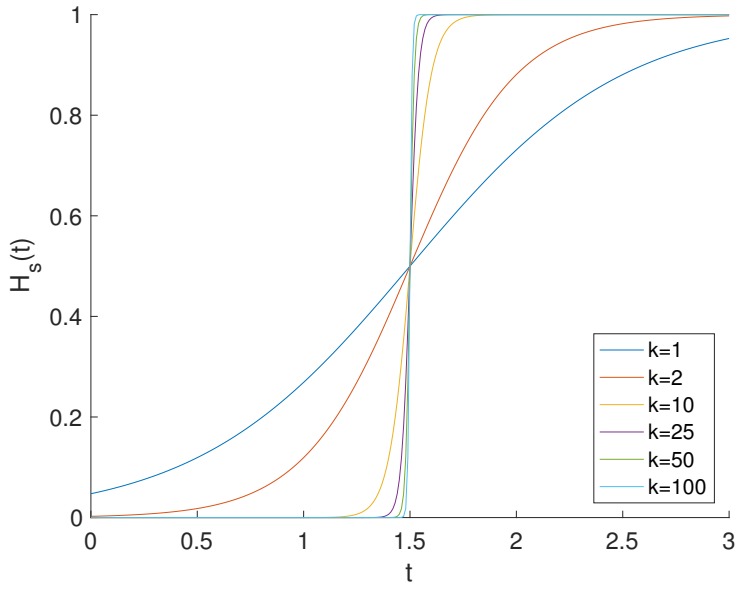


Figure 6-1: The logistic function $H_s(t)$ for $t_0 = 1.5$ and different values of k.

A simple grammar $G_{\text{ff}} = (N_{\text{ff}}, \Sigma_{\text{ff}}, P_{\text{ff}}, P^*_{\text{ff}})$ to synthesize feedforward controllers like the baseline controller could be chosen as:

$$N_{\text{ff}} = \{\text{S}_{\text{ff}}, \text{h}_1, \text{h}_2, \text{t}_1, \text{t}_2, \text{k}, \text{H}_\text{s}\} \tag{6-2}$$
$$\Sigma_{\text{ff}} = \{(,), +, -, \cdot, /, \exp, t\} \tag{6-3}$$
$$P_{\text{ff}} = \{\} \tag{6-4}$$
$$P^*_{\text{ff}} = \{\text{S}_{\text{ff}} \to h_1 H_s(t, t_1, k) - (h_1 - h_2) H_s(t, t_2, k), h_1 \to\sim [0.19, 0.2], \tag{6-5}$$
$$h_2 \to\sim [0.04, 0.09], t_1 \to 0, t_2 \to\sim [0.2, 0.3], k \to\sim [80, 600]\}.$$

This grammar ensures that the feedforward controller is given as

$$u_{\text{ff}}(t) = h_1 H_s(t, t_1, k) - (h_1 - h_2) H_s(t, t_2, k),$$

and is essentially a control input parametrized by $h_1, t_1, h_2, t_2$and k. The control signal starts with a transition from 0 to h_1 at $t = 0$, and goes down to a value of h_2 at $t = t_1$, the steepness of the transition is determined by $k \in [80, 600]$.

6-2 Simultaneous feedforward and feedback controller synthesis

A simple way to synthesize controllers consisting of a feedforward and a feedback part would be to synthesize both parts simultaneously. This way we allow the controllers in both phases to evolve at the same time, hoping that we find combinations of them which lead to good performance.

6-2-1 Results

The results of an experiment using the parameters in Table 5-1 and simultaneous feedforward and feedback controller synthesis are shown in Figure 6-2. It can be observed that much faster controllers were found than in Section 5-3. However the performance in the neighbourhood of the performance of u_{baseline} was worse and no controllers that were Pareto dominant to u_{baseline} were found. Therefore the results for this method do not seem satisfying.

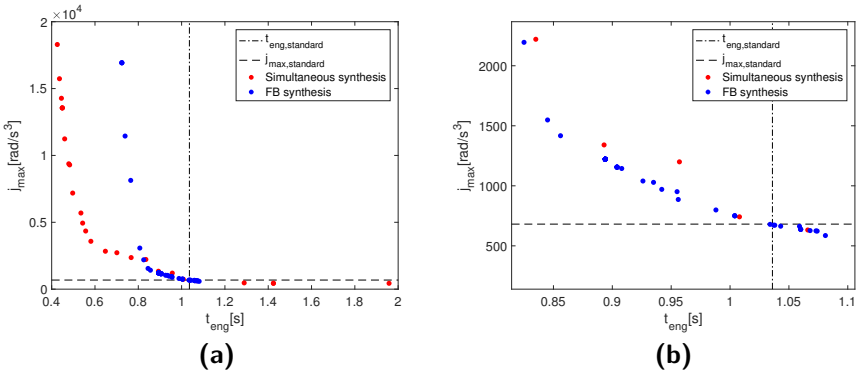


Figure 6-2: The results of the experiment with simultaneous controller synthesis, shown as the red dots, is compared to the results for pure feedback synthesis, shown as the blue dots. Each dot represents the performance of the final result. The baseline performance$(t_{\text{eng,baseline}}, j_{\text{max,baseline}})$ is shown as the dotted lines.

6-3 Decoupled feedforward and feedback controller synthesis

Another approach would be to decouple the synthesis of feedforward and feedback controllers. In that way the feedback controllers are allowed to evolve for a fixed feedforward controller just as in Suproblem 1. To do this we first need to find good feedforward controllers and then synthesize feedback controllers for the best ones. The process we will use can be summarized as:

1. Synthesize pure feedforward controllers for $\text{gen}_{\text{max,ff}}$ generations and $n_{p,\text{ff}}$ individuals.
2. Select $n_{\text{ffselected}}$ feedforward controllers from the resulting non-dominated front.
3. For each selected feedforward controller:
   (a) Compute $p_{\max}, u_{\text{switch}}, x_{1,\text{switch}}, x_{4,\text{switch}}$ and $x_{5,\text{switch}}$.
   (b) Synthesisze feedback controllers using the selected feedforward controller for $\text{gen}_{\text{max,fb}}$ generations and $n_{p,\text{fb}}$ individuals.
4. Designate the result as the non-dominated front for the combination of both the pure feedforward controllers and the combinations of feedforward and feedback controllers.

In order to obtain a diverse set of feedforward controllers we select the $n_{\text{ffselected}}$ feedforward controllers by evenly sampling the non-dominated set.

6-3-1 Results

Name	Value	Meaning
$\text{gen}_{\text{maxff}}$	10	Amount of generations for ff
$\text{gen}_{\text{maxfb}}$	6	Amount of generations for fb
$n_{\text{ffselected}}$	10	Amount of feedforward controllers selected for fb
$n_{p,\text{ff}}$	100	Population size for feedforward synthesis
$n_{p,\text{fb}}$	50	Population size for feedback synthesis
d_{max}	7	Tree depth at which only recursive production rules are used
$n_{e,\text{ff}}$	10	Elitism number for feedforward
$n_{e,\text{fb}}$	5	Elitism number for feedback
$n_{t,\text{ff}}$	25	Tournament size for feedforward
$n_{t,\text{fb}}$	13	Tournament size for feedback
$\text{Pr}_{\text{crossover}}$	50%	Crossover probability
$\text{Pr}_{\text{mutation}}$	50 %	Mutation probability

Table 6-1: The meaning and value of the variables used in the experiment.

The result of a run using the parameters in Table 6-1 can be found in Figure 6-3. We chose a higher population size and amount of generations for feedforward controller synthesis than for feedback controller synthesis, since the feedforward synthesis is done only once.

The red dots indicate the non-dominated set of pure feedforward controllers and the pluses indicates which ones are chosen for feedback controller synthesis. The non-dominated set of combination of feedback and feedforward control for each selected feedforward controller is shown as blue dots. The result for decoupled and simultaneous synthesis is similar for engagement times less than 0.6 seconds, for slower engagement times decoupled synthesis dominates the result of simultaneous synthesis, finding controllers which are significantly better than u_{baseline}. It must be noted, however, that the result for decoupled controller synthesis took more fitness function evaluations (and thus more computation time) than the result for simultaneous synthesis.

It can be seen that the performance for the combinations of feedforward and feedback control coincides with the performance for pure feedforward control for feedforward controllers that have $t_{\text{eng}} < 0.7$s. This effect is illustrated further in Figures 6-5 and Figure 6-6. In Figure 6-5 the non-dominated front of feedback controllers using the fifth selected feedforward controller is shown. It can be seen that feedback control had no influence on the performance of the combination of feedforward and feedback control. In Figure 6-6 the same is plotted for the eighth feedforward controller. For this controller allowing feedback control in the slipping phase did improve upon the performance of the pure feedforward controller. Upon further investigation this effect was due to a too small difference between t_{switch} and t_{eng} for controllers with t_{eng} smaller than about 0.7s. This meant that the feedback control only had a very small impact on the performance. One controller with a significant performance increase compared to u_{baseline}, is $u_{\text{decoupled},1}(t, \xi(t))$ with $u_{\text{ff}}(t) = \frac{0.199409}{1+\exp(-1995.89(t))} - \frac{0.136098}{1+\exp(-504.678(t-0.2693))}$ and $u_{\text{fb}}(\xi(t)) = 0.0633105 + 0.00230891(x_5 - 4.0092)$. This controller achieved a performance of $(0.905, 597.74)$, compared to a performance of $(1.036, 680.979)$ for u_{baseline}). The control input for this controller is plotted in Figure 6-7.

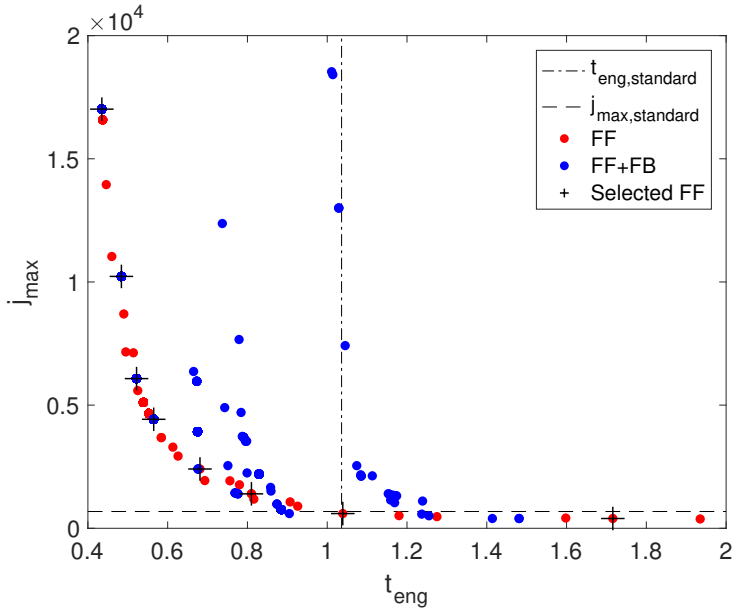


Figure 6-3: Results for the experiment with decoupled controller synthesis, the result for pure feedforward control is represented as the red dots, the selected feedforward controllers are shown as pluses. The results for the combination of feedforward and feedback for each selected feedback controller is shown as blue dots and the standard performance is shown as dotted lines.

