Modelling and coordinated control of advanced automotive transmissions

Kusters, J.R.P.

Award date:
2005
Modelling and Coordinated Control of Advanced Automotive Transmissions

J.R.P. Kusters

DCT 2005-58

Master Thesis

Supervisor: Prof. dr. ir. M. Steinbuch
Coach: Dr. ir. A.F.A. Serrarens

EINDHOVEN UNIVERSITY OF TECHNOLOGY
DEPARTMENT OF MECHANICAL ENGINEERING
SECTION CONTROL SYSTEMS TECHNOLOGY
DRIVETRAIN INNOVATIONS BV

Eindhoven, May 2005
Preface

This Master Thesis is the result of my final year project performed at the department of Mechanical Engineering at the Eindhoven University of Technology (TU(e)). The project was defined by DriveTrain Innovations (DTI) in Eindhoven. The aim of the project was to bring both the modelling and coordinated control of automotive transmissions to a higher level. The quest for more sophisticated methods for modelling and coordinated control stems from recent developments towards increasing complexity of automotive transmissions. I joined DTI in the last few months of my project to implement the new methodology on a transmission concept of theirs, called Brake-Impulse Shift Transmission (B-IST). The two different parts—modelling and coordinated control—of this thesis can be read separately. Although the original assignment only concerned coordinated control, I noticed shortcomings in the currently used modelling methods and made this part of my project.

First of all I would like to thank my supervisor Alex Serrarens and my other colleagues at DTI for their support during my research. Furthermore I would like to express my gratitude to all the people who have put an effort into supporting me in my research, in particular Maurice Heemels and Nathan van de Wouw but also Remco Leine, Friedrich Pfeiffer, Graham Goodwin and Alberto Bemporad for prompt answers in literature questions and other requests. Finally I would like to thank my friends and family for their continuous interest and support, especially my girlfriend Dorine, my sister Janine and my parents John and Jacqueline.
Abstract

Due to the desire to continually increase the performance, comfort and fuel economy of road vehicles, a trend towards an increasing complexity of automotive transmissions is emerging. This increased complexity of the transmissions expresses itself, among other things, in the number of frictional components –e.g. clutches– present in the transmission. These frictional components may introduce discontinuous dynamical behavior.

In the first part of this thesis different methods for modelling and simulating discontinuous systems, in particular automotive transmissions, have been conveyed. The Switch Model is the method currently used within the automotive engineering section at the Eindhoven University of Technology. This method works well, but becomes impracticable when the number of frictional components grows.

The first group of more sophisticated modelling methods are the event-driven integration methods. Here the model uses a standard ODE-integrator in the smooth phases whereas a separate algorithm determines the next mode when a transition is detected. The system description is then rewritten into a Linear Complementarity Problem.

The other group are the time-stepping methods, in which it is necessary to formulate the dynamics as an equality of measures. The time-stepping methods do not use the notion of events like stick/slip transitions, but determine the state of the model at the end of each time-step (fixed) by computing the overall integral of the acting forces over that time-step. Finally the new Matlab/Simulink package SimDriveline is introduced, in which models can be made in a ‘plug and play’-manner.

The first part is concluded with a test case and on the basis of the results the methods are compared with each other. Both the Switch Model and the event-driven integration methods result in fast and accurate simulation results, but for the event-driven integration methods simulation times would definitely rise when a test-case with more stick/slip transitions would be taken. The time-stepping methods are less accurate and the computation time they win because no event detection is necessary, is not apparent in the simulations. Firstly because only a few events happen during the simulation (inherently with a high-level, automotive transmission model) and secondly because the method is a fixed time-step method.

In the second part of this thesis the coordinated control of the advanced transmissions is treated. Research is done after concepts for optimal, constrained control. Firstly, an algorithm is designed which uses Dynamic Programming (DP) for deriving optimal control. The state space is discretized and at each time-step the control input with the minimized cost-to-go is fed into the system. This algorithm has been tested on clutch engagement simulations and works satisfactory. Explicit constraints can be set and performance can be tuned with the weight parameters in the cost function. But when the dimensions of the state space and the input vector grows too large, the computational time becomes exponentially high. Secondly a DP strategy is introduced which results in an explicit control law, but with the drawback that constraint violation has to be avoided by adding extra weights into the cost function, which of course decreases performance.

Finally a Model Predictive Control (MPC) approach is presented, where an MPC controller is tuned for optimal performance and then is made explicit by a multi-parametric Quadratic Programming solver. This results in a piecewise affine, continuous, state-feedback control law, which can be implemented on a ‘fast’ mechanical system like an automotive transmission. This method
has been implemented on clutch engagement and on a gear shift operation of the advanced transmission concept ‘Brake-Impulse Shift Transmission’, developed by DriveTrain Innovations (DTI).
Door de wens om voortdurend de prestaties, het comfort en de brandstofbesparing van voertuigen te verhogen, is er een trend ontstaan richting complexere transmissies in deze voertuigen. De toegenomen complexiteit uit zich onder andere in het aantal wrijvingselementen, zoals koppelingen, in de aandrijfsystemen. Deze wrijvingselementen kunnen discontinuïteiten in het dynamisch gedrag veroorzaken.

In het eerste deel van dit verslag worden verschillende methodes voor het modelleren en simuleren van discontinue systemen, in het bijzonder automotive aandrijfsystemen, onderzocht. Het Switch Model wordt op het moment gebruikt binnen de automotive sectie op de Technische Universiteit Eindhoven (TU/e). Deze methode werkt goed, maar wordt snel onpraktisch als het aantal wrijvingselementen toeneemt.

De eerste groep van ‘slimme’ modelleer methodes zijn de ‘event-driven integration’ methodes, die het model met een standaard ODE-integrator in de gladde fases integreren en een apart algoritme gebruiken om de volgende modus te bepalen –bijv. door het herschrijven van het contact probleem als een Lineair Complementariteit Probleem – als een transitie gedetecteerd wordt.

De andere groep zijn de ‘time-stepping’ methodes, waarin het nodig is de bewegingsvergelijkingen op impuls-momentum niveau te schrijven in plaats van kracht-acceleratie. Deze methodes zoeken tijdens het simuleren niet naar transities en het bijbehorende tijdstip, maar ‘stappen’ hier als het ware over heen en berekenen de toestand van het systeem aan het einde van elke tijdstap (vast) door de algehele integraal van de aangrijpende krachten in dat tijdsinterval te berekenen. Uiteindelijk wordt het nieuwe Matlab/Simulink pakket SimDriveline geïntroduceerd, waarmee modellen gebouwd kunnen worden op een ‘plug and play’ manier.

Het eerste gedeelte eindigt met een testcase en aan de hand van die resultaten worden de verschillende methodes met elkaar vergeleken. Het Switch Model en de ‘event-driven integration’ methode resulteren in nauwkeurige en snelle simulaties, maar de simulatietijd van de ‘event-driven integration’ methode zou zeker toenemen als er meer stick/slip transities zouden plaatsvinden tijdens de simulatie, omdat steeds het precieze tijdstip van de gebeurtenis bepaald moet worden. De ‘time-stepping’ methodes zijn minder nauwkeurig en de rekentijd die gewonnen zou worden door het ‘overstappen’ van transities, komt niet naar voren in de simulaties. In de eerste plaats omdat er maar een paar transities plaatsvinden op het hele traject en ten tweede omdat het een method is met een vaste tijdstap.

Het tweede deel van dit verslag heeft als onderwerp de coördinerende regeling van geavanceerde aandrijfsystemen. Er is onderzoek gedaan naar concepten voor optimale regelstrategieën voor systemen met beperkingen op de toestands- en ingangsruiunte. Allereerst is er een algoritme ontworpen met behulp van Dynamic Programming (DP) om tot een optimale regelstrategie te komen. De toestandsruimte wordt hierin gediscreetiseerd en op elk tijdstip wordt die regeling aan het systeem opgelegd die de minimale nog resterende kosten tot het einde heeft. Dit algoritme is getest op koppelings-synchronisatie en werkt goed; limieten voor toestands- en ingangsruiunte kunnen opgegeven worden. En de prestaties kunnen afgesteld worden met behulp van de weegfactoren in de kosten-functie. Echter wanneer de dimensionaliteit van de toestands en/of ingangsruiunte groter wordt, wordt de rekentijd buitenproportioneel. Daarnaast wordt er een DP strategie behandeld die resulteert in een expliciete regelwij, maar deze heeft als nadeel dat overschrijding van limieten van toestanden en ingangen voorkomen moet worden door extra weeg-termen te introduceren in
de kosten-functie, wat natuurlijk de prestaties verlaagt.
Uiteindelijk wordt er een Model Predictive Control (MPC) strategie geïntroduceerd, alwaar eerst
een MPC regelaar afgesteld wordt voor optimale prestaties om vervolgens in expliciete vorm te
worden afgeleid met behulp van een ‘multi-parametric Quadratic Programming’-solver. Dit re-
sulteert in een affiene, toestands-terugkoppeling regelwet, die continu is maar wel verschillende
definities heeft voor verschillende gebieden in de toestandsruimte. Hierdoor is voor de implemen-
tatie niets anders nodig dan het opzoeken van de op dat moment geldende optimale regelwet. Met
dezelfde methode is een regelaar ontworpen voor koppelings-synchronisatie en voor een schakelac-
tie van de ‘Brake-Impulse Shift Transmission’, een geavanceerd aandrijf-concept van DriveTrain
Innovations.
Contents

Preface 1

Abstract 3

Samenvatting 5

Introduction 11

Problem Description 11
Structure Part I 11
Structure Part II 12
Literature survey 12

I Modelling of Advanced Automotive Transmissions 13

1 Preliminaries 15

1.1 Dry Friction 15
1.2 Frictional Components in Automotive Transmissions 16
1.3 Smoothing Method 16
1.4 Dual-Clutch System 17

2 Switch Model 19

3 Event-driven Integration Method 23

3.1 Mode selection with an LCP 23
3.2 Complementarity Properties of Friction 24
3.3 Decompositions of contact law 25

3.3.1 Decomposition by Pleffer and Glocker 25
3.3.2 Decomposition by Leine and Nijmeijer 29

3.4 LCP Solution Algorithms 31

3.4.1 Enumerative Method 32
3.4.2 Lemke’s Method 32
3.5 Implementation 33
## CONTENTS

4 Time-stepping Method 35  
4.1 Discretization and Moreau's Algorithm 35  
4.2 Time-stepping with LCP 36  
4.3 Time-stepping with Augmented Lagrangian Approach 36  

5 SimDriveline 39  
5.1 Controllable Friction Clutch 39  

6 Evaluation; The Dual-clutch Case 41  
6.1 Dual-clutch Simulations 41  
6.2 Results 42  

7 Conclusions and Recommendations 45  
7.1 Conclusions 45  
7.2 Recommendations 46  

II Coordinated Control of Advanced Automotive Transmissions 47  

8 Preliminaries 49  
8.1 Automotive Drivetrain Control 49  
8.2 Optimal, Constrained Control 50  
8.2.1 Optimal Control 50  
8.2.2 Constrained Control 50  

9 Dynamic Programming 53  
9.1 Optimal Clutch Engagement with Numerical DP 54  
9.1.1 Results 56  
9.2 Analytical Dynamic Programming 56  

10 Explicit Model Predictive Control 61  
10.1 Introduction 61  
10.2 Hybrid Toolbox 62  
10.3 HYSDEL 62  
10.4 Dry clutch engagement 63  
10.5 Coordinated Control of the B-IST 63  
10.6 Implementation Issues 69  
10.6.1 General considerations 69  
10.6.2 Explicit MPC control law for linear clutch model 71  
10.6.3 Considerations for drivetrain control of the B-IST 73  

11 Conclusions and Recommendations 75  
11.1 Conclusions 75  
11.2 Recommendations 76
### III Bibliography and Appendices

#### Bibliography

#### A List of Abbreviations and Symbols

- **A.1 Abbreviations**
- **A.2 Symbols**

#### B S-function: Switch Model

#### C Analysis of initial decomposition contact law

#### D S-function: Event-Driven Integration Method with LCP

#### E Lemke’s Algorithm

#### F S-function: Time-Stepping Method with LCP

#### G S-function: Time-Stepping Method with Augmented Lagrangian

#### H Dual-clutch Simulations

#### I Hybrid Control Clutch

- **I.1 HYSDEL model**
- **I.2 Hybrid Controller**

#### J Planetary Gear Set

#### K Hybrid Control B-IST

- **K.1 HYSDEL model**
  - **K.1.1 Part 1**
  - **K.1.2 Part 2**
- **K.2 Hybrid Controller B-IST 2-3 Upshift**
  - **K.2.1 Case Constant Driveshaft Torque**
  - **K.2.2 Case Decreasing Driveshaft Torque**
- **K.3 Hybrid Controller B-IST 1-2 Upshift**
  - **K.3.1 Case Decreasing Driveshaft Torque**

#### L Explicit Controller for Linear Clutch Model

- **L.1 M-file**
- **L.2 Explicit Form**
Introduction

Problem Description

Due to the desire to continually increase the performance, comfort and fuel economy of road vehicles, a trend towards an increasing complexity of automotive transmissions is emerging. As engineers come up with new, more complex concepts for transmissions, current modelling techniques are not sophisticated and efficient enough to lead to satisfactory results. This is mainly due to the discontinuous nature of the frictional components in the transmission. Systems exposing a mixed continuous and discrete nature are also called hybrid systems [Heemels, 1999]. In the first part of this thesis different modelling techniques for discontinuous systems are presented and compared with one another in user-friendliness, accurateness and computation time, when used for an automotive application.

Furthermore, the coordinated control –i.e. controlling of the engine and transmission in a coordinated manner to optimize driveability and fuel economy [Serrarens, 2001]– also becomes more complicated and less straightforward with advanced automotive transmissions. Current methodologies of implementing control laws based on heuristics will not be able to obtain the most optimal performance of the advanced transmissions. The topic of the second part of this thesis is the design of a coordinated controller for an automotive transmission. The second goal of this work is to research different methodologies, compare them with one another and from that choose the best method to design a coordinated controller for the advanced transmission concept of DriveTrain Innovations (DTI), the Brake- Impulse Shift Transmission (B-IST).

Structure Part I

In Chapter 1 the phenomenon of dry friction is introduced and the modelling difficulties that arise from frictional components in an automotive transmission. At the end the smoothing method is mentioned which does not handle the discontinuities but simply replaces them with continuous dynamics. Finally a typical automotive system, called here the ‘Dual clutch’, is introduced which will be used as a test case for the different methodologies. In Chapter 2 the Switch Model is introduced which is an improved version of the earlier Karnopp Model. The model for a simple clutch is derived to demonstrate this method. Chapter 3 is dedicated to the event-driven integration method where a Linear Complementarity Problem (LCP) is used for the mode selection. Two different strategies are presented for the decomposition of the contact problem. A completely different approach is the time-stepping method, treated in Chapter 4 where the usual equations of motion, relating acceleration to forces, is replaced by an equality of measures, relating change in momentum with the resultant impulse. The contact problem is described either with an LCP or by means of an augmented Lagrangian approach. Chapter 5 is a short background on the recently released SimDriveline package, with the emphasis on how it deals with the discontinuities caused by stick/slip transitions. Finally in Chapter 6 the results of the ‘Dual clutch’-test case are available and in Chapter 7 the conclusions and recommendations are presented.
Structure Part II

In Chapter 8, the various approaches for optimal, constrained control are mentioned. Chapter 9 begins with numerical Dynamic Programming, a deterministic, feedforward approach, and an algorithm is presented for optimal clutch engagement with Dynamic Programming. Furthermore analytical Dynamic Programming is introduced which results in closed-form solutions with both feedforward and feedback. Finally Chapter 10 treats a recently developed method which enables obtaining an explicit form of a Model Predictive Controller. This explicit form consists of a piecewise affine, continuous, state-feedback control law. In Chapter 11 the conclusions and recommendations are presented.

Literature survey

The first publications in which the instantaneous dynamic equations of a system of rigid bodies in unilateral contact were formulated as a complementarity problem, were from Löstedt in [Löstedt, 1982] and [Löstedt, 1984]. In [Pfeiffer and Glocker, 1996] a complete description is given on the event-driven integration method with an LCP for determining the contact configuration. Recently in [Leine and Nijmeijer, 2004] this method is optimized and time-stepping methods are treated as is the augmented Lagrange approach for solving the contact problem. Currently Leine works with Glocker at the ETH Zürich, where a lot of research is done on non-smooth mechanics.

Dynamic Programming was introduced in [Bellman, 1957], and has grown very popular in a very broad field of application (e.g. financial, economic, mathematic, engineering, etc.). In [Haj-Fra'j and Pfeiffer, 2001] an analytical, explicit solution is obtained for the Dynamic Programming algorithm and implemented on gear shift operations in automatic transmissions. Also the Model Predictive Control approach in [Bemporad, 2004] leads to an explicit solution of the control law. The basis for this method was accomplished in [Bemporad et al., 2002a]. A lot of work on this topic is done by Bemporad, who is affiliated to both the ETH in Zürich and the University of Siena in Italy. On different places there one works on improvement of this method; Imperial College in London, ETH in Zürich, University of Siena in Italy, University of Trondheim in Norway. In 2003 an Imperial College London spin out company, named ParOS, was founded which is bringing this technology to the market in the automotive, biomedical, industrial and aerospace & defense field (www.parostech.com). Within the Electrical Engineering department of the Eindhoven University of Technology (TU/e) currently a Ph.D. project is done on Model Predictive Control of Hybrid Systems by M.Sc. M. Lazar, supervised by Dr. ir. W.P.M.H. Heemels.
Part I

Modelling of Advanced Automotive Transmissions
Chapter 1

Preliminaries

1.1 Dry Friction

Classical multibody dynamics of the form

\[ M \ddot{q} - D \dot{q} - Kq - f(t) = 0, \]  
(1.1)

are relatively easy to simulate with standard ODE integrators. The constitutive laws for springs and dampers fit perfectly in such a system description. However in an automotive transmission also frictional components play an important role, which will only grow in the future. Unfortunately these frictional components do not behave in such a manner that they can be described by a constitutive law. The force laws that describe these components are set-valued.

The phenomenon that takes place in the frictional components, i.e. dry friction, is defined in tribology as a force that resists relative motion between contacting surfaces of solid bodies. The difference between dry friction and viscous friction is the lack of a stick phase with the pure viscous friction. With dry friction the friction force in the stick phase – zero relative velocity – adjusts itself to make equilibrium with external forces on the bodies – i.e. set-valued, whereas with viscous friction for zero relative velocity there is no friction force. If the friction force in the stick phase exceeds a threshold, called break-away friction force or maximum static friction force, then the bodies will begin to slip over each other. The friction force during slip is often called the dynamic friction force and because viscous friction lacks the stick phase it can be regarded as dynamic friction. In literature different models are available for dry friction, usually Coulomb’s law is taken to represent this behavior,

\[ F_T \in -\mu \text{Sign}(g)F_N, \]  
(1.2)

where \( F_T \) is the resulting tangential friction force, \( g \) the relative velocity, \( F_N \) the normal force between the contacting surfaces and \( \mu \) the friction coefficient. The \( \text{Sign} \) function is defined as,

\[ \text{Sign}(g) = \begin{cases} 
-1, & g < 0, \\
[-1, 1], & g = 0, \\
1, & g > 0.
\end{cases} \]  
(1.3)

The classical Coulomb’s law assumes one value for the friction coefficient for all relative velocities, which is an idealization of the physical reality because usually the static friction coefficient \( \mu_s \) (when \( g = 0 \)) is greater than the dynamic friction coefficient \( \mu_d \). This dependency on the relative velocity, also called Strubeck-effect can cause instability, which give rise to stick-slip vibrations. Tribological experiments have also shown that the friction force during tangential acceleration is higher than during deceleration, yielding a hysteresis phenomenon.
1.2 Frictional Components in Automotive Transmissions

Mechanical systems with set-valued force laws lead to differential inclusions. In this part of the thesis the mathematical formulation of such mechanical multibody systems with unilateral contact and friction –in particular automotive transmissions– will be further elaborated. A good example of a frictional component in an automotive transmission is the well-known dry friction clutch, present in every drivetrain with a Manual Transmission. Synchromeshes are also an example of frictional components and in more advanced transmissions brake clutches are present to decelerate inertias.