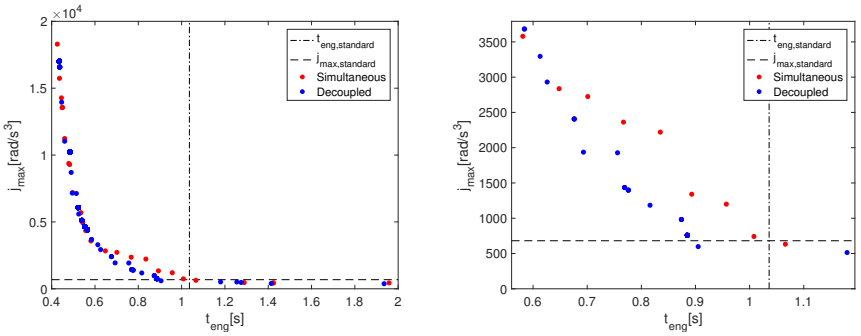


Figure 6-4: Comparison of the results of simultaneous controller synthesis, shown as the red dots, and the results for decoupled controller synthesis, shown as the blue dots. Each dot represents the performance of the final result. The baseline performance($t_{\mathrm{eng,baseline}}, j_{\mathrm{max,baseline}}$) is shown as the dotted lines.

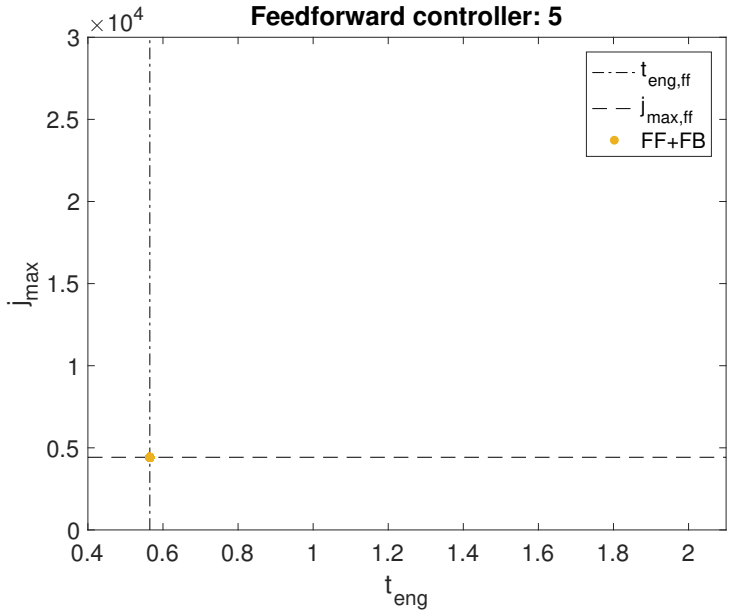


Figure 6-5: Feedback controller results for the fifth selected feedforward controller, it can be seen that the whole non-dominated front is located in one single location.

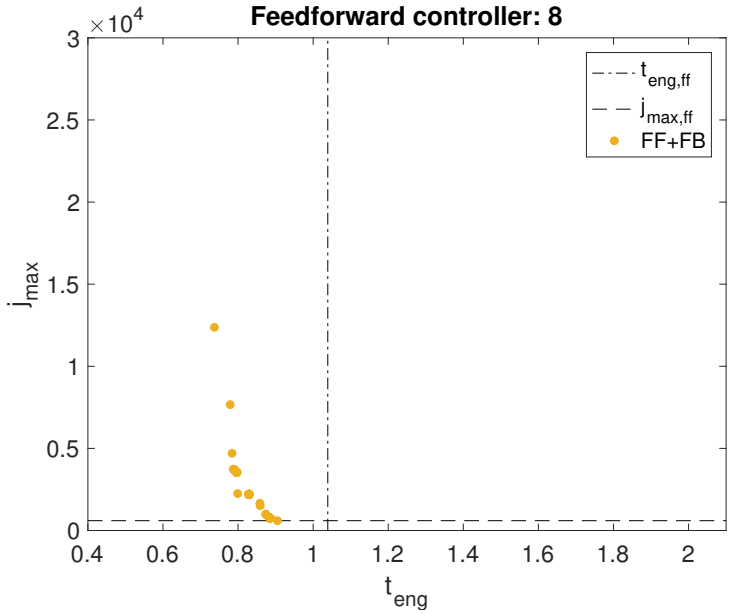


Figure 6-6: Feedback controller results for the eighth selected feedforward controller.

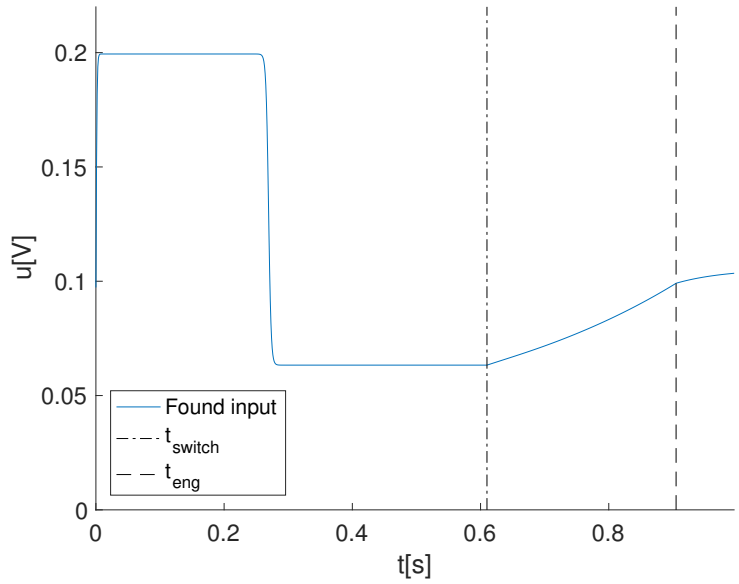


Figure 6-7: Found input of the controller $u_{\text{decoupled},1}$ with $u_{\text{ff}}(t) = \frac{0.199409}{1+\exp(-1995.89(t))} - \frac{0.136098}{1+\exp(-504.678(t-0.2693))}$ and $u_{\text{fb}}(\xi) = 0.0633105 + 0.00230891(x_5 - 4.0092)$.

Chapter 7

Synthesizing feedforward and feedback controllers considering uncertainty

In this Chapter we consider Problem 1 which concerns designing both a feedforward and feedback controller for the uncertain system.

7-1 Sampled uncertainty

Because of the long computation time of computing the worst case performance for an individual, evaluating the worst case performance for a whole population in GP is not practical. Instead we will use a finite amount of simulations for different realizations of the uncertain parameters. Instead of considering all $l \in \mathcal{L}$, we approximate this by considering $l \in \mathcal{L}_{\text{sampled}} \subseteq \mathcal{L}$ where $\mathcal{L}_{\text{sampled}}$ is a finite set. A logical choice would be to take $\mathcal{L}_{\text{sampled}}$ as the most extreme points of $\mathcal{L}$. In Figure 7-1a these four extreme (corner) points are shown in red, 49 evenly sampled other points in $\mathcal{L}$ are shown in blue. In Figure 7-1b the trajectories for the blue points are shown as black lines and the dotted lines represent the extreme values of $x_4(t)$ for the four corner points. It can be seen that most trajectories stay within these approximate bounds for most of the time, however some trajectories do cross the approximate bounds. Therefore taking only the four corner points of $\mathcal{L}$ as $\mathcal{L}_{\text{sampled}}$ covers most of the uncertainty in the model. To give formal guarantees it is still needed to consider the full uncertain set $\mathcal{L}$ and perform reachability analysis. In this case of choosing the four corner points we have

$$\mathcal{L}_{\text{sampled}} = \left\{ \begin{bmatrix} 0.8c_o^{\text{nom}} \\ 0.8\text{frac}_{\text{s}}^{\text{nom}} \end{bmatrix}, \begin{bmatrix} 0.8c_o^{\text{nom}} \\ 1.2\text{frac}_{\text{s}}^{\text{nom}} \end{bmatrix}, \begin{bmatrix} 1.2c_o^{\text{nom}} \\ 0.8\text{frac}_{\text{s}}^{\text{nom}} \end{bmatrix} \begin{bmatrix} 1.2c_o^{\text{nom}} \\ 1.2\text{frac}_{\text{s}}^{\text{nom}} \end{bmatrix} \right\}. \tag{7-1}$$

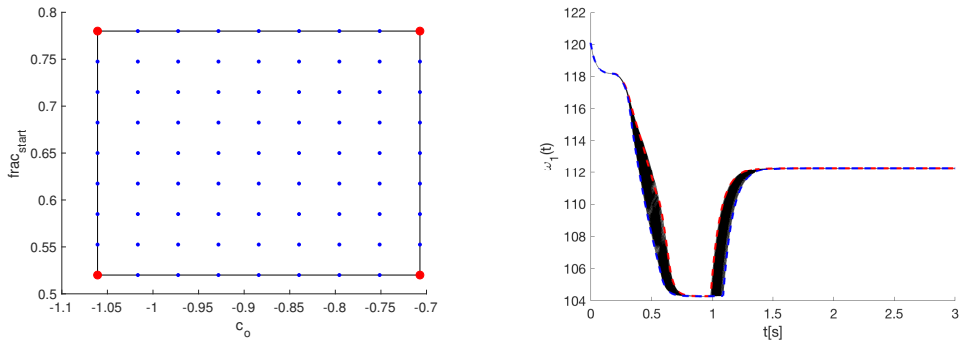
By computing the approximate worst case performance $(t_{\text{eng,apr}}, j_{\text{max,apr}})$ as

$$t_{\text{eng,apr}}^{u,\mathcal{L}} = \max_{l \in \mathcal{L}_{\text{sampled}}} t_{\text{eng}}^{u,l}, \tag{7-2}$$

and

$$j_{\text{eng,apr}}^{u,\mathcal{L}} = \max_{l \in \mathcal{L}_{\text{sampled}}} j_{\max}^{u,l}, \tag{7-3}$$

we have a way to compute a cheap indication of the robustness of an individual. Although it is not a formal guarantee on the performance, it is still useful to select robust individuals in a population.



(a) **(b)**

Figure 7-1: Overview of trajectories for different values of l. In Figure 7-1a the blue points correspond to samples of $(c_o, \text{frac}_\text{s}^\text{nom})$ taken from $\mathcal{L}$ to generate trajectories of the system of which the trajectories $x_4(t)$ are shown in Figure 7-1b as black lines. The black rectangle in Figure 7-1a represents the border of $\mathcal{L}$. The points in $\mathcal{L}_{\text{sampled}}$ are shown as red dots. The upper and lower bounds of the trajectories of $x_4(t)$ generated by the samples in $\mathcal{L}_{\text{sampled}}$ are shown as a red dotted and a blue dotted line in Figure 7-1b respectively. It can be seen that most trajectories, but not all, stay inside these approximate bounds. For the other states no trajectory crossed the approximate bound. This experiment was performed using u_{baseline} and the results be different for other control inputs.

7-2 Results

The result of a run using the same parameters as in Table 6-1 can be found in Figure 7-2. For comparison the approximate worst case performance for $u_{\text{ff,baseline}}$ is plotted. We observe that some of the found pure feedforward controllers are Pareto dominant with respect to $u_{\text{ff,baseline}}$. Surprisingly the addition of feedback lead to little improvement on the performance of the pure feedforward controllers. Several controllers using feedback were found that were faster than any pure feedforward controllers. This may be explained by the fact that we computed $u_{\text{switch}}, x_{1,\text{switch}}, x_{4,\text{switch}}$ and $x_{5,\text{switch}}$ for only one simulation, as before. And because the values of these parameters are not known beforehand when we consider uncertainty, the controller is not guaranteed to be continuous at $t = t_{\text{switch}}$ when using feedback. This effect could cause a faster t_{eng} as well as a higher $j_{\max}$. This issue could be resolved by implementing an interpolation between $u_{\text{ff}}(t)$ and $u_{\text{fb}}(\xi(t))$, instead of the sudden switch we use now.

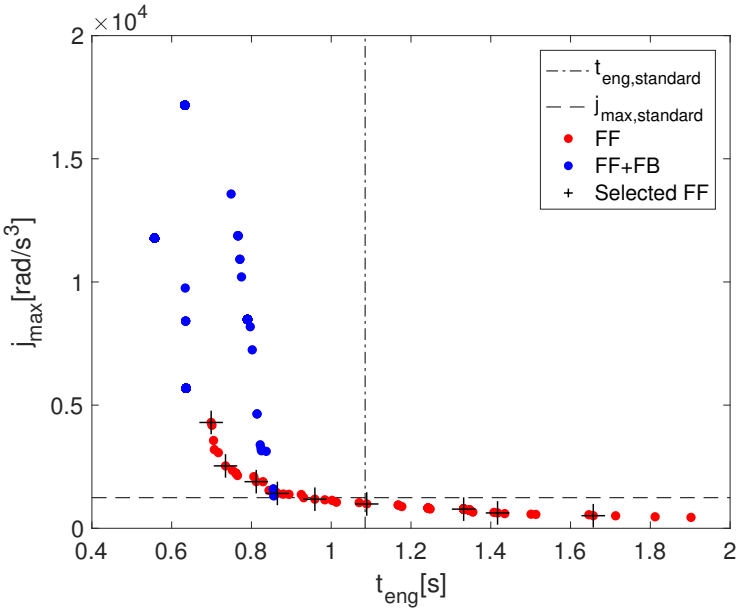


Figure 7-2: ($t_{\mathrm{eng,apr}}$ and $j_{\mathrm{max,apr}}$) plotted for the result of the experiment.

7-2-1 Comparison with previous results

To determine whether selection based on approximate worst case performance helps selecting robust controllers, we compare the results in Section 7-2 to the results from Section 5-3. For the best individuals for the controller synthesis without considering uncertainty, which we will refer to as nominal controller synthesis, we compute the approximate performance and compare it to the results for the robust controller synthesis. This comparison can be found in Figure 7-3. It can be clearly seen that the performance for the robust controller seen is much better than for the nominal controller synthesis. Especially for controllers with a t_{eng} <0.7s the approximate worst case performance for nominal controller synthesis is much worse than for the faster controllers generated by robust controller synthesis.

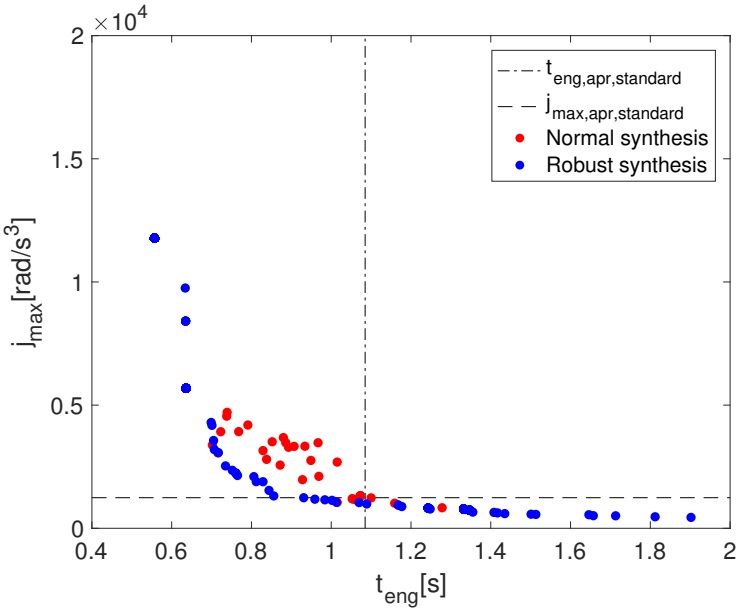


Figure 7-3: Comparison of the results for nominal and robust controller synthesis. The non-dominated front for nominal controller synthesis is shown as the red dots, the non-dominated front for robust controller synthesis is shown as the blue dots. It can be seen that performance of individuals that are selected based on approximate worst case performance are much more robust than individuals that were selected based on a single simulation.

Chapter 8

Conclusions and recommendations

In this Chapter we conclude the thesis and give recommendations for future research.

8-1 Conclusions

In this thesis we have developed a method to automatically synthesize controllers for a wet clutch, that are robust to model uncertainties, without relying on expert knowledge or a fixed structure, in a true multi-objective manner.

We are able give formal guarantees on the robustness of performance for a controller, by modeling the wet clutch system as a hybrid automaton with uncertain parameters and using reachability analysis in Flow*. From the reachable set we are able to compute the worst case performance. We only succeeded in computing the reachable set for simple controllers, however.

First, we have developed feedback controllers, given a fixed feedforward controller taken from a baseline controller, not considering uncertainty. Using this method we have found controllers with satisfying performance, including one controller which was Pareto dominant to the baseline controller. The performance of controllers with a much faster engagement time were not very satisfying.

Next, we synthesized both the feedback and the feedforward controller ourselves. We compared two methods of synthesizing pairs of feedforward and feedback controllers: simultaneous and decoupled controller synthesis. Simultaneous synthesis improved upon the result of using a single fixed feedforward controller in areas further away from the performance of u_{baseline}, but did not find a Pareto dominant controller with respect to the baseline controller. While decoupled synthesis improved in areas both close and far away, finding controllers which were Pareto dominant to u_{baseline}. Therefore, it seems that decoupled synthesis is more suitable for wet clutch controller synthesis, but for proper conclusions on this, a fairer comparison between the two methods using experiments is needed.

We then considered the full uncertain model in the controller synthesis. However, due to the need for a short computation time of the fitness function and because we were not able to compute the reachable set for the controller structure we considered, we used an approximate version of the worst case performance by using sampled uncertainty. Computing the worst case performance by considering only the four corner points of the uncertain set was found to be a reasonable approximation of the actual worst case performance. We found that selection based on this approximate worst case fitness was an appropriate way to automatically synthesize robust controller as they performed much better in terms of worst case fitness than controllers that were selected based on one simulation.

8-2 Recommendations

Several elements in the controller synthesis were not optimized rigorously. The results of the controller synthesis could be improved by optimizing the GP parameters and the controller grammar. Especially the grammar of the feedforward controller could be improved on because we used only a simple parametrized signal for it, not utilizing the power of GP to optimize over the controller structure.

The performance of the controller using feedback and considering uncertainty could be improved on by implementing an interpolation between feedforward and feedback control, instead of the sudden switch between them that we use in this thesis.

A bottleneck in the computation of the formal guarantee on the worst case performance is the big computation that is needed to compute the reachable set and the complexity of the feedforward controller. This could be improved upon exploring other methods and other tools to compute the reachable set or by using different (for example piece-wise affine) structures for the feedforward controller.

In order to truly assess the controllers that we found and compare them to other controllers in the literature on robustness of performance, it would be needed to test them on the real setup.

Appendix A

Constants

Name	Value	Meaning
a	0.0198	Current bias
k	72.1	Gain
s	0.05	Time delay
c_o^{nom}	-0.884	Nominal value of oil pressure constant
a_2	-0.001	Oil pressure to piston position coefficient
b_2	2	Oil pressure to piston position coefficient
c_2	0.002	Oil pressure to piston position coefficient
d_2	$5.2 \cdot 10^{-4}$	Oil pressure to piston position coefficient
k_t	$1.797 \cdot 10^{-6}$	Gain related to oil temperature
pos_c	0.005	Maximum piston displacement
$\text{frac}_\text{s}^{\text{nom}}$	0.65	Nominal value of the fraction of pos_c at which $p(t) = x_1(t)$ becomes true
ω_m	125.6627	Motor speed
α	$2.6064196 \cdot 10^{-4} p_{\max} - 7.2091633 \cdot 10^{-4}$	Constant used in dynamics of $x_4(t)$ and $x_5(t)$
$p_{\max}$	-	Oil pressure at piston contact
J_1	0.1	Input shaft inertia
J_3	2.706	Output shaft inertia
T_{1c}	0.0001	Input shaft Coulomb friction torque
T_{3c}	94.5	Output shaft Coulomb friction torque
b_{1v}	0.3764	Input shaft viscous friction coefficient
b_{3v}	0.15	Output shaft viscous friction coefficient
g_{rl}	5.036	Gear ratio
γ	-10^{-3}	Constant to model first phase of torque transfer
w_{th}	10^{-3}	Threshold to avoid non-physical motion
c_t	0.582	Constant used in dynamics of x_6

Table A-1: The meaning and value of the variables in the model described in Chapter 2.