![Figure 1.1: Clutch friction characteristic](image)

The tangential friction torque $T_C$ in these frictional components is defined according to the classical Coulomb’s law,

$$ T_C \in -\mu r_m \text{Sign}(\gamma) F_{NC}, $$

with $r_m$ the mean radius of the contact surfaces and $\gamma$ the relative rotational velocity. This relation is represented graphically in Figure 1.1.

1.3 Smoothing Method

A way to circumvent the non-smoothness and set-valuedness of the $\text{Sign}$ function is to use a smoothening approximation as the $\text{arctangent}$ function instead. The advantage of the smoothing method is that standard integration techniques can be directly applied and no additional programming work has to be done. Discrete modes do not really exist anymore, so event detection and mode selection are not necessary. Instantaneous jumps are replaced by (finitely) fast motions, so also the problem of re-initialization disappears. However the main disadvantage of the smoothing method is the fact that an accurate simulation requires the use of very stiff approximate laws, which are numerically expensive to solve. In the following chapters more sophisticated integration methods such as event-driven integration methods and time-stepping methods will be introduced. Also the recently released Simulink-package $\text{SimDriveline}$ will be evaluated with respect to its handling of stick/slip in the frictional components (Dry Friction Clutch and Brake Clutch). Finally all these simulation methods will be evaluated with a case of a simplified model of a modern automotive transmission; the $\text{Dual-clutch}$. 

1.4 Dual-Clutch System

The modelling methods introduced in the following chapters will be tested on the basis of the Dual-clutch system, which is represented in Figure 1.2.

![Figure 1.2: Dual-clutch System](image)

The equations of motion of the Dual-clutch system are,

\[ J_E \dot{\omega}_E = T_E - T_F - T_C, \]
\[ (J_F + \frac{r_F}{r_C} J_C)\dot{\omega}_F = T_F + \frac{r_F}{r_C} T_C - r_F r_D T_K, \]
\[ J_V \dot{\omega}_V = T_K - T_V, \]

with,

\[ T_F \in -\mu_F r_m F_{NF}, \]
\[ T_C \in -\mu_C r_m C_{NC}, \]
\[ T_K = k(r_F r_D \varphi_F - \varphi_V). \]

The engine torque \( T_E \) is considered as an input as well as both clutch normal forces \( F_{NF} \) and \( F_{NC} \). The load torque \( T_V \) is taken linearly dependent on the rotational velocity \( \omega_V \). The results of each method with this test case will be presented as a whole in Chapter 6.
Chapter 2

Switch Model

A remedy for the disadvantages of the smoothing method is found in the model represented by Karnopp [Karnopp, 1985]. The Karnopp Model introduces an interval of low relative velocity, i.e. $|\dot{\gamma}| \leq \eta$, also called 'stick band' which approximates the stick mode. For velocities within this interval, the relative acceleration, $\ddot{\gamma}$ is put to zero. The Karnopp Model leads therefore to a discontinuous system, which is non-stiff within the stick mode. However the constant small offset of the relative velocity causes a drift-off effect for large integration intervals and can cause numerical instability of the ODE integrator. The Switch Model [Leine et al., 1998] is an improved (and generalized) version of the Karnopp Model. It introduces a vector field in the stick band, which can contain transversal intersections through the stick band called transitions, and attractive and repulsive sliding modes. The attractive sliding mode represents continuous stick and causes the state of the system to be pushed to the middle of the stick band – i.e. $\gamma = 0$ in our case, also called the switching boundary. With this the drift-off effect is avoided. Of course the exponential convergence to the middle of the band only takes place while the conditions for continuous stick are met, a case of transition is when the solution passes through the stick band without ever satisfying the conditions of sticking at all. The full mathematical formulation of the Switch Model is available in [Leine and Nijmeijer, 2004], it would lead too far to reproduce it here. Therefore, this method will be introduced on the basis of an example with a simple dry friction clutch, which is schematically represented in Figure 2.1. The equations of motion for this system are,

$$\begin{align*}
J_1 \ddot{\omega}_1 &= T_1 - T_C, \\
J_2 \ddot{\omega}_2 &= T_C - T_2,
\end{align*}$$

Figure 2.1: Simple clutch model

$$\begin{align*}
J_1 \ddot{\omega}_1 &= T_1 - T_C, \\
J_2 \ddot{\omega}_2 &= T_C - T_2,
\end{align*}$$

(2.1)
where $J_1$ and $J_2$ are inertias, $\dot{\omega}_1 = \ddot{\varphi}_1$ and $\dot{\omega}_2 = \ddot{\varphi}_2$ are rotational accelerations, $T_1$ and $T_2$ are external torques and $T_C$ is the friction torque defined according to equation (1.4). The relative velocity is defined as,

$$\gamma = \omega_2 - \omega_1.$$  \hfill (2.2)

Exponential convergence to the switching boundary as long as the stick conditions are met can be established by setting,

$$\dot{\gamma} = -\tau^{-1}\gamma,$$  \hfill (2.3)

which will force $\gamma \to 0$ with the time constant $\tau$. The time constant $\tau$ has dimension $[s]$ and determines how 'fast' the solution will be pushed towards the switching boundary $\gamma = 0$. Now the normal $n(\omega)$ perpendicular to the switching boundary and the generalized accelerations are introduced, respectively,

$$n(\omega) = \nabla \gamma(\omega) = \begin{bmatrix} -1 \\ 1 \end{bmatrix},$$  \hfill (2.4)

With the definition of $n$ and the vector field in the sliding mode,

$$\dot{x}(t) = \alpha f_+ + (1 - \alpha)f_-,$$  \hfill (2.5)

the following relation is obtained,

$$\dot{\gamma}(\omega) = \frac{\partial \gamma_x}{\partial x} \frac{\partial x}{\partial t}$$  \hfill (2.6)

From (2.3) and (2.6) the variable $\alpha$ can be determined,

$$\alpha = \frac{n^T f_- + \tau^{-1}\gamma}{n^T (f_- - f_+)}.$$  \hfill (2.7)

First the system outside the stick band will be defined,

if $\gamma < -\eta$,

then $$\dot{x} = \begin{bmatrix} \frac{T_1 - \mu_r m F_{NC}}{J_1} \\ \frac{\mu_r m F_{NC} - T_2}{J_2} \end{bmatrix} = f_-;$$  \hfill (2.8)

if $\gamma > \eta$,

then $$\dot{x} = \begin{bmatrix} \frac{T_1 + \mu_r m F_{NC}}{J_1} \\ \frac{\mu_r m F_{NC} - T_2}{J_2} \end{bmatrix} = f_+.$$

Now the vector field for the system inside the stick band is defined. Under the obvious conditions $T_1 \geq 0$, $T_2 \geq 0$, $J_1 \geq 0$, $J_2 \geq 0$ and $F_{NC} \geq 0$, it can be shown that there is only one transition mode and the attractive sliding mode possible in the stick band for the simple clutch,

if $n^T f_- > 0$ $\land$ $n^T f_+ > 0$, transition

then $\dot{x} = f_+$,

if $n^T f_- > 0$ $\land$ $n^T f_+ < 0$, attractive sliding mode

then $\alpha = \frac{n^T f_- + \tau^{-1}\gamma}{n^T (f_- - f_+)}$,

$$\dot{x} = \alpha f_+ + (1 - \alpha)f_-.$$  \hfill (2.9)
The thickness parameter $\eta$ should be chosen small enough to have no qualitative influence on the solution. The Switch Model maintains the continuity of the state vector and yields a set of non-stiff ordinary differential equations. A disadvantage of the Switch Model is the exponentially increasing complexity of the logical structure with increasing number of switching boundaries. The Switch Model is therefore not suitable for mechanical systems with many frictional contacts because the combinatorial contact problem of all the different modes for the different contacts, becomes too large.

The Matlab/Simulink S-function where the Switch Model of the Dual-clutch system is programmed is available in Appendix [3].
Chapter 3

Event-driven Integration Method

Event-driven methods are based on considering the simulation interval as a union of disjoint subintervals on which the mode (active constraint set) remains unchanged. On each of these subintervals the motion of the system is smooth and can be computed by a standard ODE/DAE-integrator. As integration proceeds in such a subinterval, one has to monitor certain indicators to determine when the subinterval ends, i.e. event detection. At this event time a mode transition can occur, which means that one has to determine what the new mode will be on the next subinterval, i.e. mode selection. If the state at the event time is not consistent with the selected mode, a jump is necessary, i.e. re-initialization. For the mode selection different approaches are available, the complementary nature of unilateral contacts can be used to set up an LCP for the mode selection, which will be treated in this chapter.

The event-driven integration method integrates the system until an event occurs, calculates the next mode and proceeds integration. It therefore clearly expresses the hybrid nature of systems with friction.

3.1 Mode selection with an LCP

The methods used in this thesis to formulate an LCP to determine the next mode at a switching boundary, were taken from [Pfeiffer and Glocker, 1996] and [Leine and Nijmeijer, 2004]. In [Haj-Fraj and Pfeiffer, 1999] this method is already used in the modelling of an automatic transmission. In this method it is stated that all contact dynamical problems possess complementarity properties; for any unilateral contact either the relative kinematics is zero and some constraint forces are not zero or vice versa. The scalar product of magnitudes representing relative kinematics and constraint forces is always zero. Introducing these considerations into the equations of motion and into the active set of constraint equations allows a reduction of these equations into a standard complementarity problem.

Recall from (1.1) again, the equations of motion in classical dynamics,

\[ M\ddot{q} - h(t, q, \dot{q}) = 0, \]

where \( M \) is the mass matrix and \( h(t, q, \dot{q}) = D\dot{q} + Kq + f(t) \), with \( D \) the viscous damping matrix, \( K \) the stiffness matrix and \( f(t) \) is the column vector with externally applied forces. But here still the unilateral constraints have to be taken into account. First the following index sets are introduced,

\[ I_C = \{1, 2, \ldots, n\}, \]
\[ I_S = \{j \in I_c \mid \gamma_j \neq 0\}, \]
\[ I_T = \{i \in I_c \mid \gamma_i = 0\}. \]
The set \( I_C \) consists of the indices of all \( n \) contact elements. The elements of the set \( I_S \) are all elements that are in sliding mode. The elements of the set \( I_T \) are all elements that are potentially sticking. The constraint vectors \( w_k \) for each frictional element \( k \in I_C \) assigns its friction force/torque to the appropriate generalized coordinates and is defined as,

\[
w_k = \left( \frac{\delta \gamma_k}{\delta \mathbf{q}} \right)^T.
\]

(3.3)

When the contact torques of the stick-slip elements are added to equation (3.1) the equations of motion take the form,

\[
M \ddot{\mathbf{q}} - \mathbf{h} - \sum_{j \in I_S} w_j T_j - \sum_{i \in I_T} w_i T_i = 0.
\]

(3.4)

The contact torques can be passive torques of sticking contacts or active torques of sliding contacts. The torques in sliding contacts are expressed by the corresponding normal forces using Coulomb’s friction law (1.4),

\[
T_j = -\text{sign}(\dot{g}_j) \mu_j r_m j F_{N_j}, \quad F_{N_j} \geq 0, \quad j \in I_S.
\]

(3.5)

In this thesis only associated Coulomb friction is allowed, which means that the normal force \( F_N \) is known in advance and independent of the solution \((q, \dot{q})\) —in the applications used here, the normal force will be an input variable. The torques of potentially sticking contacts will be calculated using an LCP algorithm, which takes into account the unilateral constraints induced by sticking. The relation between relative acceleration and the transmitted torque in the contacts will be converted into a standard LCP problem. First the relative accelerations \( \dot{\gamma}_i \) of the potentially sticking contacts are formulated,

\[
\dot{\gamma}_i = w^T_i \ddot{\mathbf{q}}, \quad i \in I_T.
\]

(3.6)

Introducing,

\[
\begin{align*}
\lambda_T &= (\ldots, T_i, \ldots)^T \quad i \in I_T, \\
W_T &= (\ldots, w_i, \ldots) \\
\ddot{\gamma}_T &= (\ldots, \dot{\gamma}_i, \ldots)^T \\
\lambda_S &= (\ldots, T_j, \ldots)^T \quad j \in I_S, \\
W_S &= (\ldots, w_j, \ldots)
\end{align*}
\]

(3.7)

it is possible to write equations (3.4) to (3.6) as

\[
M \ddot{\mathbf{q}} - \mathbf{h} - W_S \lambda_S - W_T \lambda_T = 0, \\
\ddot{\gamma}_T = W_T^T \ddot{\mathbf{q}}.
\]

(3.8)

### 3.2 Complementarity Properties of Friction

In this thesis, contact between elements will be evaluated with respect to the frictional law of Coulomb. With regard to friction the following three cases are possible looking at a relative velocity \( g \),

\[
\begin{align*}
g &= 0 & \Rightarrow & \quad |\lambda_T| \leq \mu_0 \lambda_N \quad \text{(sticking)}, \\
g &< 0 & \Rightarrow & \quad \lambda_T = +\mu_0 \lambda_N \quad \text{(negative sliding)}, \\
g &> 0 & \Rightarrow & \quad \lambda_T = -\mu_0 \lambda_N \quad \text{(positive sliding)}.
\end{align*}
\]

(3.9)

This law states that during sliding the tangential friction force \( \lambda_T \) is proportional to the normal force \( \lambda_N \) existing at a contact, with \( \mu_0 \) as a general friction coefficient. For a frictional contact process like sliding or sticking, stiction is expressed by a zero relative tangential velocity and by a zero friction saturation. The friction saturation is the difference between the maximal (static) friction force and the tangential constraint resulting from system dynamics. It expresses the fact
3.3. DECOMPOSITIONS OF CONTACT LAW

that the tangential constraint force lies in the friction cone under consideration, and it gives a
measure for the 'constraint force distance' from the friction cone boundary, beyond which sliding
occurs. The above stated properties include a basic principle of contact dynamics, which states that
for a contact, either magnitudes of relative kinematics are zero and the corresponding constraint
forces or constraint force combinations are not zero, or vice versa. The scalar product of both
sets of magnitude is therefore always zero. This establishes the well known complementarity in
contact mechanics, sometimes termed Signorini’s law or corner law [Pfeiffer, 2001]. The quantity
force can be replaced with torque when there is a matter of rotational relative velocities, \( \gamma \), instead
of translational relative velocities, \( g \).

3.3 Decompositions of contact law

3.3.1 Decomposition by Pfeiffer and Glocker

In this section the decomposition of the contact law as proposed in [Pfeiffer and Glocker, 1996] will
be introduced. The set-valuedness of friction torques only applies to sticking contacts, for sliding
contacts they are linearly dependent on the normal force in the contact. Therefore, looking at
(3.8), the torques of the potentially sticking contacts in \( \lambda^p_T \) have to be dealt with. By transforming
Coulomb’s law (3.9) to the acceleration level for the contacts in \( I_T \) one obtains,

\[
| \lambda^p_{Ti} | \leq \mu_0 r_{mi} F_{Ni} \Rightarrow \dot{\gamma}_i = 0 \quad \lambda^p_{Ti} = + \mu_0 r_{mi} F_{Ni} \Rightarrow \dot{\gamma}_i < 0 \quad \lambda^p_{Ti} = - \mu_0 r_{mi} F_{Ni} \Rightarrow \dot{\gamma}_i > 0
\]

\[
\land F_{Ni} \geq 0 \quad \land | T_i | \leq \mu_0 r_{mi} F_{Ni}, \quad (3.10)
\]

and \( i \in I_T \). This is graphically represented in Figure 3.1. The equations of motion (3.8) and

![Figure 3.1: Contact law on acceleration level for potentially sticking contacts](image)

the constraint equations in (3.10) represent a complete description of the system for all transient
states. In order to transform this set of equations into a more suitable and numerically solvable
form the contact law (3.10) has to be converted into a complementary form. For clarity, the de-
composition of the contact law will also be shown graphically.