Appendix B

Jerk maps

B-1 Mode 1

$$
\begin{aligned}
j_{1,\xi}(\xi) = &-\frac{1.93\gamma\omega_m^2}{J_1\mathrm{pos_c}\left(11.8\ +e^{\frac{5.72x_4}{\omega_m}-3.77}\right)^2}+\frac{0.546\ \gamma x_4^2}{J_1\mathrm{pos_c}\left(11.8\ +e^{\frac{5.72x_4}{\omega_m}-3.77}\right)^2}\\
&+\frac{0.605\gamma\omega_m x_4}{J_1\mathrm{pos_c}\left(11.8\ +e^{\frac{5.72x_4}{\omega_m}-3.77}\right)^2}+\frac{b_{1v}\gamma x_4}{J_1\mathrm{pos_c}}-\frac{b_{3v}\gamma g_{rl}x_5}{J_3\ \mathrm{pos_c}}+\frac{\gamma^2 g_{rl}^2 x_5}{J_3{\mathrm{pos_c}}^2}\\
&-\frac{\gamma^2 x_4}{g_{rl}J_1{\mathrm{pos_c}}^2}-\frac{\gamma^2 g_{rl}x_4}{J_3{\mathrm{pos_c}}^2}+\frac{\gamma^2 x_5}{J_1{\mathrm{pos_c}}^2}-\frac{\gamma g_{rl}T_{3c}x_5}{J_3\ \mathrm{pos_c}w_{\mathrm{th}}}+\frac{\gamma T_{1c}}{J_1\mathrm{pos_c}}
\end{aligned}
\tag{B-1}
$$

B-2 Mode 2

$$
\begin{aligned}
j_{2,\xi}(\xi) = &- \frac{\gamma\left(\frac{-1138.51\omega_m x_4 - 1027.48x_4^2 + 3631.93\omega_m^2}{\left(511.885 + 1.e^{\frac{5.72x_4}{\omega_m}}\right)^2} - b_{1v}x_4 + \frac{\gamma(x_4 - g_{rl}x_5)}{g_{rl}(\mathrm{pos_c} - k_t x_3)} - T_{1c}\right)}{J_1(\mathrm{pos_c} - k_t x_3)} \\
&+ \frac{\gamma\left(-\frac{a_2 k_t x_4\left(x_2 - \frac{2ku_{\mathrm{ff}}(t)}{s}\right)}{c_2} + \frac{c_o b_2 k_t x_4}{c_2} - \frac{b_2 k_t x_1 x_4}{c_2}\right)}{(\mathrm{pos_c} - k_t x_3)^2} \\
&+ \frac{\gamma\left(\frac{a_2 g_{rl} k_t x_5\left(x_2 - \frac{2ku_{\mathrm{ff}}(t)}{s}\right)}{c_2} - \frac{d_2 g_{rl} k_t x_3 x_5}{c_2}\right)}{(\mathrm{pos_c} - k_t x_3)^2} \\
&+ \frac{\gamma\left(-\frac{c_o b_2 g_{rl} k_t x_5}{c_2} + \frac{b_2 g_{rl} k_t x_1 x_5}{c_2} + \frac{d_2 k_t x_3 x_4}{c_2}\right)}{(\mathrm{pos_c} - k_t x_3)^2} \\
&- \frac{\gamma\left(\frac{g_{rl}x_5(b_{3v}w_{\mathrm{th}} + T_{3c})(\mathrm{pos_c} - k_t x_3)}{J_3 w_{\mathrm{th}}} + \frac{\gamma g_{rl}(g_{rl}x_5 - x_4)}{J_3}\right)}{(\mathrm{pos_c} - k_t x_3)^2}
\end{aligned}
\tag{B-2}
$$

B-3 Mode 3

$$
\begin{aligned}
j_{3,\xi}(\xi) = &- \frac{x_4 x_6^2 \gamma^2}{g_{rl} J_1(\mathrm{pos_c} - k_t x_3)^2} - \frac{g_{rl} x_4 x_6^2 \gamma^2}{J_3(\mathrm{pos_c} - k_t x_3)^2} + \frac{x_5 x_6^2 \gamma^2}{J_1(\mathrm{pos_c} - k_t x_3)^2} \\
&+ \frac{g_{rl}^2 x_5 x_6^2 \gamma^2}{J_3(\mathrm{pos_c} - k_t x_3)^2} - \frac{x_4 \gamma^2}{g_{rl} J_1(\mathrm{pos_c} - k_t x_3)^2} - \frac{g_{rl} x_4 \gamma^2}{J_3(\mathrm{pos_c} - k_t x_3)^2} \\
&+ \frac{x_5 \gamma^2}{J_1(\mathrm{pos_c} - k_t x_3)^2} + \frac{g_{rl}^2 x_5 \gamma^2}{J_3(\mathrm{pos_c} - k_t x_3)^2} + \frac{2x_4 x_6 \gamma^2}{g_{rl} J_1(\mathrm{pos_c} - k_t x_3)^2} \\
&+ \frac{2g_{rl} x_4 x_6 \gamma^2}{J_3\ (\mathrm{pos_c} - k_t x_3)^2} - \frac{2x_5 x_6 \gamma^2}{J_1(\mathrm{pos_c} - k_t x_3)^2} - \frac{2g_{rl}^2 x_5 x_6 \gamma^2}{J_3(\mathrm{pos_c} - k_t x_3)^2} \\
&+ \frac{1027.48 x_4^2 \gamma}{\left(511.885\ + 1.e^{\frac{5.72x_4}{\omega_m}}\right)^2 J_1\ (\mathrm{pos_c} - k_t x_3)} - \frac{10^5 \alpha \mathrm{frac_s} x_1 x_6^2 \gamma}{(\mathrm{frac_s} - 1) J_1(\mathrm{pos_c} - k_t x_3)} \\
&- \frac{10^5 \alpha \mathrm{frac_s} g_{rl}^2 x_1 x_6^2 \gamma}{(\mathrm{frac_s} - 1) J_3(\mathrm{pos_c} - k_t x_3)} + \frac{10^5 \alpha k_t x_1 x_3 x_6^2 \gamma}{(\mathrm{frac_s} - 1) J_1 \mathrm{pos_c}(\mathrm{pos_c} - k_t x_3)} \\
&+ \frac{10^5 \alpha g_{rl}^2 k_t x_1 x_3 x_6^2 \gamma}{(\mathrm{frac_s} - 1) J_3 \mathrm{pos_c}(\mathrm{pos_c} - k_t x_3)} - \frac{10^5 \alpha \mathrm{frac_s} x_6^2 \gamma}{(\mathrm{frac_s} - 1) J_1(\mathrm{pos_c} - k_t x_3)} \\
&+ \frac{10^5 \alpha x_6^2 \gamma}{(\mathrm{frac_s} - 1) J_1(\mathrm{pos_c} - k_t x_3)} - \frac{10^5 \alpha \mathrm{frac_s} g_{rl}^2 x_6^2 \gamma}{(\mathrm{frac_s} - 1) J_3(\mathrm{pos_c} - k_t x_3)}
\end{aligned}
$$

$$
\begin{aligned}
&+\frac{10^5\alpha g_{rl}^2 x_6^2\gamma}{(\mathrm{frac_s}-1)J_3(\mathrm{pos_c}-k_t x_3)}+\frac{1138.51\omega_m x_4\gamma}{\left(511.885+1.e^{\frac{5.72x_4}{\omega_m}}\right)^2 J_1(\mathrm{pos_c}-k_t x_3)}\\
&+\frac{b_{1v}x_4\gamma}{J_1(\mathrm{pos_c}-k_t x_3)}-\frac{b_{3v}g_{rl}x_5\gamma}{J_3(\mathrm{pos_c}-k_t x_3)}-\frac{g_{rl}T_{3c}x_5\gamma}{J_3 w_{\mathrm{th}}(\mathrm{pos_c}-k_t x_3)}\\
&+\frac{b_{3v}(x_4-g_{rl}x_5)(x_6-1)\gamma}{J_3(\mathrm{pos_c}-k_t x_3)}+\frac{T_{3c}(x_4-g_{rl}x_5)(x_6-1)\gamma}{J_3 w_{\mathrm{th}}(\mathrm{pos_c}-k_t x_3)}\\
&+\frac{k_t(2a_2 k u_{\mathrm{ff}}(t)+s(ob_2-x_1b_2-a_2x_2+d_2x_3))(g_{rl}x_5-x_4)(x_6-1)\gamma}{c_2 s(\mathrm{pos_c}-k_t x_3)^2}\\
&-\frac{1027.48x_4^2x_6\gamma}{\left(511.885+1.e^{\frac{5.72x_4}{\omega_m}}\right)^2 J_1(\mathrm{pos_c}-k_t x_3)}-\frac{1138.51\omega_m x_4x_6\gamma}{\left(511.885+1.e^{\frac{5.72x_4}{\omega_m}}\right)^2 J_1(\mathrm{pos_c}-k_t x_3)}\\
&-\frac{b_{1v}x_4x_6\gamma}{J_1(\mathrm{pos_c}-k_t x_3)}+\frac{b_{3v}g_{rl}x_5x_6\gamma}{J_3(\mathrm{pos_c}-k_t x_3)}+\frac{g_{rl}T_{3c}x_5x_6\gamma}{J_3 w_{\mathrm{th}}(\mathrm{pos_c}-k_t x_3)}\\
&+\frac{3631.93\omega_m^2x_6\gamma}{\left(511.885+1.e^{\frac{5.72x_4}{\omega_m}}\right)^2 J_1(\mathrm{pos_c}-k_t x_3)}-\frac{T_{1c}x_6\gamma}{J_1(\mathrm{pos_c}-k_t x_3)}\\
&+\frac{10^5\alpha\mathrm{frac_s}x_1x_6\gamma}{(\mathrm{frac_s}-1)J_1(\mathrm{pos_c}-k_t x_3)}+\frac{10^5\alpha\mathrm{frac_s}g_{rl}^2x_1x_6\gamma}{(\mathrm{frac_s}-1)J_3(\mathrm{pos_c}-k_t x_3)}\\
&-\frac{10^5\alpha k_t x_1x_3x_6\gamma}{(\mathrm{frac_s}-1)J_1\mathrm{pos_c}(\mathrm{pos_c}-k_t x_3)}-\frac{10^5\alpha g_{rl}^2k_t x_1x_3x_6\gamma}{(\mathrm{frac_s}-1)J_3\mathrm{pos_c}(\mathrm{pos_c}-k_t x_3)}\\
&+\frac{10^5\alpha\mathrm{frac_s}x_6\gamma}{(\mathrm{frac_s}-1)J_1(\mathrm{pos_c}-k_t x_3)}-\frac{10^5\alpha x_6\gamma}{(\mathrm{frac_s}-1)J_1(\mathrm{pos_c}-k_t x_3)}\\
&+\frac{10^5\alpha\mathrm{frac_s}g_{rl}^2x_6\gamma}{(\mathrm{frac_s}-1)J_3(\mathrm{pos_c}-k_t x_3)}-\frac{10^5\alpha g_{rl}^2x_6\gamma}{(\mathrm{frac_s}-1)J_3(\mathrm{pos_c}-k_t x_3)}\\
&+\frac{\beta(x_4-g_{rl}x_5)(c_t+x_6)\gamma}{\mathrm{pos_c}-k_t x_3}-\frac{3631.93\omega_m^2\gamma}{\left(511.885+1.e^{\frac{5.72x_4}{\omega_m}}\right)^2 J_1(\mathrm{pos_c}-k_t x_3)}\\
&+\frac{T_{1c}\gamma}{J_1(\mathrm{pos_c}-k_t x_3)}-\frac{b_{3v}(T_{3c}+b_{3v}w_{\mathrm{th}})x_5}{J_3 w_{\mathrm{th}}}-\frac{T_{3c}(T_{3c}+b_{3v}w_{\mathrm{th}})x_5}{J_3 w_{th}^2}\\
&+\frac{10^5\alpha b_{3v}g_{rl}((\mathrm{frac_s}-1)\mathrm{pos_c}+x_1(\mathrm{frac_s}\mathrm{pos_c}-k_t x_3))x_6}{(\mathrm{frac_s}-1)J_3\mathrm{pos_c}}\\
&+\frac{10^5\alpha g_{rl}T_{3c}((\mathrm{frac_s}-1)\mathrm{pos_c}+x_1(\mathrm{frac_s}\mathrm{pos_c}-k_t x_3))x_6}{(\mathrm{frac_s}-1)J_3\mathrm{pos_c}w_{\mathrm{th}}}\\
&+\frac{2\cdot10^5a_2\alpha g_{rl}kk_t u_{\mathrm{ff}}(t)x_1x_6}{c_2(\mathrm{frac_s}-1)\mathrm{pos_c}s}-\frac{10^5a_2\alpha g_{rl}k_t x_1x_2x_6}{c_2(\mathrm{frac_s}-1)\mathrm{pos_c}}+\frac{10^5c_o\alpha b_2g_{rl}k_t x_1x_6}{c_2(\mathrm{frac_s}-1)\mathrm{pos_c}}\\
&-\frac{10^5\alpha b_2g_{rl}k_t x_1^2x_6}{c_2(\mathrm{frac_s}-1)\mathrm{pos_c}}+\frac{10^5\alpha d_2g_{rl}k_t x_1x_3x_6}{c_2(\mathrm{frac_s}-1)\mathrm{pos_c}}+\frac{2\cdot10^5\alpha g_{rl}kk_t u_{\mathrm{ff}}(t)x_3x_6}{(\mathrm{frac_s}-1)\mathrm{pos_c}s}\\
&-\frac{2\cdot10^5\alpha\mathrm{frac_s}g_{rl}ku_{\mathrm{ff}}(t)x_6}{(\mathrm{frac_s}-1)s}-\frac{10^5\alpha g_{rl}k_t x_2x_3x_6}{(\mathrm{frac_s}-1)\mathrm{pos_c}}+\frac{10^5\alpha\mathrm{frac_s}g_{rl}x_2x_6}{\mathrm{frac_s}-1}\\
&+\frac{10^5\alpha\beta g_{rl}((\mathrm{frac_s}-1)\mathrm{pos_c}+x_1(\mathrm{frac_s}\mathrm{pos_c}-k_t x_3))(c_t+x_6)}{(\mathrm{frac_s}-1)\mathrm{pos_c}}
\end{aligned}
$$

$$
\begin{aligned}
&- \frac{10^5 \alpha b_{3v} g_{rl} k_t w_{\text{th}} x_1 x_3 x_6}{(\text{frac}_\text{s} - 1)\text{pos}_\text{c}} + \frac{10^5 \alpha b_{3v} \text{frac}_\text{s} g_{rl} w_{\text{th}} x_1 x_6}{\text{frac}_\text{s} - 1} + \frac{10^5 \alpha b_{3v} \text{frac}_\text{s} g_{rl} w_{\text{th}} x_6}{\text{frac}_\text{s} - 1} \\
&- \frac{10^5 \alpha b_{3v} g_{rl} w_{\text{th}} x_6}{\text{frac}_\text{s} - 1} - \frac{10^5 \alpha g_{rl} k_t T_{3c} x_1 x_3 x_6}{(\text{frac}_\text{s} - 1)\text{pos}_\text{c}} + \frac{10^5 \alpha \text{frac}_\text{s} g_{rl} T_{3c} x_1 x_6}{\text{frac}_\text{s} - 1} \\
&+ \frac{10^5 \alpha \text{frac}_\text{s} g_{rl} T_{3c} x_6}{\text{frac}_\text{s} - 1} - \frac{10^5 \alpha g_{rl} T_{3c} x_6}{\text{frac}_\text{s} - 1} + b_{3v}^2 (-w_{\text{th}}) x_5 - \frac{b_{3v} \gamma g_{rl} w_{\text{th}} x_5 x_6}{\text{pos}_\text{c} - k_t x_3} \\
&+ \frac{b_{3v} \gamma g_{rl} w_{\text{th}} x_5}{\text{pos}_\text{c} - k_t x_3} + \frac{b_{3v} \gamma w_{\text{th}} x_4 x_6}{\text{pos}_\text{c} - k_t x_3} - \frac{b_{3v}\ \gamma w_{\text{th}} x_4}{\text{pos}_\text{c} - k_t x_3} - 2 b_{3v} T_{3c} x_5 \\
&- \frac{\gamma g_{rl} T_{3c} x_5 x_6}{\text{pos}_\text{c} - k_t x_3} + \frac{\gamma g_{rl} T_{3c} x_5}{\text{pos}_\text{c} - k_t x_3} + \frac{\gamma T_{3c} x_4 x_6}{\text{pos}_\text{c} - k_t x_3} - \frac{\gamma T_{3c} x_4}{\text{pos}_\text{c} - k_t x_3} - \frac{T_{3c}^2 x_5}{w_{\text{th}}}
\end{aligned} \tag{B-3}
$$

B-4 Mode 4

$$
\begin{aligned}
j_{4,\xi}(\xi, u_{\text{ff}}(t)) = &- \frac{x_5 b_{3v}^2}{J_3} - \frac{T_{3c} b_{3v}}{J_3} - \frac{\gamma\ g_{rl} x_5 b_{3v}}{J_3(\text{pos}_\text{c} - k_t x_3)} + \frac{\gamma (x_4 - g_{rl} x_5)(x_6 - 1) b_{3v}}{J_3\ (\text{pos}_\text{c} - k_t x_3)} \\
&+ \frac{10^5 \alpha\ \text{frac}_\text{s} g_{rl} x_1 x_6\ b_{3v}}{(\text{frac}_\text{s} - 1) J_3} - \frac{10^5 \alpha g_{rl} k_t\ x_1 x_3 x_6 b_{3v}}{(\text{frac}_\text{s} - 1) J_3\ \text{pos}_\text{c}} + \frac{\gamma g_{rl} x_5 x_6\ b_{3v}}{J_3(\text{pos}_\text{c} - k_t x_3)} \\
&+ \frac{10^5 \alpha\ \text{frac}_\text{s} g_{rl} x_6 b_{3v}}{(\text{frac}_\text{s} - 1)\ J_3} - \frac{10^5 \alpha g_{rl} x_6\ b_{3v}}{(\text{frac}_\text{s} - 1) J_3} \\
&- \frac{b_{3v}}{J_3} \left(-T_{3c} - b_{3v}\ x_5 + \frac{\gamma (x_4 - g_{rl} x_5)\ (x_6 - 1)}{\text{pos}_\text{c} - k_t x_3} \right) \\
&+ \frac{b_{3v}}{J_3} \left(\frac{10^5 \alpha\ g_{rl}((\text{frac}_\text{s} - 1)\text{pos}_\text{c} + x_1\ (\text{frac}_\text{s}\text{pos}_\text{c} - k_t x_3))\ x_6}{(\text{frac}_\text{s} - 1)\text{pos}_\text{c}} \right) \\
&+ \frac{1027.48 \gamma x_4^2}{\left(511.885\ + 1.\ e^{\frac{5.72 x_4}{\omega_m}} \right)^2 J_1\ (\text{pos}_\text{c} - k_t x_3)} - \frac{\gamma^2 x_4\ x_6^2}{g_{rl} J_1 (\text{pos}_\text{c} - k_t\ x_3)^2} \\
&- \frac{\gamma^2 g_{rl} x_4 x_6^2}{J_3\ (\text{pos}_\text{c} - k_t x_3)^2} + \frac{\gamma^2 x_5\ x_6^2}{J_1 (\text{pos}_\text{c} - k_t x_3)^2} + \frac{\gamma^2\ g_{rl}^2 x_5 x_6^2}{J_3 (\text{pos}_\text{c} - k_t x_3)^2} \\
&- \frac{10^5 \alpha \text{frac}_\text{s} \gamma x_1\ x_6^2}{(\text{frac}_\text{s} - 1) J_1 (\text{pos}_\text{c} - k_t\ x_3)} - \frac{10^5 \alpha \text{frac}_\text{s} \gamma g_{rl}^2\ x_1 x_6^2}{(\text{frac}_\text{s} - 1) J_3\ (\text{pos}_\text{c} - k_t x_3)} \\
&+ \frac{10^5 \alpha \gamma k_t\ x_1 x_3 x_6^2}{(\text{frac}_\text{s} - 1) J_1\ \text{pos}_\text{c} (\text{pos}_\text{c} - k_t\ x_3)} + \frac{10^5 \alpha \gamma g_{rl}^2 k_t x_1 x_3\ x_6^2}{(\text{frac}_\text{s} - 1) J_3 \text{pos}_\text{c}\ (\text{pos}_\text{c} - k_t x_3)} \\
&- \frac{10^5 \alpha\ \text{frac}_\text{s} \gamma x_6^2}{(\text{frac}_\text{s} - 1) J_1\ (\text{pos}_\text{c} - k_t x_3)} + \frac{10^5 \alpha \gamma\ x_6^2}{(\text{frac}_\text{s} - 1) J_1 (\text{pos}_\text{c} - k_t\ x_3)} \\
&- \frac{10^5 \alpha \text{frac}_\text{s} \gamma g_{rl}^2\ x_6^2}{(\text{frac}_\text{s} - 1) J_3 (\text{pos}_\text{c} - k_t\ x_3)} + \frac{10^5 \alpha \gamma g_{rl}^2\ x_6^2}{(\text{frac}_\text{s} - 1) J_3 (\text{pos}_\text{c} - k_t\ x_3)} \\
&+ \frac{1138.51 \gamma \omega_m x_4}{\left(511.885\ + 1.\ e^{\frac{5.72 x_4}{\omega_m}} \right)^2 J_1\ (\text{pos}_\text{c} - k_t x_3)} + \frac{b_{1v} \gamma x_4}{J_1\ (\text{pos}_\text{c} - k_t x_3)} \\
&+ \frac{c_o b_2 \gamma k_t\ x_4}{c_2 (\text{pos}_\text{c} - k_t x_3)^2} + \frac{2 a_2 \gamma\ k k_t u_{\text{ff}}(t) x_4}{c_2 s (\text{pos}_\text{c} - k_t\ x_3)^2} - \frac{b_2 \gamma k_t x_1 x_4}{c_2\ (\text{pos}_\text{c} - k_t x_3)^2}
\end{aligned} \tag{B-4}
$$