The fundamental idea for the decomposition of the friction characteristic in Figure 3.1 is to for-
mulate the tangential constraint as four simultaneously appearing unilateral constraints. This will
be done in two decomposition steps. In order to derive the first decomposition step we look at
the admissible values of the stick torques in \( \lambda^s_T \) from (3.7), every element of this set must lie in
its appropriate convex set \( C_{Ti} \),

\[
\lambda^s_{Ti} \in C_{Ti} = \{ \lambda^s_{Ti} | - \mu_0 r_{mi} F_{Ni} \leq \lambda^s_{Ti} \leq + \mu_0 r_{mi} F_{Ni} \}. \quad (3.11)
\]
Now it is assumed that this torque is transmitted by two simultaneously appearing constraints with each of them transferring only a part of $\lambda_T$. These parts are denoted in positive and negative direction respectively by $\lambda_T^{(+)}$ and $\lambda_T^{(-)}$, and state $\lambda_T$, as,

$$\lambda_T = \lambda_T^{(+)} - \lambda_T^{(-)}. \tag{3.12}$$

The superscripts $(+)$ and $(-)$ are used only for distinctive reasons and have no physical meaning. The values of $\lambda_T^{(+)}$ and $\lambda_T^{(-)}$ are not arbitrary but must be chosen in such a manner that the tangential force $\lambda_T$ always lies in the convex set $C_T$ defined in (3.11). This can be ensured by values of $\lambda_T^{(+)}$ and $\lambda_T^{(-)}$ to,

$$\lambda_T^{(+)} \in C_T^{(+)} = \{ \lambda_T^{(+)i} | 0 \leq \lambda_T^{(+)i} \leq \mu_0 r m_i F_{N_i} \},$$

$$\lambda_T^{(-)} \in C_T^{(-)} = \{ \lambda_T^{(-)i} | 0 \leq \lambda_T^{(-)i} \leq \mu_0 r m_i F_{N_i} \}. \tag{3.13}$$

Next for each of the new variables $\lambda_T^{(+)}$ and $\lambda_T^{(-)}$ we have to define a characteristic which connects them to the corresponding tangential relative acceleration $\dot{\gamma}_i$. This is done in the middle diagrams of Figure 3.2. It can be proven that these characteristics are equivalent to the upper diagram of Figure 3.2, which corresponds to the friction law in equation (3.10). In the second decomposition step, each characteristic in the middle of Figure 3.2 has to be split into two unilateral constraints. This enables the formulation of an LCP by using the resulting inequalities and complementary conditions together with the dynamical equations. First the tangential acceleration is split into its positive and negative parts, respectively $\dot{\gamma}_i^+$ and $\dot{\gamma}_i^-$,

$$\dot{\gamma}_i^+ = \dot{\gamma}_i^+ - \dot{\gamma}_i^-; \quad \dot{\gamma}_i^- \geq 0; \quad \dot{\gamma}_i^+ \geq 0; \quad \text{for} \ C_T^{(+)}; \quad \text{and}$$

$$\dot{\gamma}_i^- = \dot{\gamma}_i^+ - \dot{\gamma}_i^-; \quad \dot{\gamma}_i^- \geq 0; \quad \dot{\gamma}_i^+ \geq 0; \quad \text{for} \ C_T^{(-)}. \tag{3.14}$$

where we define $\dot{\gamma}_i^+ = z_i^+ = \frac{1}{2}(\dot{\gamma}_i^+ + \dot{\gamma}_i^-), \dot{\gamma}_i^- = z_i^- = \frac{1}{2}(\dot{\gamma}_i^+ - \dot{\gamma}_i^-)$. Note that the case of both values simultaneously being greater than zero is excluded. The physical meaning of the auxiliary variables $z_i^+$ and $z_i^-$ is obvious; they denote the positive and negative parts of the accelerations and are introduced for distinctive reasons only. The values of $\lambda_T^{(+)}$ and $\lambda_T^{(-)}$ in equation (3.13) can each be expressed by two inequality conditions,

$$\lambda_T^{(+)} \geq 0; \quad \mu_0 r m_i F_{N_i} - \lambda_T^{(+)i} \geq 0,$$

$$\lambda_T^{(-)} \geq 0; \quad \mu_0 r m_i F_{N_i} - \lambda_T^{(-)i} \geq 0. \tag{3.15}$$

Now the complementarity conditions between the variables in equations (3.14) and (3.15) are stated,

$$\dot{\gamma}_i^+ \lambda_T^{(+)} = 0; \quad z_i^- \left( \mu_0 r m_i F_{N_i} - \lambda_T^{(+)i} \right) = 0,$$

$$\dot{\gamma}_i^- \lambda_T^{(-)} = 0; \quad z_i^+ \left( \mu_0 r m_i F_{N_i} - \lambda_T^{(-)i} \right) = 0, \tag{3.16}$$

which are also graphically depicted in the lower part of Figure 3.2. The upper left and lower right characteristics in the lower part of the figure do not yet have the standard form of a unilateral constraint. This form, however, can be achieved by shifting the corners to the origin and reflecting the characteristics with respect to the ordinate axis, which is done by the transformations,

$$\lambda_T^{(-)} = \mu_0 r m_i F_{N_i} - \lambda_T^{(+)};$$

$$\lambda_T^{(+)} = \mu_0 r m_i F_{N_i} - \lambda_T^{(-)}. \tag{3.17}$$

The variables $\lambda_T^{(-)}$ and $\lambda_T^{(+)}$ are called friction saturations and define the differences of the maximal transferable and actual tangential torques. When the friction saturations become zero, a transition to sliding is possible. Finally we state all of the inequalities and complementary conditions (3.14).
Figure 3.2: Decomposition of contact law on acceleration level
CHAPTER 3. EVENT-DRIVEN INTEGRATION METHOD

(3.15) and (3.16), which can also be seen from the lower part of Figure 3.2

\[ \dot{\gamma}^{-i} \geq 0; \lambda_{T_i}^{(-)} \geq 0; \dot{\gamma}^{-i} \lambda_{T_i}^{(-)} = 0 \]
\[ \dot{\gamma}^{+i} \geq 0; \lambda_{T_i}^{(+)T_i} \geq 0; \dot{\gamma}^{+i} \lambda_{T_i}^{(+)T_i} = 0 \]
\[ \lambda_{T_0}^{(+)} \geq 0; z^{+}_i \geq 0; \lambda_{T_0}^{(+)T_i} z^{+}_i = 0 \]
\[ \lambda_{T_0}^{(-)} \geq 0; z^{-}_i \geq 0; \lambda_{T_0}^{(-)T_i} z^{-}_i = 0 \]

This set of relations completely describes the decomposed friction characteristic which correspond to four unilateral constraints. For each of the potentially sticking contacts these equations have to be used. In order to take into account all those contacts, equations (3.12), (3.14), (3.17) and (3.18) have to be rewritten in vector notation;

\[ \lambda_{T} = \lambda_{T}^{(+)T} - \lambda_{T}^{(-)T}, \]
\[ \dot{\gamma} = z^{+} - z^{-}; \quad \dot{\gamma} = \dot{\gamma}^{+} - \dot{\gamma}^{-}, \]
\[ \lambda_{T_0}^{(+)} = \bar{\mu}_{T} - \lambda_{T}^{(-)T}; \quad \lambda_{T_0}^{(-)} = \bar{\mu}_{T} - \lambda_{T}^{(+)T}, \]
\[ \dot{\gamma}^{-} \geq 0; \lambda_{T}^{(-)} \geq 0; \dot{\gamma}^{-} \lambda_{T}^{(-)} = 0, \]
\[ \dot{\gamma}^{+} \geq 0; \lambda_{T}^{(+)T} \geq 0; \dot{\gamma}^{+} \lambda_{T}^{(+)T} = 0, \]
\[ \lambda_{T_0}^{(+)} \geq 0; z^{+} \geq 0; \lambda_{T_0}^{(+)T} z^{+} = 0, \]
\[ \lambda_{T_0}^{(-)} \geq 0; z^{-} \geq 0; \lambda_{T_0}^{(-)T} z^{-} = 0, \]

with \( \bar{\mu}_{T} = \{\mu_{0}, r_{mi}, F_{N_i}\}, \quad i \in I_T. \)

The physical meaning of each of the sixteen \((2^4)\) complementarity combinations has been investigated and the results are available in Appendix C.

Model LCP Formulation

The decomposition of the stick/slip contacts has to be incorporated in the general model (3.8),

\[ M \ddot{q} - h - W_S \lambda S - \begin{bmatrix} W_T & -W_T \end{bmatrix} \begin{bmatrix} \lambda_T^{(+)T} \\ \lambda_T^{(-)T} \\ \lambda_T^{(+)T_0} \\ \lambda_T^{(-)T_0} \end{bmatrix} = 0, \]

\[ \begin{bmatrix} \dot{\gamma} \\ -\dot{\gamma} \end{bmatrix} = \begin{bmatrix} W_T \\ -W_T \end{bmatrix} \dot{q}. \]

The equations from (3.20) are also written in matrix notation, which yields,

\[ \begin{bmatrix} \dot{\gamma} \\ -\dot{\gamma} \end{bmatrix} = \begin{bmatrix} \dot{\gamma}^{+} \\ \dot{\gamma}^{-} \end{bmatrix} = \begin{bmatrix} E & 0 \\ 0 & E \end{bmatrix} \begin{bmatrix} z^{-} \\ z^{+} \end{bmatrix}, \]

\[ \begin{bmatrix} \lambda_{T_0}^{(-)} \\ \lambda_{T_0}^{(+)T_0} \end{bmatrix} = \begin{bmatrix} -E & 0 \\ 0 & -E \end{bmatrix} \begin{bmatrix} \lambda_T^{(+)T} \\ \lambda_T^{(-)T} \end{bmatrix} + \begin{bmatrix} \bar{\mu}_{T} \\ \bar{\mu}_{T} \end{bmatrix}. \]
Finally we state the inequalities and complementary conditions from (3.21),

\[
\begin{align*}
\begin{bmatrix} \hat{\gamma}_D^+ \\ \lambda_D \\ \lambda_{T_0 D}^-(+) \\ z_D^+ \\
\end{bmatrix} & \geq 0; \\
\begin{bmatrix} \lambda_D^+ \\ \lambda_D^- \\ \lambda_{T_0 D}^-(+) \\ z_D^+ \\
\end{bmatrix} & \geq 0; \\
\begin{bmatrix} \hat{\gamma}_D^- \\ \lambda_D \\ \lambda_{T_0 D}^-(+) \\ z_D^+ \\
\end{bmatrix} & \geq 0; \\
\begin{bmatrix} \hat{\gamma}_T^- \\ \lambda_D \\ \lambda_{T_0 D}^-(+) \\ z_D^+ \\
\end{bmatrix} & = 0.
\end{align*}
\]

(3.26)

Now equations (3.22) to (3.27) are rewritten by using the terms under the braces,

\[
\begin{align*}
M \ddot{q} - h - W_S \lambda_S - W \lambda_D &= 0, \\
\dot{\gamma}_T &= W^T \dot{q}, \\
\dot{\lambda}_{T_0 D} &= -I \lambda_D + M_T, \\
\dot{\gamma}_D &\geq 0; \quad \lambda_D \geq 0; \quad \dot{\gamma}_D^T \lambda_D = 0, \\
\lambda_{T_0 D} &\geq 0; \quad z_D \geq 0; \quad \lambda_{T_0 D}^T z_D = 0.
\end{align*}
\]

(3.28) to (3.33)

In the next step equation (3.28) is solved with respect to \(\dot{q}\) and substituted into (3.29), which is equal to (3.30). This leads to,

\[
\begin{align*}
\dot{\gamma}_D - I z_D &= W^T M^{-1} W \lambda_D + W^T M^{-1} (h + W_S \lambda_S), \\
\dot{\lambda}_{T_0 D} &= -I \lambda_D + M_T, \\
\dot{\gamma}_D &\geq 0; \quad \lambda_D \geq 0; \quad \dot{\gamma}_D^T \lambda_D = 0, \\
\lambda_{T_0 D} &\geq 0; \quad z_D \geq 0; \quad \lambda_{T_0 D}^T z_D = 0.
\end{align*}
\]

(3.34) to (3.37)

Finally, by stating the equations, inequalities and complementary conditions of (3.34) to (3.37) in matrix notation,

\[
\begin{align*}
\begin{bmatrix} \dot{\gamma}_D \\ \lambda_{T_0 D} \\
\end{bmatrix} &= \begin{bmatrix} W^T M^{-1} W & I \\ -I & 0 \\
\end{bmatrix} \begin{bmatrix} \lambda_D \\ z_D \\
\end{bmatrix} + \begin{bmatrix} W^T M^{-1} (h + W_S \lambda_S) \\ M_T \\
\end{bmatrix}, \\
\begin{bmatrix} \dot{\gamma}_D \\ \lambda_{T_0 D} \\
\end{bmatrix} &\geq 0; \quad \begin{bmatrix} \lambda_D \\ z_D \\
\end{bmatrix} \geq 0; \quad \begin{bmatrix} \dot{\gamma}_D^T \\ \lambda_{T_0 D}^T \\
\end{bmatrix} \begin{bmatrix} \lambda_D \\ z_D \\
\end{bmatrix} = 0.
\end{align*}
\]

(3.38) to (3.39)

we achieve an LCP in its standard form,

\[
y = Ax + b \quad \land \quad y \geq 0; \quad x \geq 0; \quad y^T x = 0.
\]

(3.40)

A drawback of the decomposition proposed by Pfeiffer and Glockler is the amount of slack variables introduced. These slack variables introduce ‘abundant modes’ in the LCP. All the modes for one contact have been elaborated in Appendix C.

3.3.2 Decomposition by Leine and Nijmeijer

The conclusions of Appendix C instigate to the search for a more efficient decomposition. The abundant modes in the LCP lead to a larger dimension of the LCP and thus an increase of
computing time, especially when the number of contacts grows. A better decomposition of the friction characteristic is written down in \cite{Leine2004}, which halves the dimension of the LCP. The decomposition of the contact law on acceleration level, represented in Figure \ref{fig:3.3}, involves the splitting of all $\dot{\gamma}_i$ in right and left parts. In vector notation,

$$
\begin{align*}
\dot{\gamma}_R &= \frac{1}{2}(\dot{|\gamma}| + \dot{\gamma}) , \\
\dot{\gamma}_L &= \frac{1}{2}(\dot{|\gamma}| - \dot{\gamma}) , \\
\dot{\gamma} &= \dot{\gamma}_R - \dot{\gamma}_L.
\end{align*}
$$

(3.41)

Furthermore the friction saturations for each contact need to be defined $\lambda_{R,i}$ and $\lambda_{L,i}$, in vector notation,

$$
\begin{align*}
\lambda_R &= \mu_T + \lambda_T , \\
\lambda_L &= \mu_T - \lambda_T.
\end{align*}
$$

(3.42)

where $\overline{\mu}_T = \{\mu_0, r_m F_{N_i}\}$. Addition of the equations in (3.42) yields,

$$
\lambda_R = 2 \overline{\mu}_T - \lambda_L.
$$

(3.43)
The friction saturation is, as mentioned earlier in Section 3.3.1, the difference between the maximal (static) friction torque and the tangential constraint torque resulting from system dynamics. It expresses the fact that the tangential constraint torque lies in the friction cone under consideration, and it gives a measure for the ‘constraint torque distance’ from the friction cone boundary, which indicates sliding. The friction saturations \( \lambda_R \) and \( \lambda_L \) are complementary to the acceleration vectors \( \gamma_R \) and \( \gamma_L \), and can therefore be used to set up an LCP on acceleration level for the tangential contact problem. When the second equation of (3.42) is substituted in (3.8), the generalized accelerations read,

\[
\ddot{q} = M^{-1} \left( h + W_s \lambda_s + W_T \bar{\mu}_T \right) - M^{-1} W_T \lambda_L.
\] (3.44)

This generalized acceleration is subsequently substituted in the definition of the relative acceleration in (3.8), which combined with (3.41) gives,

\[
\dot{\gamma}_L = -W_T^T M^{-1} \left( h + W_s \lambda_s + W_T \bar{\mu}_T \right) + W_T^T M^{-1} W_T \lambda_T + \dot{\gamma}_R.
\] (3.45)

Equations (3.45) and (3.43) form together an LCP on acceleration level for the tangential contact problem,

\[
\begin{pmatrix}
\dot{\gamma}_L \\
\lambda_R \\
y
\end{pmatrix} =
\begin{pmatrix}
W_T^T M^{-1} W_T & I & 0 \\
-I & 0 & 0
\end{pmatrix}
\begin{pmatrix}
\lambda_L \\
\dot{\gamma}_R \\
x
\end{pmatrix}
+ \begin{pmatrix}
W_T^T M^{-1} \left( h + W_s \lambda_s + W_T \bar{\mu}_T \right) \\
2 \bar{\mu}_T
\end{pmatrix},
\] (3.46)

with \( 0 \leq x \perp y \geq 0 \).

### 3.4 LCP Solution Algorithms

The unknowns in (3.40) and (3.46) are the vectors \( y \) and \( x \) which contain the relative accelerations, the contact forces and the friction saturations. The events of continual contact and sticking, as well as the transitions to sliding and separation, are included in this formulation. After solving (3.40) the values of \( y \) and \( x \) are obtained and the complementarity condition \( x_i y_i = 0 \) is taken into account by either \( x_i \geq 0; y_i = 0 \) or \( x_i = 0; y_i \geq 0 \). The LCP formulation of a system with unilateral constraints and friction is much more general than the representation of a frictionless bilaterally constrained system, but leads to additional complications with respect to existence and uniqueness of solutions. The most straightforward method for solving an LCP is the enumerative method, which simply tries randomly or in a preprogrammed manner, all possible combinations of the LCP; The complementarity conditions \( x_i y_i = 0 \) of an LCP of dimension \( n \) provide \( 2^n \) different combinations of \( n \) variables which are allowed to be greater than zero at the same time. The enumerative method can be used for small values of \( n \). For large \( n \), however, these methods become impractical since \( 2^n \) grows very rapidly.

The other, smarter, LCP solution algorithms can roughly be divided in three groups [Wetzels, 1999]:

- **Homotopic algorithms**, which solve the LCP by pivoting a solution matrix, e.g. Lemke and Katzenelson.

- **Iterative algorithms**: treat the LCP as an optimization problem. The solution is reached by minimizing a error criterium.

- **Contraction algorithms**: consider the LCP as a nonlinear algebraic problem, using contraction or Newton-Raphson.
3.4.1 Enumerative Method

The complementarity condition $x_i y_i = 0$ from (3.40) and (3.46) is taken into account by either $x_i \geq 0; y_i = 0$ or $x_i = 0; y_i \geq 0$. In the strategy of the enumerative method a certain combination of elements of the vector $y$ in (3.40) is set equal to zero, which means that the complementary variables in $x$ can be greater or equal to zero and vice versa. The choice for a certain mode is done with the help of a binary numerator $B$; both equations (3.40) and (3.46) can be rewritten,

$$By - AB^N x = b, \quad H\alpha = b, \quad \alpha = H^{-1}b,$$

where,

$$H = B - AB^N; \quad \alpha = By + B^N x,$$

$$B = \begin{bmatrix} b_1 & 0 & \cdots & 0 \\ 0 & b_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & b_n \end{bmatrix} \quad \wedge \quad b_i \in \{0, 1\},$$

$$B^N = \begin{bmatrix} b_1^N & 0 & \cdots & 0 \\ 0 & b_2^N & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & b_n^N \end{bmatrix} \quad \wedge \quad b_i^N \in \{0, 1\} \quad \wedge \quad b_i^N \neq b_i.$$

If the solution vector for a particular mode $\alpha_i$ suffices the condition $\alpha_i \geq 0$, then the solution of the LCP is found. The most computational effort in solving the LCP is calculating the inverse of $H$, which has to exist of course. This inverse always exists if the condition is fulfilled, that the determinants of all submatrices of $A$ greater are than zero. If the LCP is of the $n^{th}$ order, the maximum number of iterations is $2^n$.

Because in a high level model of an automotive transmission mode switches only occur at a few time steps, a smart extension of the enumerative method is to first try the mode which led to the solution in the previous time step.

3.4.2 Lemke’s Method

Lemke’s algorithm is a solution algorithm that solves the LCP with a smart choice for the next mode in the iteration process. Here the $n$-dimensional LCP in (3.40) is rewritten by introducing an artificial variable $\nu \geq 0$ together with a vector $e$ whose $n$ entries are all +1,

$$y = Ax + b + e\nu \quad \wedge \quad y \geq 0; \ x \geq 0; \ y^T x = 0.$$  

(3.49)

At this point the variables $w$, $z$, $F$ and $l$ are introduced and (3.49) is written as,

$$w = Fz + l + e\nu \quad \wedge \quad w \geq 0; \ z \geq 0; \ w^T z = 0.$$  

(3.50)

The variables $w$ and $z$ can contain both elements from $y$ and $x$. The algorithm considers $w$ to be the solution vector of the LCP so the elements of $z$ are set to zero. Only this single mode of (3.50) is considered in this algorithm. So equation (3.50) reduces to:

$$w = l + e\nu$$  

(3.51)

There always exists a very large $\nu$, that ensures the condition $w \geq 0$ is fulfilled and thus a solution is found. This is a solution to the LCP in (3.50), but not the solution to the original LCP in (3.40).
because it still contains the artificial variable $\nu$. To erase $\nu$, it is brought as close to zero until one of the complementary variables hits the boundary of the condition $w \geq 0$. This variable, also called the ‘blocking’ variable is then exchanged with its complementary counterpart in $z$. In this operation $F$ and $l$ are also updated to keep the original system. Then again in (3.51) $\nu$ is brought as close to zero until $w \geq 0$ is not fulfilled anymore, and so on. Eventually $\nu$ will be able to be set to zero whilst $w \geq 0$ holds. This is then the solution to the original LCP in (3.40).

Lemke’s algorithm does not choose the next mode in the iteration process randomly, but exchanges in (3.50) the complementary variables first to fail the complementary conditions (‘blocking’ variables).

### 3.5 Implementation

In Appendix D the Matlab/Simulink S-function is given for the the Dual-clutch system modelled with the event-driven integration method with mode selection by LCP evaluation. There is chosen for implementing the decomposition proposed in [Pfeiffer and Glocker, 1996], because the decomposition proposed in [Leine and Nijmeijer, 2004] will be used with the time-stepping method. Furthermore Lemke’s algorithm is chosen for the LCP evaluation because of two reasons: Firstly because it is more sophisticated than the enumerative method which bluntly checks all solutions in a predetermined order. Secondly because it is well-established and widely used (also in financial and economics toolboxes). In Appendix E an M-file function (authors: Paul L. Fackler & Mario J. Miranda) of Lemke’s algorithm is available.
Chapter 4

Time-stepping Method

Time-stepping schemes are based on a time-discretization of the system dynamics including the contact conditions in normal and tangential direction. The whole set of discretized equations and constraints is used to compute the next state of the motion. The time-stepping method makes no distinction between impulsive forces (e.g. due to impacts) and finite forces, because integrals of forces over each time-step are used instead of the instantaneous values of the forces. Only increments of the positions and velocities are computed. The acceleration $\ddot{q}$, which becomes infinite for impulsive forces, is thus not computed by the algorithm. The positions and velocities at the end of the time-step are found by solving an LCP which describes the contact problem or by means of an augmented Lagrangian approach, which will be explained in this chapter. In contrast to event-driven schemes this method needs no event-detection. The time-stepping method does not even use the notion of events and none or perhaps one index set is used. Multiple events might take place during one time-step, and the algorithm computes the overall integral of the forces over this time-step, which is finite. The time-stepping method is especially useful when one is interested in the global motion of systems with many contact points, leading to a large number of events. Time-stepping methods do not make use of the usual equations of motion, which relate acceleration to forces, but replace the equations of motion on acceleration level by an equality of measures [Glocker, 2001],

$$M(t, q)\dot{u} = h(t, q, u)dt + W(t, q)d\Lambda,$$

(4.1)

with $u = \dot{q}$, $W$ consisting of the constraint vectors and $\Lambda$ the impulsive friction force/torque. The lefthand side of the equation is the change in momentum –instead of acceleration– of the system and the righthand side the resultant impulse –instead of resultant force/torque– acting on the system. The time-stepping method is generally applied to systems with impacts, because impacts are not present in the models in this thesis, $d\Lambda = \lambda dt$ applies.

4.1 Discretization and Moreau’s Algorithm

From [Leine and Nijmeijer, 2004] a derivation of the difference-equation of motion can be found. There Moreau’s time-stepping method, which is a special kind of midpoint DAE-integrator, is presented. It uses the positions $q_A$ and velocities $u_A$ known at the beginning of the time-step at time $t_A$ to first take a half time-step for the positions at the midpoint $q_M = q_A + \frac{1}{2} \Delta t u_A$. The midpoint is used to classify the status of the normal constraints, i.e. determine if the contacts are engaged. For the simulations in this thesis, contacts are presumed to be engaged at all times (no impacts). The velocity $u_E$ at the end of the time-step, $t_E = t_A + \Delta t$, is calculated by a trapezoidal scheme,

$$M(u_E - u_A) = h_M \Delta t + W_T \lambda_T \Delta t,$$

(4.2)
with the set-valued force law from (4.3),

\[ \lambda_{T,i} \in -\mu_{i}r_{m,i}\text{Sign}(\gamma_{i})F_{N,i}, \quad (4.3) \]

and \( h_{m} = h(t_{M}, q_{m}, u_{A}) \).

### 4.2 Time-stepping with LCP

In this section the time-stepping method with an LCP to determine the set-valued contact forces will be introduced. Coulomb’s friction characteristic is split up again into two unilateral complementary conditions, as in Section 3.3.2, only this time on velocity level. The relative rotational velocities and their right and left parts at time \( t = t_{E} \) are defined as,

\[ \gamma_{E} = \gamma_{RE} - \gamma_{LE}, \quad \gamma_{RE} = \frac{1}{2}(|\gamma_{E}| + \gamma_{E}), \quad \gamma_{LE} = \frac{1}{2}(|\gamma_{E}| - \gamma_{E}), \quad (4.4) \]

and the friction saturations,

\[ \lambda_{R} = \bar{\mu}_{T} + \lambda_{T}, \quad \lambda_{L} = \bar{\mu}_{T} - \lambda_{T}, \quad (4.5) \]

with again \( \bar{\mu}_{T} = \{\mu_{0}, r_{m,i}, F_{N,i}\} \). Using (4.4) and (4.5), the force law (4.3) can be written as the set of complementarity conditions,

\[ 0 \leq \gamma_{RE} \perp \lambda_{R} \geq 0, \quad 0 \leq \gamma_{LE} \perp \lambda_{L} \geq 0. \quad (4.6) \]

Substitution of (4.2) in \( \gamma_{TE} = W_{T}^{T}u_{E} \) together with equations (4.4) and (4.5) gives,

\[ \gamma_{LE} = W_{T}^{T}M^{-1}W_{T}^{T}(\lambda_{L} - \bar{\mu}_{T}) - W_{T}^{T}(u_{A} + M^{-1}h_{M}\Delta t) + \gamma_{RE}. \quad (4.7) \]

Equation (4.7) together with the addition of the two equations from (4.5) form an LCP,

\[ \begin{pmatrix} \gamma_{LE} \\ \lambda_{R} \\ \lambda_{L} \end{pmatrix} = \begin{pmatrix} 0 & -I \\ -I & 0 \\ \bar{\mu}_{T} & 0 \end{pmatrix} \begin{pmatrix} \lambda_{L} \\ \gamma_{RE} \\ x \end{pmatrix} + \begin{pmatrix} W_{T}^{T}(u_{A} + M^{-1}h_{M} + M^{-1}W_{T}^{T}\bar{\mu}_{T}) \\ 2\bar{\mu}_{T} \end{pmatrix}, \quad (4.8) \]

with \( 0 \leq x \perp y \geq 0. \) The LCP (4.8) is solved in each integration step. The velocities \( u_{E} \) are subsequently found with (4.2) and for the positions holds \( q_{E} = q_{M} + \frac{1}{2}\Delta t u_{E}. \)

In Appendix F the Matlab/Simulink S-function is available where the time-stepping method with LCP evaluation is used to model the Dual-clutch system.

### 4.3 Time-stepping with Augmented Lagrangian Approach

The contact problem can also be solved with an augmented Lagrangian approach. The augmented Lagrangian method is identical with the exact regularization method. The method transforms differential inclusions for set-valued force laws to a set of non-smooth continuous algebraic equations, which might be solved by a root-finding algorithm. The idea of an exact regularization is to shift the regularized contact law such that the exact solution is obtained. This seems a paradox, because the solution has to be known in advance. Indeed, one has to solve for the shift together with the equation of motion and the regularized contact law, such that the exact solution is obtained. So, instead of determining the contact forces from a set-valued contact law, the shift from
4.3. TIME-STEPPING WITH AUGMENTED LAGRANGIAN APPROACH

A single-valued contact law is calculated. The latter is much easier because we can solve for it with a root-finding algorithm. The admissible values of each of the elements of $\lambda$ form a convex set and are bounded by the values of the (externally applied) normal force,

$$C_i = \{ \lambda_i | -\mu_i r_{m_i} F_{N_i} \leq \lambda_i \leq \mu_i r_{m_i} F_{N_i} \}.$$  

(4.9)

As regularization the following single-valued relation between $\lambda$ and $\gamma$ will be used,

$$\lambda_i = \text{prox}_{C_i}(\lambda_i^* - p_i \gamma_i), \quad p_i > 0,$$

(4.10)

where $\lambda^*$ is the shift of the regularization and $p > 0$ is a steepness parameter (the slope) of the regularization, as can be seen in Figure 4.1. The proximal point of a convex set $C$ to a point $z$,

```
prox_{C}(z),
```

is the closest point in $C$ to $z$. Let $x = \text{prox}_{C}(z)$, then it holds that $x \in C$ and,

$$\begin{cases}
  x = z, & \text{if } z \in C, \\
  x \in \text{bdry}C, & \text{if } z \notin C,
\end{cases}$$

where $\text{bdry}C$ is the boundary of the convex set $C$. Equations (4.2) and (4.10) result into the following set of equations,

$$\begin{align*}
  M \Delta \boldsymbol{u} - h \Delta t - W \lambda \Delta t = & \quad 0, \\
  \lambda^*_{T_k} - \text{prox}_{C}(\lambda^*_{T_k} - p \gamma) = & \quad 0.
\end{align*}$$

(4.11)

These equations can be solved with a root-finding algorithm like a fixed point iteration method. The iteration starts with an initial value for the contact torque, $\lambda^*_{T_0}$, of zero, with which the rotational velocities, $\boldsymbol{u}$, are calculated according to the first equation of (4.11). From these values the resulting relative rotational velocity, $\gamma_0(\boldsymbol{u})$, is determined. With this $\gamma_0$, according to the second equation of (4.11), $\lambda^*_{T_1}$ is determined, which again is used to calculate $\gamma_1$. This iteration is maintained until either a shift $\lambda^*_{T_k}$ is found which results in $\gamma_k = 0$ (stick mode), or when $\lambda^*_{T_{k-1}} = \lambda^*_{T_k} \in \text{bdry}C$ (slip mode) in two following iterations.

We therefore solve for the shift $\lambda^*$, but the difference between the shift and $\lambda$ is formal as they agree at the solution, i.e. $\lambda = \lambda^*$. A very elegant method is therefore obtained to solve a differential inclusion which describes a frictional contact problem. The method can be extended to deal with impacts also very efficiently. A fundamental background on this method, together with the proof that the exact regularization method is identical to the augmented Lagrangian method in optimization theory can be found in [Leine and Nijmeijer, 2004].

In Appendix G the Matlab/Simulink S-function is available where the time-stepping method with augmented Lagrangian approach is used to model the Dual-clutch system.
Chapter 5

SimDriveline

With SimDriveline, it is possible to model drivetrain and powertrain systems in an easy way within MATLAB and Simulink. It is a block diagram modelling environment for the engineering design and simulation of drivelines, or idealized powertrain systems. With SimDriveline, one can represent a driveline machine with a connected block diagram, like other Simulink models, and blocks can be grouped into hierarchical subsystems. Rotational motion can be initiated and maintained in a driveline with actuators while measuring, via sensors, the motions of driveline elements and the torques acting on them. Sensor signals can be returned to the driveline via actuators, forming feedback loops and the basis for controls. The SimDriveline libraries offer blocks to represent rotating bodies; gear constraints among bodies; special dynamic elements such as spring-damper forces, rotational stops, and clutches; transmissions; and sensors and actuators. SimDriveline is part of Simulink Physical Modelling, encompassing the modelling and design of systems according to basic physical principles. Physical Modelling runs within the Simulink environment and interfaces seamlessly with the rest of Simulink and with MATLAB. Unlike other Simulink blocks, which represent mathematical operations or operate on signals, Physical Modelling blocks represent physical components or relationships directly. In the light of the project the method used for clutch modelling is studied.

5.1 Controllable Friction Clutch

The Controllable Friction Clutch in the SimDriveline Library encompasses three states; unengaged, when it applies no friction at all; engaged, when it applies kinetic friction; and locked, when it applies static friction. There is also a fourth, virtual state between locked and unlocked called the wait state. If it locks, a Controllable Friction Clutch block imposes a constraint on the driveline system by requiring that two otherwise independent angular velocities be equal. A locked clutch thus reduces the number of independent degrees of freedom by one. Locking requires that the absolute relative speed $|\gamma|$ be smaller than a velocity threshold $\eta$. The static friction torque controls the unlocking of a friction clutch. The clutch can optionally be locked at the start of the simulation as well. When the clutch is locked, it remains locked unless the friction constraint torque across the clutch exceeds a static friction limit. By default, SimDriveline decides when to unlock a clutch after repeatedly suspending the simulation in time, entering a nonphysical algebraic loop, and testing a set of unlocking conditions. This virtual testing is called mode iteration. If the pressure in the clutch is large enough to ensure that the absolute value of the friction constraint torque, $T_{fc}$, in the clutch does not exceed the static friction limit, $\mu_0 F_N$, the friction clutch locks the two driveline axes together when $|\gamma| \leq \eta$. When the clutch is locked and the absolute value of the friction constraint torque exceeds the static friction limit, the clutch enters the wait state and might unlock. The unlocking of a friction clutch is a conditional, multistep process implemented internally by SimDriveline:
1. SimDriveline first checks the relative acceleration $\dot{\gamma}$ of the two driveline axes, given the torques present when the clutch enters the wait state. If the whole machine requires the axes to turn in the relative forward direction, but $\dot{\gamma}$ is negative; or if the whole machine requires the axes to turn in the relative reverse direction, but $\dot{\gamma}$ is positive, the clutch returns from the wait state to the locked state.

2. If the clutch remains in the wait state instead of returning to locked, SimDriveline integrates the relative acceleration in time to obtain the absolute value of the virtual angular velocity and checks this result against angular velocity tolerance $\eta$. If the result is within the stick band, the clutch returns to the start of the wait state and the relative acceleration check. If the result exceeds the stick band, the clutch unlocks.

3. In the unlocked state, the clutch begins applying kinetic friction again. SimDriveline also begins again to check for the locking condition ($|\gamma| \leq \eta$ with $T_{fc} \leq \mu_0 F_N$).

In Figure 5.1 the Matlab/Simulink model is shown where the Dual-clutch system is modelled. The comparison of this simulation method with the other methods is available in Chapter 6.
Chapter 6

Evaluation; The Dual-clutch Case

In the preceding chapters five methods for the modelling of automotive transmissions have been introduced. In this chapter the results of the implementation of these methods will be discussed. For the discussion the same system has been modelled with every method, the Dual-clutch is chosen as case-system because it is not too complex to model with the Switch Model since it has still only two frictional elements and because it is very representative for an advanced automotive transmission.

6.1 Dual-clutch Simulations

The Dual-clutch system is shown in Figure 6.2. For each method an S-function is written to represent the Dual-clutch system in Simulink as in Figure 6.1 is done for the augmented Lagrange approach. The Dual-clutch is also built in Simulink with the blocks available in the SimDriveline library, this model can be seen in Figure 5.1.

![Simulink Model with augmented Lagrange approach](image)

Figure 6.1: Simulink Model with augmented Lagrange approach

The timespan of the simulation of the system is ten seconds. The inputs are chosen in such a manner that both frictional elements will reach their stick mode at least once, as can be seen in Figure 6.2 where the simulation trajectories are plotted. There are two main criterias on which the different methods have to be evaluated; computation time and accurateness. Accurateness, however, is less important when for simulation only the global motion of the whole system is of...
interest. For simulations involving gear rattling the accurateness is very important to quantify the vibrations stemming from it. In the case of simulations for parameter variation of the transmission, one is more interested in the global motion of the transmission and computation time becomes a bigger issue as usually many simulations have to be run. The following simulation parameters have been varied to determine their influence:

- tolerances variation
- sample time
- regularization steepness (augmented Lagrange approach)

Because time-stepping is a fixed-step simulation method in which the next time-step is calculated with an Euler-scheme, both the Switch Model and event-driven integration method have also been simulated with a fixed time-step with the Matlab solver 'ODE1 (Euler)'. In Appendix H all the results of the simulations are shown.

### 6.2 Results

The results available in Appendix H have been summarized in Table 6.2. It shows that the Switch Model is definitely the fastest simulation method, which seems also obvious because all contact modes have to be programmed a priori in the simulation model, whereas the other methods call some other algorithm to determine stick/slip transitions. Furthermore the mean error between the Switch Model and the event-driven integration method, which both use Coulomb’s law on acceleration level, is very small compared to the mean error of these methods with the time-stepping methods, which use Coulomb’s law on velocity level. This also applies for the mean
6.2. RESULTS

<table>
<thead>
<tr>
<th>Modelling Method</th>
<th>Computation Time</th>
<th>Accurateness</th>
<th>Model Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Switch Model</td>
<td>++</td>
<td>++</td>
<td>– –</td>
</tr>
<tr>
<td>Event-driven with LCP</td>
<td>+</td>
<td>++</td>
<td>+</td>
</tr>
<tr>
<td>Time-stepping with LCP</td>
<td>0</td>
<td>+</td>
<td>++</td>
</tr>
<tr>
<td>Time-stepping with augmented Lagrangian</td>
<td>0</td>
<td>+</td>
<td>++</td>
</tr>
<tr>
<td>SimDriveline</td>
<td>++</td>
<td>+</td>
<td>++</td>
</tr>
</tbody>
</table>

Table 6.1: Results Dual-clutch Simulations

error between the time-stepping methods. Simulation time and accurateness with the augmented Lagrange approach can be exchanged with one another quite easily by adjusting the steepness parameter and regularization tolerance. Finally it is stated as a result of this case that the SimDriveline package incorporates the easiest way to construct a model. With its ‘plug-and-play’ strategy it is not even necessary to derive the equations of motion.

It has to be reminded that these conclusions have been drawn on the basis of simulations with the Dual-clutch system, which is a high-level model of an automotive transmission with just a few stick/slip transitions. More detailed models, models with more compliances or models with far more stick/slip transitions (e.g. gear rattling analysis) could lead to slightly different conclusions. But with the information in this report preliminary conclusions on that matter can be drawn.
Chapter 7

Conclusions and Recommendations

7.1 Conclusions

The first part of this thesis addresses the difficulties that arise when modelling and simulating discontinuous systems, with the emphasis on automotive applications. Systems exposing a mixed continuous and discrete nature are also called hybrid systems [Heemels, 1999]. Different modelling approaches are introduced and applied to an automotive drivetrain. The Switch Model maintains the continuity of the state vector and yields a set of non-stiff ordinary differential equations. A disadvantage of the Switch Model is the exponentially increasing complexity of the logical structure with increasing number of switching boundaries. The Switch Model is therefore not suitable for mechanical systems with many frictional contacts because the combinatorial contact problem becomes too large.

One approach to model multibody systems with unilateral contacts results in a set of nonlinear ordinary differential equations of second order together with some inequality constraints due to unilateral contacts, [Pfeiffer and Glocker, 1996] and [Leine and Nijmeijer, 2004]. The problem can be converted in a (N)LCP and the solution is achieved by integrating the unilateral constrained system in smooth phases of the motion and computing the actual contact configuration with the help of Lemke’s algorithm [Cottle et al., 1992]. Due to the instantaneous changes in the contact force over a stick-slip transition, the new contact configuration of a system is far from obvious when multiple contacts are active. However, each change of the contact configuration, that means the beginning or the end of a contact event, has to be evaluated by indicators, which is time-consuming, especially in case of many multiple contacts. Moreau introduced the time-stepping method in [Moreau, 1988], which uses integrals of forces over each time-step instead of the instantaneous values of the forces. The time-stepping method does therefore not use the notion of ‘events’. Time-stepping in combination with an LCP formulation works well for a large amount of practical problems as in automotive applications, however there are still unsolved problems for large systems. Especially with respect to spatial contact configurations including friction and with respect to systems performing impacts with friction in several of the existing contacts [Pfeiffer et al., 2005].