$$
\begin{aligned}
&- \frac{a_2\gamma k_t x_2\ x_4}{c_2(\mathrm{pos_c}-k_t x_3)^2} + \frac{d_2\gamma\ k_t x_3 x_4}{c_2(\mathrm{pos_c}-k_t\ x_3)^2} - \frac{\gamma^2 x_4}{g_{rl}J_1\ (\mathrm{pos_c}-k_t x_3)^2} \\
&- \frac{\gamma^2 g_{rl}\ x_4}{J_3(\mathrm{pos_c}-k_t x_3)^2} - \frac{c_o b_2\gamma\ g_{rl}k_t x_5}{c_2(\mathrm{pos_c}-k_t\ x_3)^2} - \frac{2a_2\gamma g_{rl}kk_t u_{\mathrm{ff}}(t)x_5}{c_2 s(\ \mathrm{pos_c}-k_t x_3)^2} \\
&+ \frac{b_2\gamma g_{rl}k_t\ x_1 x_5}{c_2(\mathrm{pos_c}-k_t x_3)^2} + \frac{a_2\ \gamma g_{rl}k_t x_2 x_5}{c_2(\mathrm{pos_c}-k_t\ x_3)^2} - \frac{d_2\gamma g_{rl}k_t x_3 x_5}{c_2\ (\mathrm{pos_c}-k_t x_3)^2} \\
&+ \frac{\gamma^2 x_5}{J_1\ (\mathrm{pos_c}-k_t x_3)^2} + \frac{\gamma^2 g_{rl}^2\ x_5}{J_3(\mathrm{pos_c}-k_t x_3)^2} - \frac{10^5\alpha\ b_2 g_{rl}k_t x_1^2 x_6}{c_2(\mathrm{frac_s}-1)\ \mathrm{pos_c}} \\
&- \frac{1027.48\gamma x_4^2\ x_6}{\left(511.885\ +1.e^{\frac{5.72x_4}{\omega_m}}\right)^2 J_1(\mathrm{pos_c}-k_t x_3)} - \frac{2\cdot 10^5\alpha\ \mathrm{frac_s} g_{rl}k u_{\mathrm{ff}}(t)\ x_6}{(\mathrm{frac_s}-1)s} \\
&+ \frac{2\cdot 10^5 a_2\alpha g_{rl}\ kk_t u_{\mathrm{ff}}(t)x_1 x_6}{c_2(\mathrm{frac_s}-1)\ \mathrm{pos_c}s} + \frac{10^5 c_o\alpha b_2 g_{rl}k_t x_1\ x_6}{c_2(\mathrm{frac_s}-1)\ \mathrm{pos_c}} + \frac{10^5\alpha\mathrm{frac_s}\ g_{rl}x_2 x_6}{\mathrm{frac_s}-1} \\
&- \frac{10^5 a_2\alpha\ g_{rl}k_t x_1 x_2 x_6}{c_2(\mathrm{frac_s}-1)\ \mathrm{pos_c}} + \frac{2\cdot 10^5\alpha g_{rl}kk_t u_{\mathrm{ff}}(t)\ x_3 x_6}{(\mathrm{frac_s}-1)\mathrm{pos_c}\ s} + \frac{10^5\alpha d_2 g_{rl}k_t x_1 x_3 x_6}{c_2\ (\mathrm{frac_s}-1)\mathrm{pos_c}} \\
&- \frac{10^5\alpha\ g_{rl}k_t x_2 x_3 x_6}{(\mathrm{frac_s}-1)\ \mathrm{pos_c}} - \frac{1138.51\gamma\omega_m x_4\ x_6}{\left(511.885\ +1.e^{\frac{5.72x_4}{\omega_m}}\right)^2 J_1(\mathrm{pos_c}-k_t x_3)} \\
&- \frac{b_{1v}\gamma x_4\ x_6}{J_1(\mathrm{pos_c}-k_t x_3)} - \frac{c_o b_2\gamma\ k_t x_4 x_6}{c_2(\mathrm{pos_c}-k_t x_3)^2} - \frac{2\ a_2\gamma kk_t u_{\mathrm{ff}}(t)x_4 x_6}{c_2 s\ (\mathrm{pos_c}-k_t x_3)^2} \\
&+ \frac{b_2\gamma k_t x_1\ x_4 x_6}{c_2(\mathrm{pos_c}-k_t x_3)^2} + \frac{a_2\ \gamma k_t x_2 x_4 x_6}{c_2(\mathrm{pos_c}-k_t\ x_3)^2} - \frac{d_2\gamma k_t x_3 x_4 x_6}{c_2\ (\mathrm{pos_c}-k_t x_3)^2} \\
&+ \frac{2\gamma^2 x_4\ x_6}{g_{rl}J_1(\mathrm{pos_c}-k_t x_3)^2} + \frac{2\ \gamma^2 g_{rl}x_4 x_6}{J_3(\mathrm{pos_c}-k_t\ x_3)^2} + \frac{c_o b_2\gamma g_{rl}k_t x_5 x_6}{c_2\ (\mathrm{pos_c}-k_t x_3)^2} \\
&+ \frac{2a_2\gamma g_{rl}k\ k_t u_{\mathrm{ff}}(t)x_5 x_6}{c_2 s(\mathrm{pos_c}-k_t\ x_3)^2} - \frac{b_2\gamma g_{rl}k_t x_1 x_5 x_6}{c_2\ (\mathrm{pos_c}-k_t x_3)^2} - \frac{a_2\gamma g_{rl}k_t\ x_2 x_5 x_6}{c_2(\mathrm{pos_c}-k_t\ x_3)^2} \\
&+ \frac{d_2\gamma g_{rl}k_t x_3 x_5 x_6}{c_2\ (\mathrm{pos_c}-k_t x_3)^2} - \frac{2\gamma^2 x_5\ x_6}{J_1(\mathrm{pos_c}-k_t x_3)^2} - \frac{2\gamma^2\ g_{rl}^2 x_5 x_6}{J_3(\mathrm{pos_c}-k_t\ x_3)^2} \\
&+ \frac{3631.93\gamma\omega_m^2 x_6}{\left(511.885\ \ +1.e^{\frac{5.72x_4}{\omega_m}}\right)^2 J_1\ (\mathrm{pos_c}-k_t x_3)} - \frac{\gamma T_{1c}x_6}{J_1\ (\mathrm{pos_c}-k_t x_3)} \\
&+ \frac{\gamma g_{rl}T_{3c}\ x_6}{J_3(\mathrm{pos_c}-k_t x_3)} + \frac{10^5\alpha\ \mathrm{frac_s}\gamma x_1\ x_6}{(\mathrm{frac_s}-1)J_1(\mathrm{pos_c}-k_t\ x_3)} + \frac{10^5\alpha\mathrm{frac_s}\gamma g_{rl}^2\ x_1 x_6}{(\mathrm{frac_s}-1)J_3\ (\mathrm{pos_c}-k_t x_3)} \\
&- \frac{10^5\alpha\gamma k_t\ x_1 x_3 x_6}{(\mathrm{frac_s}-1)J_1\ \mathrm{pos_c}(\mathrm{pos_c}-k_t\ x_3)} - \frac{10^5\alpha\gamma g_{rl}^2 k_t x_1 x_3\ x_6}{(\mathrm{frac_s}-1)J_3\mathrm{pos_c}\ (\mathrm{pos_c}-k_t x_3)} \\
&+ \frac{10^5\alpha\ \mathrm{frac_s}\gamma x_6}{(\mathrm{frac_s}-1)J_1(\ \mathrm{pos_c}-k_t x_3)} - \frac{10^5\alpha\gamma\ x_6}{(\mathrm{frac_s}-1)J_1(\mathrm{pos_c}-k_t\ x_3)} \\
&+ \frac{10^5\alpha\mathrm{frac_s}\gamma g_{rl}^2\ x_6}{(\mathrm{frac_s}-1)J_3(\mathrm{pos_c}-k_t\ x_3)} - \frac{10^5\alpha\gamma g_{rl}^2\ x_6}{(\mathrm{frac_s}-1)J_3(\mathrm{pos_c}-k_t\ x_3)} \\
&+ \frac{10^5\alpha\beta g_{rl}((\mathrm{frac_s}-1)\ \mathrm{pos_c}+x_1(\mathrm{frac_s}\ \mathrm{pos_c}-k_t x_3))\ (c_t+x_6)}{(\mathrm{frac_s}-1)\ \mathrm{pos_c}}
\end{aligned}
$$

$$+\frac{\beta\gamma(x_4-g_{rl}x_5)\ (c_t+x_6)}{\text{pos}_\text{c}-k_tx_3}-\frac{3631.93\gamma\ \omega_m^2}{\left(511.885\ +1.e^{\frac{5.72x_4}{\omega_m}}\right)^2 J_1(\text{pos}_\text{c}-k_tx_3)}$$
$$+\frac{\gamma T_{1c}}{J_1(\text{pos}_\text{c}-k_tx_3)}-\frac{\gamma g_{rl}T_{3c}}{J_3(\text{pos}_\text{c}-k_tx_3)} \tag{B-5}$$

B-5 Mode 5

$$\begin{aligned}j_{5,\xi}(\xi(y),u_\text{ff}(t)) = &-\frac{2\cdot10^5a_2\alpha g_{rl}kk_tu_\text{ff}(t)x_1}{c_2\ (1-\text{frac}_\text{s})\text{pos}_\text{c}s}+\frac{10^5a_2\alpha g_{rl}k_tx_1x_2}{c_2(1-\text{frac}_\text{s})\text{pos}_\text{c}}-\frac{10^5c_o\ \alpha b_2g_{rl}k_tx_1}{c_2(1-\text{frac}_\text{s})\ \text{pos}_\text{c}}\\&+\frac{10^5\alpha b_2g_{rl}k_tx_1^2}{c_2\ (1-\text{frac}_\text{s})\text{pos}_\text{c}}-\frac{10^5\alpha d_2g_{rl}k_tx_1x_3}{c_2(1-\text{frac}_\text{s})\text{pos}_\text{c}}-\frac{2\cdot10^5\ \alpha g_{rl}kk_tu_\text{ff}(t)x_3}{(1-\text{frac}_\text{s})\text{pos}_\text{c}\ s}\\&+\frac{2\cdot10^5\alpha\text{frac}_\text{s}g_{rl}k\ u_\text{ff}(t)}{(1-\text{frac}_\text{s})s}+\frac{10^5\alpha g_{rl}k_tx_2x_3}{(1-\text{frac}_\text{s})\text{pos}_\text{c}}\\&-\frac{10^5\alpha\text{frac}_\text{s}\ g_{rl}x_2}{1-\text{frac}_\text{s}}\end{aligned} \tag{B-6}$$