A completely different approach uses the Augmented Lagrangian. A very good description is given by [Leine and Nijmeijer, 2004], which also shows that the augmented Lagrangian method in optimization theory is identical to the exact regularization method. The augmented Lagrangian method transforms differential inclusions for set-valued force laws to a set of non-smooth continuous algebraic equations, which might be solved by a root-finding algorithm. The time-stepping method can be combined with the augmented Lagrangian method for integrating the set of non-smooth nonlinear equations resulting from the evaluation process including the augmented La-
grange function and the relevant optimization procedure. Finally, with the recently introduced SimDriveline package, automotive transmission models can be created and simulated in a plug and play manner within Matlab/Simulink. In this package also some sort of iteration method is used to determine the contact torques in clutches and other components incorporating unilateral contacts. Unfortunately, literature accompanying the package remains silent on the exact method implemented.

After reading this part one should be able to choose the approach that is most efficient for the type of model and simulation the reader desires.

7.2 Recommendations

The different modelling approaches have been mutually compared and conclusions on their performances are drawn. Still there are some topics of research. It would be advisable to find out what modelling method is used within the SimDriveline package. Furthermore, time-stepping methods are quite recently introduced and until now only fixed-step algorithms are known. A variable-step algorithm would absolutely improve the (computational) efficiency of the time-stepping method. Also in this thesis the modelling methods were used on high level models of automotive transmissions. These high level models are both used for parameter variation analysis and controller synthesis. For parameter variation analysis of phenomena like gear rattling more detailed models are necessary, and next to friction also impact has to be dealt with efficiently. An extension of the time-stepping method would be able to achieve this. But because time-stepping methods would step over the impacts during a time-step, it is not clear a priori what the best modelling method would be for gear rattling analysis. Finally, it is recommended to investigate methods for controller synthesis.
Part II

Coordinated Control of Advanced Automotive Transmissions
Chapter 8

Preliminaries

8.1 Automotive Drivetrain Control

Not only does the trend towards more advanced automotive transmissions instigate to the use of more sophisticated modelling methods, also the control strategies that operate the drivetrains cannot depend on heuristics only anymore. The control of an automotive drivetrain with an automated manual transmission can be divided in four levels as is depicted in Figure 8.1.

![Figure 8.1: Drivetrain control hierarchy](image)

The four levels can be defined in the following way:

- Strategic Control; the highest level in the control hierarchy, is a open-loop controller. Here the user’s driveability wishes are translated in specific weights for criteria as fuel economy,
acceleration responsiveness and comfort. These weights are then fed into the supervisory controller.

- **Supervisory Control**: the supervisor determines the most important states of the drivetrain in terms of optimal driveability, i.e. the desired driveshaft torque, $T_{ds,d}$, and desired gear. It calculates these states on the basis of the user’s instantaneous inputs (e.g. gas, brake), peripheral conditions (weather conditions, vehicle state) and the drivetrain states (from measurements).

- **Coordinated Control**: the coordinated controller simply coordinates the setpoints for all the different components of the drivetrain (e.g. desired engine torque, $T_{e,d}$, desired clutch torque, $T_{cl,d}$ and if present, the desired flywheel brake-clutch torque, $T_{f,d}$ (B-IST concept)) to let the drivetrain operate in its optimal state (optimal state determined by the supervisor). But when a gear change is required, the coordinated controller has to coordinate the different components to achieve an optimal gear shift operation according to the criteria weights imposed by the supervisor (e.g. fast clutch engagement for better fuel economy against slow clutch engagement for more comfort).

- **Component Control**: at the lowest level, typical SISO control loops are found to ensure the desired setpoints from the coordinated controller are reached.

The requirements for the controllers on the supervisory and coordinated level are similar; they have to find some optimal balance between conflicting criteria within a certain set of constraints. These requirements strongly correspond to those often encountered in the process industry. There Model Predictive Control is the strategy most often used in these sort of control problems.

## 8.2 Optimal, Constrained Control

In this part of this work there is looked for a control strategy which optimizes performance with respect to a predetermined criterium, also called *optimal control*. Of course this performance has to be achieved within a set of constraints, which can be divided in *state* and *input constraints*.

### 8.2.1 Optimal Control

A control strategy to obtain maximum performance of a system can be achieved by minimizing a certain weighted criterium cost function. Dynamic Programming (DP) and Bellman’s Optimality Principle \cite{Bellman,1957}, which roughly states that every part of an optimal trajectory is optimal itself, is a way to resolve the input that leads to the minimum of the criterium cost function and thus maximum performance. A particularly nice approach for the control design is to use quadratic optimization, because it leads to simple closed-form solutions for linear systems (LQR, Ricatti equation).

### 8.2.2 Constrained Control

There are various strategies to approach constrained control problems and deal with the constraints \cite{Goodwin et al., 2005}:

- **cautious**: with this approach performance is deliberately reduced to not violate any of the constraints.

- **serendipitous**: no special precautions are taken to handle constraints, actuators reach saturation and states exceed their allowed values.
8.2. OPTIMAL, CONSTRAINED CONTROL

- evolutionary; this approach starts from the serendipitous approach but then adds modifications and embellishments to avoid the negative consequences of the constraints, whilst ensuring that performance goals are attained.

- tactical; here the constraints are formulated from the beginning in the control design process.

Intuitively the tactical approach will lead to the best result, the other approaches work their way around the constraints instead of using them to maximize the performance.
Chapter 9

Dynamic Programming

Dynamic Programming (DP) interprets any optimization as a multistage decision problem, which is broken into several single-stage subproblems. This reduces computations drastically. The results of DP can become known locally after the search for the best trajectory is completed. DP can be used both in continuous and discrete-time domains. Let us take a system of the form,

$$\dot{x} = f(x(t), u(t), t),$$  \hfill (9.1)

with a cost function/performance criterium, $J$, defined as,

$$J = \nu g(x(te), te) \int_{t_0}^{t_e} h(x(t), u(t), t) dt,$$  \hfill (9.2)

with $g(x(te), te)$ the end cost function and $\nu$ the accompanying weight factor. Furthermore $h(x(t), u(t), t)$ is defined to be the process cost function. The goal is to find an input signal, $u_o(t)$, for $t \in [t_0, t_e]$, such that

$$u_o(t) = \arg \min_{u(t)} \left\{ \nu g(x(te), te) + \int_{t_0}^{t_e} h(x(t), u(t), t) dt \right\}. \hfill (9.3)$$

In DP both closed-loop and open-loop minimization of the cost can be used. In open-loop minimization all inputs are calculated in advance whilst with closed-loop minimization the control signal of each time sample is calculated with the knowledge of the current state. For deterministic systems there is no difference in result between open-loop and closed-loop minimization. Obviously for non-deterministic systems, with closed loop policies, it is possible to achieve lower cost, essentially by taking advantage of the extra information (the value of the current state). The reduction in cost is called the value of information and can be significant indeed [Bertsekas, 1995].

If the information is not available, the controller cannot adapt appropriately to unexpected values of the state, and as a result the cost can be adversely affected. Form the foregoing it can be concluded that the open-loop policy has a fixed control horizon and that the closed-loop policy has to work with a receding control horizon.

The DP algorithm can be implemented both analytically and numerically. Analytical implementation leads to closed-form solutions, which are from a computational point of view very attractive; the cost minimization and with that the control law are computed in advance and the reduces the online control problem to a simple function evaluation. But when the cost function $J$ is not differentiable or the system model is nonlinear, minimization cannot be done analytically. With numerical implementation the state space has to be discretized and the cost at each point has to be calculated, the number of points increase exponentially with the increase of states. This refers to Bellman’s ‘curse of dimensionality’, which next to the state vector, also holds for the dimension of the input vector. There are various techniques to diminish the computational burden.
like multi-pass numerical DP, where first an optimization is run on a coarse grid and in a second optimization of the same process the grid in a band around the initial coarse optimal trajectory, is refined. Still an advantage with the numerical implementation is the ability to impose hard constraints to the optimization process.

With reinforcement learning and neuro-DP a non-differentiable cost function $J$ is replaced by a differentiable function with higher-order or exponential terms, whose variables are adjusted so this function fits the cost function. And let the minimum of this function lead to the optimal input signal.

Another way to deal with non-differentiable cost functions is to calculate the costs of all possible state combinations and construct with this a look-up-table, which can be accessed online to determine the optimal path at every time step.

9.1 Optimal Clutch Engagement with Numerical DP

In this section the results will be presented of a case where the most optimal engagement of a clutch is determined with numerical DP with an open-loop policy. Recently, the engagement control of automotive dry clutches is becoming more and more important, due to the increasing use of automated manual transmission in modern vehicles. The engagement control of dry clutches must satisfy different and sometimes conflicting objectives; small friction losses, fast lockup, preservation of driver comfort (driveability). To this aim several control strategies have been proposed in literature as can be read in [Bemporad et al., 2001a]. In this section the engagement of the clutch will be considered as a multi-stage decision problem and the optimal input will be computed by a DP algorithm based on the principal of optimality, formulated in [Bellman, 1957]. There is started with a discretization of the state space of the system. The model used for this exercise is shown in Figure 2.1. In the DP algorithm the Switch Model from Chapter 2 will be incorporated. The vector $[\omega_1 \omega_2]^T$ is used as state vector for the discretization, with at every integer state combination a grid-point. The ‘no engine-stall’-condition will be used to constraint the state space discretization to a minimum rotational velocity of the engine of 80 rad/sec, the maximum rotational velocity is set to 150 rad/sec. Furthermore the relative velocity $\gamma = \omega_1 - \omega_2$ is constraint between 150 rad/sec and the minimum of the stick band, $-\eta$ rad/sec. The engine torque $T_1$ is assumed constant and the load torque $T_2$ is linear dependent on the rotational velocity of $J_2$. The clutch torque $T_c$ can be controlled by the clutch normal pressure $T_c = \mu_0 u$, but only during slip. The lumped friction coefficient $\mu_0$ is set to 1. The cost function is determined to be;

$$J = \sum_{i=1}^{N} \left( \kappa \Delta u + \nu \gamma^2 + \xi (\omega_1 - \omega_{ref})^2 \right) + \theta \dot{\gamma}_{lockup},$$  \hspace{1cm} (9.4)

where $\kappa$, $\nu$, $\xi$, $\theta$ are weights, $\Delta u$ is the input increment, $\omega_{ref}$ the reference rotational velocity of the engine, and $\dot{\gamma}_{lockup}$ the relative acceleration near a slip-stick transition. Furthermore $N$ is the number of time-steps. The largest part of the algorithm is the ‘administration’ part, for each time-step the reached states with its cost and current input have to be stored. When from different begin states within a time-step the same end state is reached only the one with the lowest cost is saved (Bellman’s Principal of Optimality) and used as begin state in the next time-step. When the last time-step is done from all end states satisfying the predetermined conditions (in the simple clutch case this is sticking), the one with the lowest cost is taken. It is necessary to store at each time-step for each reached state the complete input history to be able to find back the most optimal input $u_o$. The complete algorithm has been depicted in Figure 9.1. As one can see the algorithm designed by the author proceeds in a forward manner in time. An equivalent backward algorithm can also be used. The choice for forward DP is preferred since then a unique initial state before the clutch engages can be chosen and more importantly, there are several end states that satisfy the end constraint (i.e. stick) from which the one with the lowest cost can be selected after the algorithm finishes. This results in a feedforward only strategy.

An other implementation of this control strategy can make use of the value of information by
Figure 9.1: Dynamic Programming Algorithm Clutch Engagement
letting the control input depend on the current state. In order to achieve this the most optimal input at each grid point of the discretization has to be calculated in advance and stored in a look-up table. The online control action consists only looking up the control action dependent on the current state. This is already very similar to a Model Predictive Control (MPC) strategy.

9.1.1 Results

The DP algorithm is programmed in such a manner that the evolution of the states through time can be monitored, a screenshot is shown in Figure 9.3. For the optimization procedure the following weights have been chosen; \( \kappa = 3 \), \( \nu = 5 \), \( \xi = 0 \), \( \theta = 0.1 \ast N \). The optimization has a time span of 1 second with a sample time of 0.05 second which leads to 20 stages for the DP algorithm. The result is shown in Figure 9.3. As one can see from the results the optimal control strategy stemming from the algorithm is mostly concerned in reducing the relative velocity, meanwhile satisfying the constraint of a minimum rotational velocity of the engine of 80 rad/sec. At the time of lockup the strategy tries to have a minimal relative acceleration. It was also attempted to add the engine torque \( T_1 \) to the input vector, but this effort was cancelled by the effect of the ‘curse of dimensionality’; computation time grew out of proportion.

The application of this numerical DP lies in the set-point generation and feedforward design. Because in engineering almost never is worked with deterministic systems, it is very desirable to add feedback to the control strategy.

9.2 Analytical Dynamic Programming

If one wants to incorporate feedback control in the DP control strategy, this can be achieved with analytical DP. In [Haj-Fraj and Pfeiffer, 2001] this method is applied to achieve optimal control of gear shift operations in automatic transmissions. In the concerned publication the gear upshifting

\[ \begin{align*}
\text{Je} & \quad \text{One way clutch} \quad \text{Jv} \\
\quad \text{Wet clutch} \\
& \quad \text{Pre-control phase}; \text{ the wet clutch engages and torque starts passing through this branch also while the torque through the one-way clutch decreases.} \\
& \quad \text{Controllable phase}; \text{ starts when all the torque passes through the wet clutch and thus the one-way clutch unlocks. At this time there is no fixed kinematic relationship anymore between the input and output shaft.}
\end{align*} \]

Figure 9.2: Automatic Transmission model for 1-2 upshift
Figure 9.3: DP Algorithm: Monitor (upper) and Results (lower)
• **Synchronous point;** this is when the wet clutch sticks and the two shafts again have a fixed kinematic relationship (second gear). No inputs are available for control purposes.

In the publication a linearized model for the controllable phase is derived in state space form. Then a cost function is defined in the following way,

$$ J = \kappa g(x(t_e), t_e) + \int_{t_0}^{t_e} h(x(t), u(t), t) \, dt. \quad (9.5) $$

The process cost function \( h(x(t), u(t), t) \) evaluates the cost of the process in the second phase, whereas the end cost function \( g(x(t_e), t_e) \) evaluates the cost resulting from the end state \( x(t_e) \), which is the synchronous point. The weighting factor \( \kappa \) is included to permit the adjustment of the relative importance of the terms in \( J \). Making use of the DP method based on the principle of optimality, formulated in [Bellman, 1957], the controllable phase of the gear shift can be considered as a multistage decision process. For this aim the state space model is discretized and also the cost function, which results in,

$$ J = \kappa G(x_K) + \sum_{k=0}^{K-1} H(x_k, u_k, k). \quad (9.6) $$

For the process cost function \( H(x_k, u_k, k) \) the acceleration change is marked as the most critical issue which affects the passenger comfort during the gear shift process; the smoother the acceleration the more comfortable is the gear shift. The process cost function is formulated to be,

$$ H(x_k, u_k, k) = \gamma_k^2 + u_k^T R_k u_k. \quad (9.7) $$

The process cost function reflects the desire to keep the jerk close to zero without excessive expenditure of the control effort. The end cost function \( G(x_K) \) depends only on the end state, but it can describe a behavior which extends over a time interval after the synchronous point. In [Haj-Fraj and Pfeiffer, 2001] a criterion is chosen which evaluates the jerk after the end of the gear shift over a period of time. Since no control can be applied to the powertrain after synchronization and the end state of the controllable phase acts as begin condition for the next phase, the whole trajectory after the synchronous point can be formulated as a function of the end state only.

Now the method of DP can be applied to determine an optimal control law which can lead the system from the initial state \( x_0 \) to the end state \( x_K \) which satisfies the constraint \( \gamma(x_K) = 0 \) by minimizing the overall cost function \( J \). Because a backward DP algorithm is used, there is started with the cost resulting from driving the system during the last stage to the required end state of the process. The end state \( x_K \) depends on \( x_{K-1} \) and \( u_{K-1} \), but because of the constraint \( \gamma(x_K) = 0 \) there is only one control vector which satisfies and this control input is only dependent on again \( x_{K-1} \). So the cost function for the last stage has only one dependence, \( J_{K-1}(x_{K-1}) \). In the next step the cost of operation over the last two stages is considered,

$$ J_{K-2} = J_{K-1}(x_{K-1}) + H(x_{K-2}, u_{K-2}). \quad (9.8) $$

Again because \( x_{K-1} \) is a function of \( x_{K-2} \) and \( u_{K-2} \) - i.e. \( x_{K-1}(x_{K-2}, u_{K-2}) \) - also the cost function \( J_{K-2} \) is a function of \( x_{K-2} \) and \( u_{K-2} \) - i.e. \( J_{K-2}(x_{K-2}, u_{K-2}) \). To minimize the cost \( J_{K-2}(x_{K-2}, u_{K-2}) \), its partial derivative is set to zero,

$$ \frac{\partial J_{K-2}(x_{K-2}, u_{K-2})}{\partial u_{K-2}} = 0, \quad (9.9) $$

which leads to an unique solution because \( J_{K-2}(x_{K-2}, u_{K-2}) \) is quadratic in \( u_{K-2} \). This unique solution is the input with the lowest cost and thus the optimal input at time \( K-2 \), \( u_{K-2}^{\text{opt}} \). Then it is posed that the matrix of second partials,

$$ \frac{\partial^2 J_{K-2}(x_{K-2}, u_{K-2})}{\partial^2 u_{K-2}} = 0, \quad (9.10) $$
is positive definite, which yields that $u_{K-2}^{\text{opt}}$ is the absolute or global minimum. When $u_{K-2}^{\text{opt}}$ is substituted in $J_{K-2}(x_{K-2}, u_{K-2})$ one gets the lowest cost from that stage to the end state, $J_{K-2}^{\text{opt}}(x_{K-2})$.

Subsequently for the next stage, the cost is again defined as:

$$J_{K-3} = J_{K-2}^{\text{opt}}(x_{K-2}) + H(x_{K-3}, u_{K-3}),$$  \hspace{1cm} (9.11)

and the same procedure as above is used to obtain $J_{K-3}^{\text{opt}}(x_{K-3})$. This is done recursively until the first stage of the process to obtain the optimal control law in explicit, analytical form. It is important to note here that the optimal control law consists of both a feedforward and a feedback portion. Also it is important to realize that this solution is based on the assumption that the control values are not bounded. Since the control expenditure is included in the cost function, the control values can be forced to lie in the admissible intervals, by choosing appropriate weighting. Of course this at the expense of performance on the other criteria of the cost function. Because of this disadvantage this method is not implemented in a case. Instead another method was found which does not have this disadvantage, explicit Model Predictive Control.
Chapter 10

Explicit Model Predictive Control

10.1 Introduction

When Model Predictive Control (MPC) was introduced to the world in the late seventies in the last century, its use was limited to ‘slow’ systems like in the process industry (e.g. petrochemical). In these applications it grew enormously popular and has nowadays become the standard in the process industry. The rationale for its success lies in its effective handling of constraints, which allows it to truly apply optimal control under practical circumstances.

MPC is a moving or receding control horizon strategy. This means that at a specific time sample $t_k$ from state measurements the controller computes for $K$ following time samples (i.e. the control horizon) the optimal, constrained control, then applies only the input for the current time sample. This is done for each next time sample and in this way the control horizon recedes through time.

The drawback of this method is that at each time sample a quadratic program has to be solved, which is quite a computational cost. The computation time has prohibited the use of this method to applications in the automotive and other mechatronic areas. But in recent publications this drawback is eliminated by a method which makes the MPC controller available in explicit form.

The breakthrough in this field was a.o. described in [Bemporad et al., 2002a], where a technique is presented to compute the explicit state-feedback solution to both the finite and infinite horizon linear quadratic optimal control problem subject to state and input constraints. It is shown in [Bemporad et al., 2002a] that this closed form solution is piecewise linear and continuous and identical to the original MPC controller. In the publication an algorithm for a multi-parametric-QP (mp-QP) solver is presented and because the MPC control problem can be converted in a mp-QP problem, the explicit form of the control law can be calculated. Consequently, it is stated, that constrained linear quadratic regulation becomes attractive also for systems with high sampling rates, as on-line quadratic programming solvers are no more required for the implementation.

Thus the range of applicability of MPC grows to problems where anti-windup schemes and other ad-hoc techniques dominated up to now. Such an explicit form of the controller provides also additional insight for better understanding the control policy of MPC.

In this project there has been made use of the Hybrid Toolbox which is managed by Prof. A. Bemporad and is available from http://www.dii.unisi.it/~bemporad. In several publications automotive applications of the explicit MPC controller have been shown; in [Bemporad et al., 2001a] the control of dry clutch engagement is treated, [Bemporad et al., 2001b] is concerned with traction control and [Bemporad et al., 2003] uses MPC for the hybrid control of an automotive robotized gearbox to reduce emissions.
10.2 Hybrid Toolbox

As said in the previous section, there has been made use of the Hybrid Toolbox for Matlab and Simulink. It can be read in [Bemporad, 2004] that there are three types of controllers available in this toolbox:

- **@LINCON** objects; controllers for constrained linear systems based on quadratic optimization. This controller needs as input arguments: LTI model, controller **TYPE** (either constrained regulator or output reference tracking), **COST** structure (i.e. different weights on state, input, etc.), **INTERVALS** structure (a.o. control horizon) and **LIMITS** structure (contains the constraints on inputs, input increments and outputs).

- **@HYBCON** objects; controllers for hybrid systems based on mixed-integer linear optimization. This controller needs as inputs: a HYSDEL model (explained in Section 10.3), **COST** structure (i.e. different weights on state, input, etc.), control horizon **N**, **LIMITS** structure (contains the constraints on inputs, input increments and outputs) and **REFSIGNALS** structure (contains reference signals on outputs, inputs, states and z-vectors).

- **@EXPCON** objects; explicit controllers for linear or hybrid systems. This controller needs as input: a constrained optimal controller (either **@LINCON** object, **@HYBCON** object or MPC Controller defined by the MPC Toolbox for Matlab), **RANGE** structure (contains the range of initial states and references (i.e. the parameters of the multiparametric program) for which the explicit solution is computed).

As can be read in [Bemporad et al., 2002a] the stability of MPC feedback loops was investigated by numerous researchers and is in the meanwhile well established. Stability is, in general, a complex function of the various tuning parameters. Most approaches for proving stability follow in spirit the arguments of [Keerthi and Gilbert, 1988] which establishes the fact that under some conditions, the value function \( V(t) = J(u_{\text{opt}}(t); t) \) attained at the minimizer \( u_{\text{opt}}(t) \) is a Lyapunov function for the system. Further specific stability results can be read in [Bemporad et al., 2002a]. Reachability analysis (or safety analysis or formal verification) aims at detecting if a hybrid model will eventually reach an unsafe state configuration or satisfy a temporal logic formula. Reachability analysis relies on a reach set computation algorithm, which strongly depends on to the mathematical model of the system. In the Hybrid Toolbox an algorithm is available that computes the set of states that a given hybrid system can reach within a certain amount of sampling times from any initial state and under any possible excitation. The importance of reach set computations is twofold. Firstly it allows one to check for safety/liveness properties, for instance that the trajectories of the hybrid system will never enter some unsafe regions of state space, or that all the trajectories will reach a target region within a given maximum time. Interesting control-theoretical questions like stability and observability can be reformulated as reachability questions. Secondly, the proposed algorithmic ingredients for reachability analysis can be suitably used for optimal control purposes: if an optimization stage is performed in parallel with reach-set computation, the latter can be selectively carried out according to convenient strategy that discards evolutions which are recognized not to be optimal, and finally determines the desired optimal input sequence. How this is applied in detail can be read in [Torrisi et al., 2003].

10.3 HYSDEL

A modelling language was proposed in [Torrisi et al., 2003] to describe Discrete Hybrid Automata (DHA) models, called HYbrid System DEscription Language (HYSDEL). DHA models are a form in which hybrid systems can be represented. The HYSDEL description of a DHA is an abstract modelling step. The associated HYSDEL compiler then translates the description into several computational models, in particular into a Mixed Logical Dynamical (MLD) system and into
PieceWise Affine (PWA) form, which are again other forms in which a hybrid system can be represented. More detailed elaboration on these different forms for hybrid systems can be found in [Bemporad, 2003]. HYSDEL can also generate a simulator that runs as a function in Matlab. A HYSDEL list is composed of two parts: the first one, called INTERFACE, contains the declaration for all the variables and parameters, so that it is possible to make the proper type checks. The second part, IMPLEMENTATION, is composed of specialized sections where the relations among the variables are defined. Within this part there are different sections which enable one to define the four different elements of a DHA:

- Switched Affine System (SAS); An SAS is a collection of linear affine systems. A SAS can be rewritten as the combination of linear terms and if-then-else rules.
- Event Generator (EG); An EG is a mathematical object that generates a logic signal according to the satisfaction of a linear affine constraint.
- Finite State Machine (FSM); An FSM is a discrete dynamic process that evolves according to a logical state update function.
- Mode Selector (MS); An MS selects the dynamic mode of the SAS through a boolean function. Because this toolbox is discrete-time-based, a mode switch can only occur at sampling instants.

Further information on HYSDEL can be found in [Torrisi et al., 2002], [Bemporad, 2003] and [Bemporad, 2004].

### 10.4 Dry clutch engagement

As a first exercise the control problem of Section 9.1 –optimal clutch engagement– is taken. Instead of only one input (i.e. clutch torque) also the engine torque is taken as input. The system including stick/slip transitions has been modelled in HYSDEL and the result can be seen in Appendix I.1. The SAS in the model consists of three linear affine systems, one represents positive slip, one negative slip and one represents stick. The EG monitors the sign of the relative velocity, the continuous stick condition when in stick and if a transition to stick is imminent. When an event occurs, the MS will select the new mode of the SAS according to the logic rules in the FSM. A H@HYBCON controller is designed with three performance criteria in the cost function; tracking of the reference engine speed, $\omega_{e ref}$, tracking of zero relative velocity, $\gamma_{ref}$ and smooth lockup, $\dot{\gamma}_{ref}$,

$$
J = \sum_{k=1}^{N} \left( ||\theta(\omega_e - \omega_{e ref})||_2 + ||\kappa(\gamma - \gamma_{ref})||_2 + ||\nu(\dot{\gamma} - \dot{\gamma}_{ref})||_2 \right)
$$

(10.1)

Where it has to be mentioned that the smooth lockup cost only becomes active when the relative velocity passes a threshold value towards zero (stick). The weight factors are tune to the following values: $\theta = 1$, $\kappa = 4$ and $\nu = 4$. The control horizon is set to 10 time-steps of 0.01 seconds. The other controller parameters are available in Appendix I.2 as is the Simulink model. The ‘Switch’-block in the Simulink-model is to switch from the hybrid controller, after the clutch lockup, to a simple control law ensuring continuous stick. The results can be seen in Figure 10.1. The figure clearly shows that in the first part the dominant action is to reduce the relative velocity. Also it can be seen that the constraint on the minimum engine rotational velocity (80 rad/sec) is perfectly handled. In the last part the controller puts all his effort in ensuring a smooth lockup.

### 10.5 Coordinated Control of the B-IST

As a final exercise the hybrid toolbox will be used to obtain a coordinated control strategy for the B-IST, developed by DTI. The layout of the B-IST is depicted in Figure 10.2. The benefit of
CHAPTER 10. EXPLICIT MODEL PREDICTIVE CONTROL

Figure 10.1: Optimal engagement of clutch with model predictive approach

Figure 10.2: Schematic layout of B-IST
the B-IST is both fuel-efficiency and improved driveability [Werner, 2004]. The B-IST module is an add-on component for an Automated Manual Transmission (AMT). The heart of this module is the planetary gear set which, with its three branches, is connected to the engine flywheel, the secondary axis from the transmission and to an extra flywheel. This flywheel can be decelerated by a brake clutch which connects it to the bell house. The module compensates the so-called ‘torque-dip’, inherently associated with the AMT when shifting gears, by applying torque to the driveshaft via the planetary gear set when the clutch is open. In this light the B-IST can be seen as a torque assistant. For upshifting in lower gears the brake clutch on the flywheel is used to decelerate the engine while still transmitting torque to the driveshaft. When driving in a high gear with low engine rotational velocity and acceleration of the vehicle is desired by the driver, a downshift to a lower gear is made and the flywheel is used to instantly accelerate both the engine and the vehicle during the shift. With this strategy a gearbox with lower ratios can be used for the AMT and the vehicle can be driven with much lower engine rotational velocities than without B-IST module. This can improve fuel-efficiency with almost 10%.

For this exercise an optimal, constrained coordinated controller will be designed for an upshift from first to second gear and for an upshift from second to third gear. The inputs for this system are $T_E$, the engine torque, $F_{NB}$ the flywheel brake normal force, and $F_{NC}$, the clutch normal force. With these inputs the drive shaft torque has to be controlled and the shift has to be as short as possible. The shift can be divided into three parts:

- **Clutch release:** In this phase the clutch is disengaged to be able to make the gear shift in the AMT in the next phase. The clutch normal force is brought to zero while the engine torque and flywheel brake torque compensate have to ensure that this has no negative effects on the driveshaft torque.

- **Gear change:** In this phase there is no torque transfer through the transmission and the actual gear change can take place. During this phase the clutch normal force is zero and still the engine torque and flywheel brake torque have to ensure an optimal course for the drive shaft torque.

- **Clutch engagement:** In this phase the engine rotational speed has to be synchronized again with the rotational speed of the primary axis of the transmission. In the current configuration of the B-IST, the flywheel speed will change its rotational direction during an upshift from second to third gear and it is necessary to have the brake clutch on the flywheel disengaged at the time the rotational speed crosses zero (no-flywheel-stick constraint). This because a stick/slip transition will induce an extra discontinuity. This phase and with that the entire shift, ends when the clutch is in stick again.

From a control perspective the shift is divided in two parts:

- The first part is the clutch release phase and the gear change phase together. A weighted penalty is allocated to setpoint deviation of the clutch torque, $T_{Cref}$, driveshaft torque, $T_{DSref}$, and flywheel speed, $\omega_{Fref}$. Furthermore a weighted penalty is allocated to the use of the flywheel brake, $T_B$. Even though the brake is useful for torque assistance, using it is sheer energy dissipation. This leads to the following definition of the cost function for the first part,

$$J_I = \sum_{k=1}^{N} \left( ||\theta(\omega_F - \omega_{Fref})||_2 + ||\kappa(T_{DS} - T_{DSref})||_2 + ||\nu(T_B - T_{Bref})||_2 + ||\xi(T_C - T_{Cref})||_2 \right).$$

(10.2)

- The second part consists of the clutch engagement, where both the flywheel brake and relative velocity are penalized as is setpoint deviation of the driveshaft torque. There is also a possibility to penalize the relative acceleration for small relative velocity as is done in Section 10.4 to ensure smooth lockup. But by setting the constraint for the minimum
relative velocity to zero, the controller already approaches this constraint (and thus the slip/stick transition) smoothly. This leads to the following definition of the cost function for the second part,

$$J_{II} = \sum_{k=1}^{N} \left( ||\theta(\gamma - \gamma_{ref})||_2 + ||\kappa(T_{DS} - T_{DSref})||_2 + ||\nu(T_B - T_{Bref})||_2 \right). \quad (10.3)$$

A simplified representation of the B-IST vehicle, with its parameters and sign conventions for torques and rotational velocities, is represented in Figure [10.3].

![Figure 10.3: Schematic representation of B-IST](image)

The equations of motion become,

$$J_F \dot{\omega}_F = T_F - T_B, \quad (10.4)$$
$$J_E \dot{\omega}_E = T_E - T_A - T_C, \quad (10.5)$$
$$J_{PS} \dot{\omega}_S = \frac{T_C}{r_T} + \frac{T_P}{r_P} - r_D T_{DS}, \quad (10.6)$$
$$J_V \dot{\omega}_V = T_{DS} - T_V, \quad (10.7)$$

where $J_{PS} = J_S + J_P/r_F^2$ and with,

$$\dot{T}_{DS} = k(\omega_S r_D - \omega_V), \quad (10.8)$$
$$T_V = b_V \omega_V, \quad (10.9)$$
$$T_C \in -\mu_C r_m C \text{Sign}(\omega_E - \omega_C) F_{NC}, \quad (10.10)$$
$$T_B \in -\mu_B r_m B \text{Sign}(\omega_F) F_{NB}. \quad (10.11)$$

In Appendix J, an expression for the acceleration of the inertia attached to the sun –here the flywheel $J_F$– is derived. This is substituted in (10.4), which leads to the following expression for the flywheel torque,

$$T_F = T_B + J_F \left( \frac{z + 1}{r_F} \dot{\omega}_S - z \dot{\omega}_E \right). \quad (10.12)$$
This expression is the substituted in equations (10.5) and (10.6) according to (10.4) and this leads to the following equations,

\[
J_L \dot{\omega}_E = \left( \frac{r_{GN}}{z^2} J_{PS} + J_F \right) T_E - \left( \frac{r_{GN}}{z^2} J_{PS} + r_{GN} J_F - \frac{r_{GN}}{r_T} J_F \right) T_C - \frac{r_{GN}}{z} J_{PS} T_B - r_{GN} r_D J_F T_{DS},
\]

\[
J_L \dot{\omega}_S = r_{GN} J_F T_E + \left( \frac{r_{GN}}{z^2} J_E + \left( \frac{r_{GN}}{r_T} - 1 \right) r_{GN} J_F \right) T_C + \frac{r_{GN}}{z} J_E T_B - \left( r_D r_{GN} J_F + \frac{r_{GN}^2}{z^2} r_D J_E \right) T_{DS},
\]

\[
J_V \dot{\omega}_V = T_{DS} - T_V,
\]

with,

\[
J_L = J_F J_E + \frac{r_{GN}^2}{z^2} J_{PS} J_F + \left( \frac{r_{GN}}{z} \right)^2 J_{PS} J_E,
\]

\[
r_{GN} = \frac{z r_P}{z + 1}.
\]

The equations (10.13) to (10.15) are used to program the HYSDEL model for the Hybrid Toolbox, the models for both parts of the shift are available in Appendix K.1. The goal of the controller is to have as less variation and oscillation in the driveshaft torque as is possible. From that point of view two different situations have been identified for an upshift in general:

- **Constant driveshaft torque**: the situation where it is possible to have the same driveshaft torque after the shift (e.g. third gear) as before the shift (e.g. second gear).

- **Decreasing driveshaft torque**: the situation where the driveshaft torque before the shift (e.g. second gear) is higher than the maximum driveshaft torque possible after the shift (e.g. third gear).

![Figure 10.4: Traction diagram: Maximum driveshaft torque at each speed for each gear](image-url)
CHAPTER 10. EXPLICIT MODEL PREDICTIVE CONTROL

For the upshift from first to second gear, the coordinated controller for decreasing driveshaft torque has been designed. To be able to identify what the preferred driveshaft torque trajectory is during a shift, a traction diagram is used, which is shown in Figure 10.4.

As the first case, from this map a gear shift from second to third is chosen, in which a constant driveshaft torque is possible; a shift from a driveshaft torque of 600 Nm and a rotational driveshaft speed of 75 rad/sec in second gear, to the same point in third gear. The weight factors (according to equation (10.2) and control horizon for the first phase are, respectively,

\[
\theta = 3, \quad \kappa = 50, \quad \nu = 1, \quad \xi = 100, \\
N = 10, \quad \text{with} \quad t_s = 0.01 \text{ sec}.
\]  

(10.18)

The weight factors (according to equation (10.3) and control horizon for the second phase are, respectively,

\[
\theta = 3, \quad \kappa = 100, \quad \nu = 1, \\
N = 10, \quad \text{with} \quad t_s = 0.01 \text{ sec}.
\]  

(10.19)

The other parameters for the controllers of both parts of the shift are available in Appendix K.2.1. The results are shown in Figure 10.5.

Figure 10.5: Simulation results, hybrid controller B-IST 2-3 upshift, constant driveshaft torque

The controller of the first part is active for 0.5 seconds and it can be clearly seen that the decreasing clutch normal force is compensated by the flywheel brake and engine torque to ensure a constant driveshaft torque of 600 Nm. In the second part the big objective is the clutch engagement whilst ensuring constant driveshaft torque. The controller is benevolent to bring the flywheel brake force down as this input is penalized, but on the other hand using the brake more causes a faster clutch engagement. And above that there is the constraint that the flywheel brake may not be engaged when the rotational velocity of the flywheel changes sign. All these criteria can be distilled from the input trajectories which the controller calculates. Finally it can be seen that the constraint of the minimum of zero relative velocity results in a low relative acceleration at lockup.

The case of decreasing driveshaft torque when upshifting from second to third, occurs typically when rapid acceleration is asked by the driver. In this situation the engine is accelerated to give maximum torque in second gear, but when shifting to third gear it can be seen from Figure 10.4.
that this driveshaft torque cannot be sustained. For the case a hybrid controller is designed for a shift from a driveshaft torque of 800 Nm and a rotational driveshaft speed of 75 rad/sec in second gear to a driveshaft torque of 600 Nm with about the same driveshaft speed in third gear. The weight factors (according to equation (10.2)) and control horizon for the first phase are, respectively,

\[
\theta = 3, \quad \kappa = 50, \quad \nu = 1, \quad \xi = 100, \\
N = 10, \quad \text{with} \quad t_s = 0.01 \text{ sec.}
\]

The weight factors (according to equation (10.3)) and control horizon for the second phase are, respectively,

\[
\theta = 5, \quad \kappa = 50, \quad \nu = 2, \\
N = 10, \quad \text{with} \quad t_s = 0.01 \text{ sec.}
\]

The other parameters and constraints for the controllers of both parts of the shift are available in Appendix K.2.2. The results are shown in Figure 10.6. It can be seen that for the decreasing driveshaft torque the control action globally looks the same as for the case with the constant driveshaft torque. The driveshaft torque has to track a trajectory that brings it down as smoothly as possible.