B-6 Mode 6

$$j_{6,\xi}(\xi)=\frac{10^5\alpha g_{rl}(sx_2-2ku_\text{fb}(\xi))}{s} \tag{B-7}$$

B-7 Mode 7

$$\begin{aligned}\rho(\xi) = &\frac{\omega_m^2\left(-\frac{0.546x_4^2}{\omega_m^2}-\frac{0.605x_4}{\omega_m}+1.93\right)}{\left(\frac{J_3}{g_{rl}^2}+J_1\right)\left(11.8\ +e^{\frac{5.72x_4}{\omega_m}-3.77}\right)^2}-\frac{b_{1v}x_4}{\frac{J_3}{g_{rl}^2}+J_1}-\frac{b_{3v}x_4}{g_{rl}^2J_1+J_3}-\frac{T_{1c}}{\frac{J_3}{g_{rl}^2}+J_1}-\frac{g_{rl}T_{3c}}{g_{rl}^2J_1+J_3}\\
j_{7,\xi}(\xi) = &\frac{J_3}{g_{rl}}\left(\frac{\omega_m^2\left(-\frac{1.092\rho(\xi)x_4}{\omega_m^2}-\frac{0.605\rho(\xi)}{\omega_m}\right)}{\left(\frac{J_3}{g_{rl}^2}+J_1\right)\left(11.8\ +e^{\frac{5.72x_4}{\omega_m}-3.77}\right)^2}\right)\\&-\frac{J_3}{g_{rl}}\left(\frac{11.44\rho(\xi)\omega_me^{\frac{5.72x_4}{\omega_m}-3.77}\left(-\frac{0.546x_4^2}{\omega_m^2}-\frac{0.605x_4}{\omega_m}+1.93\right)}{\left(\frac{J_3}{g_{rl}^2}+J_1\right)\left(11.8+e^{\frac{5.72x_4}{\omega_m}-3.77}\right)^3}\right)\\&-\frac{J_3}{g_{rl}}\left(\frac{b_{1v}\rho(\xi)}{\frac{J_3}{g_{rl}^2}+J_1}-\frac{b_{3v}\rho(\xi)}{g_{rl}^2J_1+J_3}\right)+\frac{b_{3v}\rho(\xi)}{g_{rl}}\end{aligned} \tag{B-8}$$

Appendix C

Flow* model file

```
hybrid reachability
{
 state var x1,x2,x3,x4,x5,x6,t,co,fracstart

 par
 {
   uff=0.1    ufb=0.063  k=72.1    kt=1.797e-6 s=0.05    a=0.0198 #co
       =-0.884#-1.0608#-0.7072
    a2=-1e-3  b2=2       c2=0.002    d2=5.2e-4 wm=125.6637
   b1v=0.3764 b3v=0.15  J1=0.1    J3=2.706    T1c=1e-4  T3c=94.5
   grL=5.036  alpha=9.6900e-05    posContact=0.005      #fracstart=0.65
   wth=1e-3   treg=0.1  beta=10   const=0.5820 gamma= -1e-3 eps= 1e-6
   invwth = 1000 invJ3 = 0.36955 invposContact = 200 invgrL=0.1985703
   trans2 = 1.8085718515e3 trans5 = 2.7824e3#2.782418233075385e3
 }

 setting
 {
  adaptive steps { min 0.000001 , max 0.01 }
  time  1.5
  remainder estimation 1e-5
  QR precondition
  matlab octagon t,x5
  fixed orders 4
  cutoff 1e-6
  precision 53
  output clutchbase
  max jumps 100
  print on
 }

 modes
 {
```

```
33  mode1
34  {
35   nonpoly ode
36   {
37    x1' = x2 - 2*k/s*uff
38    x2' = -6/s^2*x1 - 4/s*(x2 - 2*k/s*uff) - 6*a*k/s^2 + 6*k/s^2*uff
39    x3' = b2/c2*x1 + a2/c2*(x2 - 2*k/s*uff) - d2/c2*x3 - co*b2/c2
40    x4' = -b1v/J1*x4 - T1c/J1 - gamma/(J1*posContact)*(x5 - 1/grL*x4) +
          wm^2*(-0.546*(x4/wm)^2 - 0.605*(x4/wm) + 1.93)/(J1*(exp(5.72*x4/wm
           - 3.77) + 11.8)^2)
41    x5' = 1/wth*(-b3v/J3*wth - T3c/(J3))*x5 + grL*gamma/(J3*posContact)*(
          x5 - 1/grL*x4)
42    x6' = 0
43    t'  = 1
44    fracstart'=0
45    co' = 0
46    }
47   inv
48   {
49    x3 <= 0
50   }
51  }
52  mode2
53  {
54   nonpoly ode
55   {
56     x1' = x2 - 2*k/s*uff
57     x2' = -6/s^2*x1 - 4/s*(x2 - 2*k/s*uff) - 6*a*k/s^2 + 6*k/s^2*uff
58     x3' = b2/c2*x1 + a2/c2*(x2 - 2*k/s*uff) - d2/c2*x3 - co*b2/c2
59     x4' = -b1v/J1*x4 - T1c/J1 - gamma/(J1*(posContact- kt*x3))*(x5 - 1/
          grL*x4) + wm^2*(-0.546*(x4/wm)^2 - 0.605*(x4/wm) + 1.93)/(J1*(exp
          (5.72*x4/wm - 3.77) + 11.8)^2)
60     x5' = 1/wth*(-b3v/J3*wth - T3c/(J3))*x5 + grL*gamma/(J3*(posContact-
           kt*x3))*(x5 - 1/grL*x4)
61     x6' = 0
62     t'  = 1
63     fracstart'=0
64     co' = 0
65    }
66   inv
67   {
68    x3 <= trans2
69   }
70  }
71  mode3
72  {
73   nonpoly ode
74   {
75     x1' = x2 - 2*k/s*uff
76     x2' = -6/s^2*x1 - 4/s*(x2 - 2*k/s*uff) - 6*a*k/s^2 + 6*k/s^2*uff
77     x3' = b2/c2*x1 + a2/c2*(x2 - 2*k/s*uff) - d2/c2*x3 - co*b2/c2
```

```
78      x4' = -b1v/J1*x4 - T1c/J1 - (1 -x6)*gamma/(J1*(posContact- kt*x3))*(
            x5 - 1/grL*x4) -x6*alpha*(((kt*x3/posContact-fracstart)/(1-
            fracstart))*1e5*x1+1e5)/J1 + wm^2*(-0.546*(x4/wm)^2 - 0.605*(x4/
            wm) + 1.93)/(J1*(exp(5.72*x4/wm - 3.77) + 11.8)^2)
79      x5' = 1/wth*(-b3v/J3*wth - T3c/(J3))*x5 + (1 -x6)*grL*gamma/(J3*(
            posContact- kt*x3))*(x5 - 1/grL*x4) + x6*grL*alpha*(((kt*x3/
            posContact-fracstart)/(1-fracstart))*1e5*x1+1e5)/J3
80      x6' = beta*(x6 + const)
81      t' = 1
82      fracstart'=0
83      co' = 0
84     }
85    inv
86    {
87     x5 <= wth
88    }
89   }
90   mode4
91   {
92    nonpoly ode
93    {
94      x1' = x2 - 2*k/s*uff
95      x2' = -6/s^2*x1 - 4/s*(x2 - 2*k/s*uff) - 6*a*k/s^2 + 6*k/s^2*uff
96      x3' = b2/c2*x1 + a2/c2*(x2 - 2*k/s*uff) - d2/c2*x3 - co*b2/c2
97      x4' = -b1v/J1*x4 - T1c/J1 - (1 -x6)*gamma/(J1*(posContact- kt*x3))*(
            x5 - 1/grL*x4) -x6*alpha*(((kt*x3/posContact-fracstart)/(1-
            fracstart))*1e5*x1+1e5)/J1 + wm^2*(-0.546*(x4/wm)^2 - 0.605*(x4/
            wm) + 1.93)/(J1*(exp(5.72*x4/wm - 3.77) + 11.8)^2)
98      x5' = -b3v/J3*x5-T3c/J3+(1 -x6)*grL*gamma/(J3*(posContact- kt*x3))*(
            x5 - 1/grL*x4) +x6*grL*alpha*(((kt*x3/posContact-fracstart)/(1-
            fracstart))*1e5*x1+1e5)/J3
99      x6' = beta*(x6 + const)
100     t' = 1
101     fracstart'=0
102     co' = 0
103    }
104   inv
105   {
106    x6 <= 1
107   }
108  }
109  mode5
110  {
111    nonpoly ode
112    {
113      x1' = x2 - 2*k/s*uff
114      x2' = -6/s^2*x1 - 4/s*(x2 - 2*k/s*uff) - 6*a*k/s^2 + 6*k/s^2*uff
115      x3' = b2/c2*x1 + a2/c2*(x2 - 2*k/s*uff) - d2/c2*x3 - co*b2/c2
116      x4' = -b1v/J1*x4 - T1c/J1 -alpha*(((kt*x3/posContact-fracstart)/(1-
            fracstart))*1e5*x1+1e5)/J1 + wm^2*(-0.546*(x4/wm)^2 - 0.605*(x4/
            wm) + 1.93)/(J1*(exp(5.72*x4/wm - 3.77) + 11.8)^2)
117     x5' = -b3v/J3*x5-T3c/J3 +grL*alpha*(((kt*x3/posContact-fracstart)
            /(1-fracstart))*1e5*x1+1e5)/J3
```

```
118       x6' = 0
119       t'  = 1
120       fracstart'=0
121       co' = 0
122     }
123     inv
124     {
125       x3<=trans5
126     }
127   }
128   mode6a
129   {
130     nonpoly ode
131     {
132     x1' = x2 - 2*k/s*ufb
133     x2' = -6/s^2*x1 - 4/s*(x2 - 2*k/s*ufb) - 6*a*k/s^2 + 6*k/s^2*ufb
134     x3' = b2/c2*x1 + a2/c2*(x2 - 2*k/s*ufb) - d2/c2*x3 - co*b2/c2
135     x4' = -b1v/J1*x4 - T1c/J1  -alpha*(((posContact/posContact-fracstart)
          /(1-fracstart))*1e5*x1+1e5)/J1 + wm^2*(-0.546*(x4/wm)^2 - 0.605*(
          x4/wm) + 1.93)/(J1*(exp(5.72*x4/wm - 3.77) + 11.8)^2)
136     x5' = -b3v/J3*x5-T3c/J3 +grL*alpha*(((posContact/posContact-fracstart
          )/(1-fracstart))*1e5*x1+1e5)/J3
137     x6' = 0
138     t'  = 1
139     fracstart'=0
140     co' = 0
141     }
142     inv
143     {
144     x5- 0.1986*x4<=-0.05
145     }
146   }
147   mode6b
148   {
149     nonpoly ode
150     {
151     x1' = x2 - 2*k/s*ufb
152     x2' = -6/s^2*x1 - 4/s*(x2 - 2*k/s*ufb) - 6*a*k/s^2 + 6*k/s^2*ufb
153     x3' = b2/c2*x1 + a2/c2*(x2 - 2*k/s*ufb) - d2/c2*x3 - co*b2/c2
154     x4' = -b1v/J1*x4 - T1c/J1  -alpha*(((posContact/posContact-fracstart)
          /(1-fracstart))*1e5*x1+1e5)/J1 + wm^2*(-0.546*(x4/wm)^2 - 0.605*(
          x4/wm) + 1.93)/(J1*(exp(5.72*x4/wm - 3.77) + 11.8)^2)
155     x5' = -b3v/J3*x5-T3c/J3 +grL*alpha*(((posContact/posContact-fracstart
          )/(1-fracstart))*1e5*x1+1e5)/J3
156     x6' = 0
157     t'  = 1
158     fracstart'=0
159     co' = 0
160     }
161     inv
162     {
163     x5- 0.1986*x4<=-0.0005
164     }
```

```
  }
  mode7
  {
    nonpoly ode
    {
    x1' = x2 - 2*k/s*ufb
    x2' = -6/s^2*x1 - 4/s*(x2 - 2*k/s*ufb) - 6*a*k/s^2 + 6*k/s^2*ufb
    x3' = b2/c2*x1 + a2/c2*(x2 - 2*k/s*ufb) - d2/c2*x3 - co*b2/c2
    x4' = -b1v/(J1+J3/grL^2)*x4 -T1c/(J1+J3/grL^2) + wm^2*(-0.546*(x4/wm)
        ^2 - 0.605*(x4/wm) + 1.93)/((J1+J3/grL^2)*(exp(5.72*x4/wm - 3.77)
        + 11.8)^2) -b3v/(J1*grL^2+J3)*x4 -T3c*grL/((J1*grL^2+J3))
    x5' = 1/grL*(-b1v/(J1+J3/grL^2)*x4 -T1c/(J1+J3/grL^2) + wm
        ^2*(-0.546*(x4/wm)^2 - 0.605*(x4/wm) + 1.93)/((J1+J3/grL^2)*(exp
        (5.72*x4/wm - 3.77) + 11.8)^2) -b3v/(J1*grL^2+J3)*x4 -T3c*grL/((J1
        *grL^2+J3)))
    x6' = 0
    t'  = 1
    fracstart'=0
    co' = 0
    }
    inv
    {
      #x5 - 0.1986*x4>=-0.0005
    }
  }
}

jumps
{
   mode1 -> mode2
   guard { x3 >= 0}
   reset { }
   parallelotope aggregation { }
   mode2 -> mode3
   guard { x3 >= trans2 }
   reset { }
   parallelotope aggregation { }
   mode3 -> mode4
   guard { x5 >= wth }
   reset { }
   parallelotope aggregation { }
   mode4 -> mode5
   guard { x6>=1 }
   reset { }
   parallelotope aggregation { }
   mode5 -> mode6a
   guard { x3 >=trans5 }
   reset { }
   parallelotope aggregation { }
   mode6a -> mode6b
   guard { x5 - 0.1986*x4>=-0.04999999 }
   reset { }
   parallelotope aggregation { }
```

```
213     mode6b -> mode7
214     guard { x5 - 0.1986*x4>=-0.0005 }
215     reset { }
216     parallelotope aggregation { }
217   }
218
219   init
220   {
221    mode1
222    {
223    x1 in [-1.4272,-1.4271]
224    x2 in [288.400000224,288.400000225]
225    x3 in [-1.9306e3,-1.9e3]
226    x4 in [120.1,120.1139]
227    x5 in [8.5e-7,8.5069e-7]
228    x6 in [0,0]
229    t in [0,0]
230    fracstart in [0.64,0.66]
231    co in [-1.0608,-0.7072]
232    }
233   }
234 }
```