For the upshift from first to second gear with decreasing driveshaft torque, a first gear state with a driveshaft torque and speed of 1600 Nm and 40 rad/sec, respectively, has been taken. The second gear state is chosen to have a driveshaft torque of 850 Nm and approximately the same driveshaft speed. For a first to second gear upshift the flywheel velocity does not change sign, so no extra precautions have to be taken. The weight factors (according to equation (10.2)) and control horizon for the first phase are, respectively,

\[
\theta = 0, \quad \kappa = 50, \quad \nu = 1, \quad \xi = 100, \\
N = 10, \quad \text{with} \quad t_s = 0.01 \text{ sec.}
\]

The weight factors (according to equation (10.3)) and control horizon for the second phase are, respectively,

\[
\theta = 1, \quad \kappa = 50, \quad \nu = 20, \\
N = 10, \quad \text{with} \quad t_s = 0.01 \text{ sec.}
\]

In this case there is no reference flywheel speed imposed, because in this shift from first to second gear the flywheel speed does not change sign. The extra attention to the flywheel speed to prevent it from passing any stick/slip transitions, as done with the shift from second to third gear, is not necessary here. The other parameters and constraints for the controllers of both parts of the shift are available in Appendix K.3.1 The results are shown in Figure 10.7

### 10.6 Implementation Issues

#### 10.6.1 General considerations

In the previous two sections an implicit MPC controller has been tuned. The next step is to make this controller explicit within a certain range of the parameter space. This done with the EXPCON tool in the Hybrid Toolbox. With this tool the MPC problem is rewritten to a multi-parametric Quadratic Programming problem,

\[
\min \frac{1}{2}z'Qz + \zeta'C'z \\
\text{s.t.} \quad Gz \leq W + S\zeta,
\]

and subsequently solved with the help of the unique mp-QP solver available in the Hybrid Toolbox. The parameter vector \(\zeta\) consists of the states, reference signals, previous control inputs and measured disturbances. The free vector \(z\) consists of the free inputs in the optimal control problem. The way this algorithm globally works is as follows [Bemporad et al., 2002a]: for a random
Figure 10.6: Simulation results, hybrid controller B-IST 2-3 upshift, decreasing driveshaft torque

Figure 10.7: Simulation results, hybrid controller B-IST 1-2 upshift, decreasing driveshaft torque
10.6. IMPLEMENTATION ISSUES

initial parameter vector the region inside the constrained state space in which the parameters remain constant is calculated. After that the parameter sets which are active across each separate boundary of this first region are determined. Then the boundaries of the new parameter sets are determined. This is repeated until the whole constrained parameter space is covered. Then it is tried to bring the number of regions down, duplicates can be easily eliminated by recognizing regions where the combination of active constraints is the same and neighboring cells are tried to be united. The number of regions in the solution to the mp-QP problem depends on the dimensions of $\zeta$ and $z$, the number of constraints $q$ in \[10.24\] and the size of the constrained parameter space. The dependence of the number of regions on the number $q$ of constraints is exponential. The number of constraints $q$ increases with the length of the control horizon (multiplied by the number of constrained control inputs) for input constraints. For output constraints, the number $q$ of constraints increases also with the length $p$ of the prediction horizon (multiplied by the number of constrained outputs). For each region then the value function is evaluated and an optimal explicit control law derived. In [Bemporad et al., 2002a] it is proven that the combination of control laws of the entire constrained parameter space is continuous and piecewise affine.

The following points should be kept in mind while tuning an MPC controller:

- For a long enough prediction horizon $N$, when the constraints are not active the MPC controller performs as LQ control. Therefore, the same noise rejection/bandwidth/sensitivity considerations apply; the larger the ratio between output weight and input weight, the more aggressive the controller, but the larger the sensitivity to noise. Short prediction horizons $N$ also make the controller more aggressive.

- The larger the input horizon $N_u$, the better the performance, because the number of degrees of freedom is larger. On the other hand, the complexity increases with $N_u$. Always choose $N_u$ as small as possible, by progressively reducing it until performance remains acceptable.

- First tune an implicit MPC controller (either using LINCON or HYBCON). When satisfied with the design, compute the explicit form using EXPCON. If you get too many regions, revise the MPC design.

- Be thrifty with constraints. They may provide an excellent design, but later generate too many regions in the explicit MPC partition.

For HYSDEL models the complexity of the explicit controller also depends on the number of binary states (stemming from hybrid systems), inputs and auxiliary variables involved in the dynamics. The use of long optimization horizons may therefore lead to very complex partitions. Most of the time a trade-off has to be made to achieve a lower number partitions at the cost of a lower optimality.

10.6.2 Explicit MPC control law for linear clutch model

As a final encore, the explicit MPC control law for a linear clutch model will be derived in this section. In all the previous cases hybrid models were used for the control synthesis (HYBCON), now a linear clutch model will be used, because this leads to the least complex controller (LINCON), and thus the least amount of regions of the control law. The clutch model is represented in Figure 2.1. The equations of motion are defined to be,

\[
\begin{align*}
J_E \dot{\omega}_E &= T_E - T_C - b_E \omega_E, \\
J_V \dot{\omega}_V &= T_C - b_V \omega_V,
\end{align*}
\]

where the subscript 1 is replaced with $E$ and the subscript 2 is replaced with $V$. Furthermore, an extra, small dissipation term is added to the first equation of \[10.25\], in comparison with the earlier equations of motion \[2.1\]. The load torque in the second equation of \[10.25\] is defined.
as a linear dependence on the rotational velocity. When in system (10.25) the engine torque is modelled as a step disturbance, this leads to the following state space representation,

\[
\dot{x} = \begin{bmatrix}
-\frac{b_E}{J_E} & 0 & \frac{1}{J_E} \\
-\frac{b_V}{J_E} + \frac{b_V}{J_V} & -\frac{b_V}{J_V} & 0 \\
0 & 0 & 0 \\
\end{bmatrix} x + \begin{bmatrix}
-\mu \frac{1}{J_E} \\
-\mu \frac{1}{J_V} + \frac{1}{J_E} \\
0 \\
\end{bmatrix} u,
\]

\[y = \begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
\end{bmatrix} x,
\]

(10.26)

where \(x = [\omega_E, \omega_E - \omega_V, T_E]'\) and \(u = F_N\). Note that the relative rotational velocity, \(\omega_E - \omega_V = \gamma\), has been considered as a state variable instead of the vehicle speed, since the main goal of the controller will be to ensure a suitable profile to this variable thus guaranteeing a smooth engagement process. The cost function is defined as,

\[J = \sum_{k=1}^{N} \left( \theta (\omega_E - \omega_{Eref})^2 + \kappa (\gamma - \gamma_{ref})^2 \right).\]

(10.27)

The weight factors, \(\theta\) and \(\kappa\), the optimal control interval over which the cost function is summed, \(N\), the number of free optimal control moves, \(N_u\), and the number of intervals upon which the output and input constraints are checked, respectively \(N_{cy}\) and \(N_{cu}\), are determined as follows,

\[\theta = 1, \quad \kappa = 20, \quad N = 10, \quad N_u = 1, \quad N_{cy} = 2, \quad N_{cu} = 1.\]

(10.28)

The other parameters and constraints for this controller can be read in Appendix L.1. For a case where the reference engine rotational velocity is set to 120 rad/sec and the engine torque is set to 120 Nm, the results are shown in Figure 10.8.

![State variables](image)

**Figure 10.8:** Simulation results linear clutch model

Now this \texttt{LINCON} controller is converted to a piecewise affine form within a constrained state space, with the help of the Hybrid Toolbox. This leads to a total of 21 regions for the entire
10.6. IMPLEMENTATION ISSUES

parameter space. For the case of $\omega_{Eref} = 120$ rad/sec, $\gamma_{ref} = 0$ and $T_E = 120$ the control law for the parameter space of the remaining elements of $\zeta$ in (10.24) (i.e. $\omega_E$ and $\gamma$) is divided in six regions, which are shown in Figure 10.9 where for clarity reasons the state space is converted from $(\omega_E, \gamma)$ to $(\omega_E, \omega_V)$. In Appendix L.2 the controller is available in explicit form.

![Figure 10.9: Partition of the $(\omega_E, \omega_V)$-plane associated with the explicit controller](image)

10.6.3 Considerations for drivetrain control of the B-IST

Besides the general considerations mentioned in the previous section there are also some specific implementation issues for the drivetrain control of B-IST. Because the explicit MPC and mp-QP solving is a very new methodology its full potential is not yet deployed, still it has already been applied to numerous automotive applications. Now take in mind again Figure 8.1. Instead of designing one controller for the combined supervisory and coordinated control, this has to be split up in different control problems because the resulting controller would be too complex. First for all shifts one or two MPC controllers have to be tuned as is done for the upshift from second to third gear in Section 10.5. After that supervisory control has to be developed which can also be done efficiently and optimal by the hybrid toolbox, taking [Bemporad et al., 2003] as an example where receding horizon optimal control has been applied for supervising an automotive robotized gearbox. The Supervisor decides when and which shift is executed and the Coordinated Controller determining the accompanying explicit MPC control law. For the Strategic Control the psychology of a driver has to be studied more in depth, e.g. how to quantify comfortable driving against for instance sportive driving.
Chapter 11
Conclusions and Recommendations

11.1 Conclusions

Coordinated control of automotive transmissions is becoming more and more important due to the increasing complexity of these transmissions. The heuristic control approach, which is mostly used in the past, will in general not lead to the most optimal control law anymore because of two reasons. Firstly, the number of inputs in advanced transmissions is increasing; extra components like a flywheel with brake clutch lead to situations where three inputs simultaneously have to be determined to ensure an optimal output. Secondly the requirements for the transmissions are increasing; fuel emissions have to be absolutely minimal, the driveability of the car has to improved in every new model of a car. As these requirements are often conflicting, this asks for a more sophisticated control approach than those based on heuristics.

In this thesis a few approaches are introduced along with their advantages and disadvantages. The numerical DP algorithm designed in this thesis, works perfectly on deterministic systems. But for automotive applications a control approach without a feedback strategy is not recommendable. However the numerical DP strategy can also be implemented with feedback, but this results in extremely large lookup-tables because of Bellman’s ‘Curse of Dimensionality’ [Bellman, 1957]. The deterministic approach of numerical DP has been implemented in dry clutch engagement simulations.

The analytic DP approach leads to an explicit feedforward and feedback control law [Haj-Fraj and Pfeiffer, 2001], but no explicit constraints can be set and constraint violation has to be prevented by adding weighted criteria to the cost function. Also the cost function has to be differentiable and the system to be controlled has to be a linear one. The Explicit MPC approach, available in the Hybrid Toolbox for Matlab and Matlab/Simulink [Bemporad, 2004], leads to a piecewise affine explicit control law, where explicit constraints can be set. This approach can derive an optimal control law for both linear and hybrid systems. The hybrid systems can be very flexibly defined in the high level modelling language HYSDEL, also available in the Hybrid Toolbox. This approach has been implemented on dry clutch engagement simulations as well, but also on the coordinated control of a gear shift operation in the B-IST system.

From the analysis of the different approaches, the Hybrid Toolbox leads to the best results because of its ability to both incorporate explicit constraints and to result in an explicit optimal control law. This makes it possible to implement it on standard automotive hardware. Also this methodology has already been applied to numerous automotive applications; dry-clutch engagement [Bemporad et al., 2001a], traction control [Bemporad et al., 2001b], gearbox control [Bemporad et al., 2003] and engine control [Bemporad et al., 2002b].
11.2 Recommendations

The theory of DP is well-established in the meanwhile and has been implemented on numerous applications. The application of analytical DP on gear shift operations, as done in [Haj-Fraj and Pleiffer, 2001], is to the knowledge of the author quite unique, with the only drawback that constraint violation has to be avoided at the cost of performance. Explicit MPC emerges as the method with the most potential. Further recommended research is twofold. Firstly, the implementation of the method on a real-life setup would be the logical next step for this research. Secondly, the method itself of obtaining an explicit control law from a MPC controller could be further optimized. The fact that the complexity of the control law rapidly grows very large, is something that should be addressed. Also it is recommended to investigate the possibility to incorporate adaptive control within this method. This could for instance be realized by adding parameters of the model to the parameter vector of the mp-QP problem and specifying a certain range in which these model parameters can vary.
Part III

Bibliography and Appendices
Bibliography


[Pfeiffer et al., 2005] Pfeiffer, F.G., Foerg, M., Ulbrich, H., "Numerical Aspects of non-smooth Multibody Dynamics", to be published, 2005


Appendix A

List of Abbreviations and Symbols

A.1 Abbreviations

B-IST  Brake- Impulse Shift Transmission
DAE   Differential-Algebraic Equation
DHA   Discrete Hybrid Automata
DP    Dynamic Programming
DTI   DriveTrain Innovations
EG    Event Generator
ETHZ  Eidgenôssische Technische Hochschule Zürich
FSM   Finite State Machine
HYSDEL HYbrid Systems DEscriptive Language
LCP   Linear Complementarity Problem
MLD   Mixed Logical Dynamical
mp-QP  multi-parametric Quadratic Programming
MPC   Model Predictive Control
MS    Mode Selector
ODE   Ordinary Differential Equation
PWA   PieceWise Affine
SAS   Switched Affine System
SISO  Single Input Single Output
TU/e  Eindhoven University of Technology
### A.2 Symbols

<table>
<thead>
<tr>
<th>Greek</th>
<th>Definition</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>α</td>
<td>Switch Model parameter</td>
<td>[-]</td>
</tr>
<tr>
<td>γ</td>
<td>relative rotational velocity</td>
<td>[rad/s]</td>
</tr>
<tr>
<td>γ̇</td>
<td>relative rotational acceleration</td>
<td>[rad/s²]</td>
</tr>
<tr>
<td>ζ</td>
<td>parameter vector in mp-QP problem</td>
<td>n/a</td>
</tr>
<tr>
<td>η</td>
<td>stick band</td>
<td>[-]</td>
</tr>
<tr>
<td>θ</td>
<td>criterium weight</td>
<td>[-]</td>
</tr>
<tr>
<td>κ</td>
<td>criterium weight</td>
<td>[-]</td>
</tr>
<tr>
<td>λ</td>
<td>friction force/torque</td>
<td>[N]/[Nm]</td>
</tr>
<tr>
<td>Λ</td>
<td>impulsive friction force/torque</td>
<td>[Ns]/[Nms]</td>
</tr>
<tr>
<td>λ₀</td>
<td>friction saturation</td>
<td>[N]/[Nm]</td>
</tr>
<tr>
<td>µ</td>
<td>friction coefficient</td>
<td>[-]</td>
</tr>
<tr>
<td>ν</td>
<td>criterium weight</td>
<td>[-]</td>
</tr>
<tr>
<td>ξ</td>
<td>criterium weight</td>
<td>[-]</td>
</tr>
<tr>
<td>τ</td>
<td>time constant</td>
<td>[s]</td>
</tr>
<tr>
<td>φ</td>
<td>rotational angle</td>
<td>[rad]</td>
</tr>
<tr>
<td>ω</td>
<td>rotational velocity</td>
<td>[rad/s]</td>
</tr>
<tr>
<td>ω̇</td>
<td>rotational acceleration</td>
<td>[rad/s²]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Roman</th>
<th>Definition</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>convex set</td>
<td>n/a</td>
</tr>
<tr>
<td>D</td>
<td>viscous damping matrix</td>
<td>[Nms/rad]</td>
</tr>
<tr>
<td>F</td>
<td>force</td>
<td>[N]</td>
</tr>
<tr>
<td>g</td>
<td>relative velocity</td>
<td>[m/s]</td>
</tr>
<tr>
<td>I</td>
<td>contact elements set</td>
<td>n/a</td>
</tr>
<tr>
<td>J</td>
<td>criteria cost function</td>
<td>n/a</td>
</tr>
<tr>
<td>J₀</td>
<td>Inertia</td>
<td>[kgm²]</td>
</tr>
<tr>
<td>K</td>
<td>stiffness matrix</td>
<td>[Nm/rad]</td>
</tr>
<tr>
<td>M</td>
<td>mass matrix</td>
<td>[kgm²]</td>
</tr>
<tr>
<td>p</td>
<td>regularization steepness parameter</td>
<td>[-]</td>
</tr>
<tr>
<td>q</td>
<td>generalized displacements</td>
<td>[m]/[rad]</td>
</tr>
<tr>
<td>˙q, u</td>
<td>generalized velocities</td>
<td>[m/s]/[rad/s]</td>
</tr>
<tr>
<td>˙q̇</td>
<td>generalized accelerations</td>
<td>[m/s²]/[rad/s²]</td>
</tr>
<tr>
<td>r</td>
<td>gear ratio</td>
<td>[-]</td>
</tr>
<tr>
<td>rₘ</td>
<td>mean radius</td>
<td>[m]</td>
</tr>
<tr>
<td>t</td>
<td>time</td>
<td>[s]</td>
</tr>
<tr>
<td>w</td>
<td>constraint vector</td>
<td>[-]</td>
</tr>
</tbody>
</table>
Appendix B

S-function: Switch Model

function [sys,x0,str,ts] = sfundualclutchSWITCH(t,x,u,flag)

switch flag,

%%%%%%%%%%%%%%%%%%
% Initialization %
%%%%%%%%%%%%%%%%%%

case 0,
    [sys,x0,str,ts]=mdlInitializeSizes;

%%%%%%%%%%%%%%%%%%
% Derivatives %
%%%%%%%%%%%%%%%%%%

case 1,
    sys=mdlDerivatives(t,x,u);

%%%%%%%%%%%%%%%%%%
% Outputs %
%%%%%%%%%%%%%%%%%%

case 3,
    sys=mdlOutputs(t,x,u);

case {2, 4, 9 }
    sys = []; % Unused flags

%%%%%%%%%%%%%%%%%%
% Unexpected flags %
%%%%%%%%%%%%%%%%%%

otherwise
    error(['Unhandled flag = ',num2str(flag)]);
end

% end sfuntmpl

% %=============================================================================
% mdlInitializeSizes
% Return the sizes, initial conditions, and sample times for the S-function.
%=============================================================================%
% function [sys,x0,str,ts]=mdlInitializeSizes

global Je Jc Jf Jv rf rc rd mu_c mu_f k w_f w_c tau

Je=0.2; Jc=0.01; Jf=0.01; Jv=110; rf=0.6; rc=0.4; rd=0.2;
mu_c=1; mu_f=1;
k=6000;

w_f = [0 -1 0 1 0 0]; w_c = [0 -1 0 rf/rc 0 0]; tau = 100;

% call simsizes for a sizes structure, fill it in and convert it to a % sizes array.
sizes = simsizes;

sizes.NumContStates = 6; sizes.NumDiscStates = 0;
sizes.NumOutputs = 5; sizes.NumInputs = 3;
sizes.DirFeedthrough = 1;
sizes.NumSampleTimes = 1;   % at least one sample time is needed

sys = simsizes(sizes);

%
% initialize the initial conditions
%

x0 = [0;150;0;149.995;0;18];

%
% str is always an empty matrix
%
str = [];

%
% initialize the array of sample times
%

ts = [0 0];

% end mdlInitializeSizes

%=============================================================================%
% mdlDerivatives
% Return the derivatives for the continuous states.
%=============================================================================%
% function sys=mdlDerivatives(t,x,u)

global Je Jc Jf Jv rf rc rd mu_c mu_f k w_f w_c tau
\[ \text{fn}_f = \text{u}(1); \ \text{fn}_c = \text{u}(2); \ \text{te} = \text{u}(3); \]

\[ \text{tv} = \text{x}(6) \times 20; \quad \% \text{Motion Resistances} \]