Bibliography

[1] T. Ouyang, S. Li, G. Huang, F. Zhou, and N. Chen, "Mathematical modeling and performance prediction of a clutch actuator for heavy-duty automatic transmission vehicles," *Mechanism and Machine Theory*, vol. 136, pp. 190 – 205, 2019.

[2] A. Dutta, Y. Zhong, B. Depraetere, K. V. Vaerenbergh, C. Ionescu, B. Wyns, G. Pinte, A. Nowe, J. Swevers, and R. D. Keyser, "Model-based and model-free learning strategies for wet clutch control," *Mechatronics*, vol. 24, no. 8, pp. 1008 – 1020, 2014.

[3] J. Wong, *Theory of ground vehicles*. John Wiley & Sons, 3rd ed., 2001.

[4] V. K. Dabhi and S. Chaudhary, "Empirical modeling using genetic programming: a survey of issues and approaches," *Natural Computing*, vol. 14, pp. 303–330, Jun 2015.

[5] B. Depraetere, G. Pinte, W. Symens, and J. Swevers, "A two-level iterative learning control scheme for the engagement of wet clutches," *Mechatronics*, vol. 21, no. 3, pp. 501 – 508, 2011.

[6] A. Dutta, B. Depraetere, C.-M. Ionescu, G. Pinte, J. Swevers, and R. De Keyser, "Comparison of two-level nmpc and ilc strategies for wet-clutch control," *CONTROL ENGINEERING PRACTICE*, vol. 22, pp. 114–124, 2014.

[7] B. Depraetere, G. Pinte, and J. Swevers, "A reference free iterative learning strategy for wet clutch control," in *Proceedings of the 2011 American Control Conference*, pp. 2442–2447, June 2011.

[8] M. Gagliolo, K. V. Vaerenbergh, A. Rodríguez, A. Nowé, S. Goossens, G. Pinte, and W. Symens, "Policy search reinforcement learning for automatic wet clutch engagement," in *15th International Conference on System Theory, Control and Computing*, pp. 1–6, Oct 2011.

[9] Y. Zhong, B. Wyns, R. De Keyser, G. Pinte, and J. Stoev, "An implementation of genetic-based learning classifier system on a wet clutch system," in *Applied Stochastic Models and Data Analysis Conference, 14th, Proceedings*, pp. 1431–1439, 2011.

[10] P. Tabuada, *Verification and Control of Hybrid Systems: A Symbolic Approach*. Springer Publishing Company, Incorporated, 1st ed., 2009.

[11] M. Z. Romdlony and B. Jayawardhana, "Stabilization with guaranteed safety using control lyapunov–barrier function," *Automatica*, vol. 66, pp. 39 – 47, 2016.

[12] C. Verdier and J. Mazo, M., "Formal controller synthesis via genetic programming," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 7205–7210, 2017. cited By 2.

[13] X. Chen, E. Ábrahám, and S. Sankaranarayanan, "Flow*: An analyzer for non-linear hybrid systems," in *Computer Aided Verification* (N. Sharygina and H. Veith, eds.), (Berlin, Heidelberg), pp. 258–263, Springer Berlin Heidelberg, 2013.

[14] G. Tod, "Wet clutch: Modelling and simulation, problem casted into odes." 2018.

[15] M. Negnevitsky, *Artificial Intelligence: A Guide to Intelligent Systems*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1st ed., 2001.

[16] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: University of Michigan Press, 1975. second edition, 1992.

[17] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA, USA: MIT Press, 1992.

[18] P. Whigham and D. O. C. Science, "Grammatically-based genetic programming," 1995.

[19] C. F. Verdier and M. M. Jr., "Formal synthesis of analytic controllers for sampled-data systems via genetic programming," *CoRR*, vol. abs/1812.02711, 2018.

[20] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE Transactions on Evolutionary Computation*, vol. 6, pp. 182–197, Apr 2002.

[21] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1st ed., 1989.

[22] B. L. Miller and D. E. Goldberg, "Genetic algorithms, tournament selection, and the effects of noise," *Complex Systems*, vol. 9, pp. 193–212, 1995.

[23] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evol. Comput.*, vol. 8, pp. 173–195, June 2000.

[24] R. Poli, W. B. Langdon, and N. F. McPhee, *A field guide to genetic programming*. Published via `http://lulu.com` and freely available at `http://www.gp-field-guide.org.uk`, 2008. (With contributions by J. R. Koza).

[25] M. Črepinšek, S.-h. Liu, and M. Mernik, "Exploration and exploitation in evolutionary algorithms: A survey," *ACM Computing Surveys*, vol. 45, p. Article 35, 06 2013.

[26] E. Asarin, T. Dang, G. Frehse, A. Girard, C. L. Guernic, and O. Maler, "Recent progress in continuous and hybrid reachability analysis," in *2006 IEEE Conference on Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control*, pp. 1582–1587, Oct 2006.

[27] R. Alur, C. Courcoubetis, N. Halbwachs, T. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine, "The algorithmic analysis of hybrid systems," *Theoretical Computer Science*, vol. 138, no. 1, pp. 3 – 34, 1995. Hybrid Systems.

[28] M. Althoff, *Reachability Analysis and its Application to the Safety Assessment of Autonomous Cars*. PhD thesis, Technische Universität München, 7 2010.

[29] O. Stursberg and B. H. Krogh, "Efficient representation and computation of reachable sets for hybrid systems," in *Proceedings of the 6th International Conference on Hybrid Systems: Computation and Control*, HSCC'03, (Berlin, Heidelberg), pp. 482–497, Springer-Verlag, 2003.

[30] M. Althoff, "An introduction to cora 2015," in *ARCH14-15. 1st and 2nd International Workshop on Applied veRification for Continuous and Hybrid Systems* (G. Frehse and M. Althoff, eds.), vol. 34 of *EPiC Series in Computing*, pp. 120–151, EasyChair, 2015.

[31] D. Bresolin, L. Geretti, T. Villa, and P. Collins, *An Introduction to the Verification of Hybrid Systems Using Ariadne*, pp. 339–346. Cham: Springer International Publishing, 2015.

[32] X. Chen, "Reachability analysis of non-linear hybrid systems using taylor models," 2015.

[33] MATLAB, *version 9.3.0.713579 (R2017b)*. Natick, Massachusetts: The MathWorks Inc., 2017.

[34] W. R. Inc., "Mathematica, Version 11.2." Champaign, IL, 2017.

[35] rsmenon and szhorvat, "Matlink." http://matlink.org/.