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% SWITCH specific
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

\[ \text{tk} = k \times (\text{x}(3) \times \text{rf} \times \text{rd} - \text{x}(5)); \]

if abs(x(4) - x(2)) < 1e-2
    \[ \text{tc} = \text{mc} \times \text{fn}_c; \]
    \[ \text{tf}_\text{karnopp} = ((\text{te} - \text{tf}) \times (\text{jf} + \text{jc} \times \text{rf}^2 / \text{rc}^2) + (\text{tk} \times \text{rd} - \text{tc} / \text{rc}) \times \text{je}) / (\text{je} + \text{jf} + \text{jc} \times \text{rf}^2 / \text{rc}^2); \]
    if abs(Tf_karnopp) < \text{mu}_f \times \text{fn}_f
        \[ \text{tf}_\text{min} = \text{mu}_f \times \text{fn}_f; \quad \% \text{ Tf with negative relative velocity} \]
        \[ \text{tf}_\text{plus} = -\text{mu}_f \times \text{fn}_f; \quad \% \text{ Tf with positive relative velocity} \]
        \[ \text{f}_\text{min} = [\text{x}(2); (\text{te} - \text{tf}_\text{min} - \text{tc}) / \text{je}; \text{x}(4); \]
            \[ (\text{tf}_\text{min} + \text{tc} / \text{rc} - \text{tk} \times \text{rd}) / (\text{jf} + \text{jc} \times \text{rf}^2 / \text{rc}^2); \text{x}(6); (\text{tk} - \text{tv}) / \text{jv}] \]
        \[ \text{f}_\text{plus} = [\text{x}(2); (\text{te} - \text{tf}_\text{plus} - \text{tc}) / \text{je}; \text{x}(4); \]
            \[ (\text{tf}_\text{plus} + \text{tc} / \text{rc} - \text{tk} \times \text{rd}) / (\text{jf} + \text{jc} \times \text{rf}^2 / \text{rc}^2); \text{x}(6); (\text{tk} - \text{tv}) / \text{jv}] \]
        \[ \text{alfa} = (\text{w}_f \times \text{f}_\text{min} + \tau \times \text{w}_f \times \text{x}) / (\text{w}_f \times (\text{f}_\text{min} - \text{f}_\text{plus})) ; \]
        \[ \text{sys} = \text{alfa} \times \text{f}_\text{plus} + (1 - \text{alfa}) \times \text{f}_\text{min} ; \]
    else
        \[ \text{tf} = \text{mu}_f \times \text{fn}_f \times \text{sign}(\text{tf}_\text{karnopp}); \]
        \[ \text{sys} = [\text{x}(2); (\text{te} - \text{tf}) / \text{je}; \text{x}(4); \]
            \[ (\text{tf} + \text{tc} / \text{rc} - \text{tk} \times \text{rd}) / (\text{jf} + \text{jc} \times \text{rf}^2 / \text{rc}^2); \text{x}(6); (\text{tk} - \text{tv}) / \text{jv}] \]
    end
elseif abs(x(4) \times \text{rf} / \text{rc} - x(2)) < 1e-2
    \[ \text{tf} = \text{mu}_f \times \text{fn}_f; \]
    \[ \text{tc}_\text{karnopp} = ((\text{te} - \text{tf}) \times (\text{jf} \times \text{rc} + \text{jc} \times \text{rf} / \text{rc}) \]
        \[ + (\text{tk} \times \text{rd} - \text{tf}) \times \text{je}) / (\text{je} \times \text{rf} / \text{rc} + \text{jf} \times \text{rc} + \text{jc} \times \text{rf} / \text{rc}); \]
    if abs(Tc_karnopp) < \text{mc} \times \text{fn}_c
        \[ \text{tc}_\text{min} = \text{mc} \times \text{fn}_c; \quad \% \text{ Tf with negative relative velocity} \]
        \[ \text{tc}_\text{plus} = -\text{mc} \times \text{fn}_c; \quad \% \text{ Tf with positive relative velocity} \]
        \[ \text{f}_\text{min} = [\text{x}(2); (\text{te} - \text{tf}_\text{min} - \text{tc}_\text{min}) / \text{je}; \text{x}(4); \]
            \[ (\text{tf}_\text{min} + \text{tc}_\text{min} \times \text{rc} / \text{rf} - \text{tk} \times \text{rd}) / (\text{jf} + \text{jc} \times \text{rf}^2 / \text{rc}^2); \text{x}(6); (\text{tk} - \text{tv}) / \text{jv}] \]
        \[ \text{f}_\text{plus} = [\text{x}(2); (\text{te} - \text{tf}_\text{plus} - \text{tc}_\text{plus}) / \text{je}; \text{x}(4); \]
            \[ (\text{tf}_\text{plus} + \text{tc}_\text{plus} \times \text{rc} / \text{rf} - \text{tk} \times \text{rd}) / (\text{jf} + \text{jc} \times \text{rf}^2 / \text{rc}^2); \text{x}(6); (\text{tk} - \text{tv}) / \text{jv}] \]
        \[ \text{alfa} = (\text{w}_c \times \text{f}_\text{min} + \tau \times \text{w}_c \times \text{x}) / (\text{w}_c \times (\text{f}_\text{min} - \text{f}_\text{plus})) ; \]
        \[ \text{sys} = \text{alfa} \times \text{f}_\text{plus} + (1 - \text{alfa}) \times \text{f}_\text{min} ; \]
    else
        \[ \text{tf} = \text{mu}_f \times \text{fn}_f \times \text{sign}(\text{tf}_\text{karnopp}); \]
        \[ \text{sys} = [\text{x}(2); (\text{te} - \text{tf}) / \text{je}; \text{x}(4); \]
            \[ (\text{tf} + \text{tc} \times \text{rc} - \text{tk} \times \text{rd}) / (\text{jf} + \text{jc} \times \text{rf}^2 / \text{rc}^2); \text{x}(6); (\text{tk} - \text{tv}) / \text{jv}] \]
    end
end
else
    Tc=mu_c*Fn_c*sign(Tc_karnopp);
    sys = [x(2);
    (Te-Tf-Tc)/Je;
    x(4);
    (Tf+Tc*rf/rc-Tk*rf*rd)/(Jf+Jc*rf^2/rc^2);
    x(6);
    (Tk-TV)/Jv];
else
    Tf=sign(x(2)-x(4))*mu_f*Fn_f;
    Tc=sign(x(2)-x(4)*rf/rc)*mu_c*Fn_c;
    sys = [x(2);
    (Te-Tf-Tc)/Je;
    x(4);
    (Tf+Tc*rf/rc-Tk*rf*rd)/(Jf+Jc*rf^2/rc^2);
    x(6);
    (Tk-TV)/Jv];
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% end SWITCH specific
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% end mdlDerivatives
%=============================================================================
% mdlOutputs
% Return the block outputs.
%=============================================================================
% function sys=mdlOutputs(t,x,u)

global rf rc rd

sys = [x(3)*rf*rd-x(5);x(2);x(4);x(4)*rf/rc;x(6)];

% end mdlOutputs
Appendix C

Analysis of initial decomposition contact law

In the specific case of decomposing a tangential friction characteristic (on the acceleration level) into a LCP formulation, as is done by Pfeiffer and Glocker in [Pfeiffer and Glocker, 1996], it is worthwhile to examine the equivalence between this friction characteristic and its decomposed counterpart written in LCP-formulation. Because in Pfeiffer’s book one tangential friction characteristic is decomposed into a fourth order LCP. A fourth order LCP has sixteen $2^4$ different modes, but looking at the characteristic, shown in Figure C.1, it physically shows only three modes, where a mode is defined to be a interconnected part of the characteristic where one quantity is a constant and the other varies.

![Figure C.1: Three modes in a potentially sticking contact](image)

In order to thoroughly examine the cause of this increase in modes, first will all sixteen modes of the LCP will be elaborated and it will be shown if and why modes are abundant or to which part of the friction characteristic they correspond.

It can be read in Section 3.3.1 that the decomposition of the friction characteristic leads to four
pairs of inequalities and complementary conditions:

\[
\begin{align*}
\dot{\gamma}_i^+ & \geq 0; & \lambda_T^{(+)} & \geq 0; & \dot{\gamma}_i^- & \lambda_T^{(+)} = 0 \\
\dot{\gamma}_i^- & \geq 0; & \lambda_T^{(-)} & \geq 0; & \dot{\gamma}_i^+ & \lambda_T^{(-)} = 0 \\
\lambda_{T_{0;i}}^{(-)} & \geq 0; & z_i^- & \geq 0; & \lambda_{T_{0;i}}^{(+)} & z_i^- = 0 \\
\lambda_{T_{0;i}}^{(+)} & \geq 0; & z_i^+ & \geq 0; & \lambda_{T_{0;i}}^{(-)} & z_i^+ = 0 \\
\end{align*}
\]  

(C.1)

For clarity sake I will state the LCP from section 3.3.1 here again:

\[
\begin{pmatrix}
\dot{\gamma}_D \\
\lambda_{T_{0;D}} \\
y
\end{pmatrix}
= 
\begin{pmatrix}
R^T T M^{-1} W & I & 0 \\
-I & 0 & 0 \\
A & x
\end{pmatrix}
+ 
\begin{pmatrix}
R^T T M^{-1} (Dq + Kq + h + W_S \lambda_S) \\
M_T \\
b
\end{pmatrix}
, \quad (C.2)
\]

where,

\[
\begin{align*}
\begin{pmatrix}
\dot{\gamma}_D \\
\lambda_{T_{0;i}}^{(+)} \\
\lambda_{T_{0;i}}^{(-)}
\end{pmatrix}
g & \geq 0; & \begin{pmatrix}
\lambda_T^{(+)} \\
\lambda_T^{(-)}
\end{pmatrix}
& \geq 0; & \begin{pmatrix}
\dot{\gamma}_D \\
\lambda_{T_{0;i}}^{(+)} \\
\lambda_{T_{0;i}}^{(-)}
\end{pmatrix}
& \lambda_{T_{0;i}}^{(+)} = 0 \\
\lambda_{T_{0;i}}^{(-)} & \geq 0; & z_i^- & \geq 0; & \lambda_{T_{0;i}}^{(+)} & z_i^- = 0 \\
\lambda_{T_{0;i}}^{(+)} & \geq 0; & z_i^+ & \geq 0; & \lambda_{T_{0;i}}^{(-)} & z_i^+ = 0 \\
\end{align*}
\]  

(C.3)

From the definition of this LCP-formulation it turns out that this LCP has a standard form with of course for every different system also different values. But the standard form can be defined as follows:

\[
\begin{pmatrix}
\dot{\gamma}_i^+ \\
\lambda_T^{(+)} \\
\lambda_{T_{0;i}}^{(-)}
\end{pmatrix}
= 
\begin{pmatrix}
-M & M & 1 & 0 \\
M & -M & 0 & 1 \\
-1 & 0 & 0 & 0
\end{pmatrix}
\begin{pmatrix}
\lambda_T^{(+)} \\
\lambda_T^{(-)} \\
z_i^-
\end{pmatrix}
- 
\begin{pmatrix}
q \\
-q \\
M_T \\
M_T
\end{pmatrix}
. \quad (C.5)
\]

In the notation, the subscript \(i\) will be omitted for clarity sake and all possible modes will be written down underneath each other. In the left column are the conditions for each of the variables in the middle the resulting values from (C.5) and on the right the mode number and its relevance.

\[
\begin{align*}
\dot{\gamma}_i^+ & \geq 0; & \lambda_T^{(+)} & = 0 & \Rightarrow & \dot{\gamma}_i^+ & = q; & \lambda_T^{(+)} & = 0 & \text{MODE 1: The conditions are only} \\
\dot{\gamma}_i^- & \geq 0; & \lambda_T^{(-)} & = 0 & \Rightarrow & \dot{\gamma}_i^- & = -q; & \lambda_T^{(-)} & = 0 & \text{met when } q = 0, \text{which generally} \\
\lambda_T^{(-)} & \geq 0; & z_i^- & = 0 & \lambda_T^{(+)} & z_i^- & = 0 & \text{will not occur.} \\
\lambda_T^{(+)} & \geq 0; & z_i^+ & = 0 & \lambda_T^{(-)} & z_i^+ & = 0
\end{align*}
\]

MODE 2: does not satisfy

\[
\begin{align*}
\dot{\gamma}_i^+ & \geq 0; & \lambda_T^{(+)} & = 0 & \Rightarrow & \dot{\gamma}_i^+ & = q - z_i^-; & \lambda_T^{(+)} & = 0 \\
\dot{\gamma}_i^- & \geq 0; & \lambda_T^{(-)} & = 0 & \Rightarrow & \dot{\gamma}_i^- & = z_i^+ - q; & \lambda_T^{(-)} & = 0 \\
\lambda_T^{(-)} & \geq 0; & z_i^- & = 0 & \lambda_T^{(+)} & z_i^- & = 0 & \text{MODE 3: does not satisfy} \\
\lambda_T^{(+)} & = 0; & z_i^+ & \geq 0 & \lambda_T^{(-)} & z_i^+ & = 0 \\
\lambda_T^{(-)} & = 0; & z_i^- & \geq 0 & \lambda_T^{(+)} & z_i^- & = 0
\end{align*}
\]
\[ \dot{\gamma}^+ \geq 0; \quad \lambda_T^{(+)} = 0 \quad \Rightarrow \quad \dot{\gamma}^+ = q + z^-; \quad \lambda_T^{(+)} = 0 \]  
\[ \dot{\gamma}^- \geq 0; \quad \lambda_T^{(-)} = 0 \quad \Rightarrow \quad \dot{\gamma}^- = z^- - q; \quad \lambda_T^{(-)} = 0 \]  
MODE 4: does not satisfy

\[ \lambda_T^{(0)} = 0; \quad z^- \geq 0 \quad \Rightarrow \quad \lambda_T^{(0)} = M_T \neq 0; \quad z^- = \dot{\gamma}^+ + q \]  
\[ \lambda_T^{(0)} = 0; \quad z^+ \geq 0 \quad \Rightarrow \quad \lambda_T^{(0)} = M_T \neq 0; \quad z^+ = \dot{\gamma}^- - q \]  

\[ \dot{\gamma}^+ \geq 0; \quad \lambda_T^{(+)} = 0 \quad \Rightarrow \quad \dot{\gamma}^+ = M\lambda_T^{(-)} + q = 0; \quad \lambda_T^{(+)} = 0 \]  
\[ \dot{\gamma}^- \geq 0; \quad \lambda_T^{(-)} \geq 0 \quad \Rightarrow \quad \dot{\gamma}^- = -M\lambda_T^{(-)} - q = 0; \quad \lambda_T^{(-)} = -\frac{q}{M} \]  
MODE 5: It follows that in this mode, \( \dot{\gamma} = 0 \) and \( \lambda_T = \frac{q}{M} \), with the condition \( -M_T \leq \frac{q}{M} \leq 0 \)

\[ \lambda_T^{(0)} \geq 0; \quad z^- = 0 \quad \Rightarrow \quad \lambda_T^{(0)} = M_T; \quad z^- = 0 \]  
\[ \lambda_T^{(0)} \geq 0; \quad z^+ = 0 \quad \Rightarrow \quad \lambda_T^{(0)} = M_T - \lambda_T^{(-)}; \quad z^+ = 0 \]  

\[ \dot{\gamma}^+ \geq 0; \quad \lambda_T^{(+)} = 0 \quad \Rightarrow \quad \dot{\gamma}^+ = M\lambda_T^{(-)} + z^-; \quad \lambda_T^{(+)} = 0 \]  
\[ \dot{\gamma}^- \geq 0; \quad \lambda_T^{(-)} \geq 0 \quad \Rightarrow \quad \dot{\gamma}^- = -M\lambda_T^{(-)} - z^+ - q = 0; \quad \lambda_T^{(-)} = -\frac{q}{M} \]  
MODE 6: It follows that in this mode, \( \dot{\gamma} = M\lambda_T^{(-)} + q \) and \( \lambda_T = -M_T \)

\[ \lambda_T^{(0)} \geq 0; \quad z^- = 0 \quad \Rightarrow \quad \lambda_T^{(0)} = M_T \neq 0; \quad z^- = \dot{\gamma}^+ - M\lambda_T^{(-)} - q \]  
\[ \lambda_T^{(0)} \geq 0; \quad z^+ = 0 \quad \Rightarrow \quad \lambda_T^{(0)} = M_T - \lambda_T^{(-)}; \quad z^+ = \dot{\gamma}^- + M\lambda_T^{(-)} + q \]

\[ \dot{\gamma}^+ \geq 0; \quad \lambda_T^{(+)} = 0 \quad \Rightarrow \quad \dot{\gamma}^+ = M\lambda_T^{(-)} + z^- + q; \quad \lambda_T^{(+)} = 0 \]  
\[ \dot{\gamma}^- = 0; \quad \lambda_T^{(-)} = 0 \quad \Rightarrow \quad \dot{\gamma}^- = -M\lambda_T^{(-)} - z^+ = 0; \quad \lambda_T^{(-)} = 0 \]  
MODE 7: does not satisfy

\[ \lambda_T^{(0)} \geq 0; \quad z^- = 0 \quad \Rightarrow \quad \lambda_T^{(0)} = M_T \neq 0; \quad z^- = \dot{\gamma}^+ - M\lambda_T^{(-)} - q \]  
\[ \lambda_T^{(0)} \geq 0; \quad z^+ = 0 \quad \Rightarrow \quad \lambda_T^{(0)} = M_T - \lambda_T^{(-)}; \quad z^+ = \dot{\gamma}^- + M\lambda_T^{(-)} + q \]

\[ \dot{\gamma}^+ = 0; \quad \lambda_T^{(+)} \geq 0 \quad \Rightarrow \quad \dot{\gamma}^+ = -M\lambda_T^{(+)} + q = 0; \quad \lambda_T^{(+)} = \frac{q}{M} \]  
\[ \dot{\gamma}^- \geq 0; \quad \lambda_T^{(-)} = 0 \quad \Rightarrow \quad \dot{\gamma}^- = -M\lambda_T^{(-)} - q = 0; \quad \lambda_T^{(-)} = 0 \]  
MODE 8: does not satisfy

\[ \lambda_T^{(0)} \geq 0; \quad z^- = 0 \quad \Rightarrow \quad \lambda_T^{(0)} = M_T \neq 0; \quad z^- = \dot{\gamma}^+ - M\lambda_T^{(-)} - q \]  
\[ \lambda_T^{(0)} \geq 0; \quad z^+ = 0 \quad \Rightarrow \quad \lambda_T^{(0)} = M_T - \lambda_T^{(+)}; \quad z^+ = \dot{\gamma}^- + M\lambda_T^{(+)} + q \]

\[ \dot{\gamma}^+ = 0; \quad \lambda_T^{(+)} \geq 0 \quad \Rightarrow \quad \dot{\gamma}^+ = -M\lambda_T^{(+)} + q = 0; \quad \lambda_T^{(+)} = \frac{q}{M} \]  
\[ \dot{\gamma}^- \geq 0; \quad \lambda_T^{(-)} = 0 \quad \Rightarrow \quad \dot{\gamma}^- = -M\lambda_T^{(-)} - q = 0; \quad \lambda_T^{(-)} = 0 \]  
MODE 9: It follows that in this mode, \( \dot{\gamma} = 0 \) and \( \lambda_T = \frac{q}{M} \), with the condition \( 0 \leq \frac{q}{M} \leq M_T \)

\[ \lambda_T^{(0)} \geq 0; \quad z^- = 0 \quad \Rightarrow \quad \lambda_T^{(0)} = M_T \neq 0; \quad z^- = \dot{\gamma}^+ - M\lambda_T^{(+)} - q \]  
\[ \lambda_T^{(0)} \geq 0; \quad z^+ = 0 \quad \Rightarrow \quad \lambda_T^{(0)} = M_T - \lambda_T^{(+)}; \quad z^+ = \dot{\gamma}^- + M\lambda_T^{(+)} + q \]

\[ \dot{\gamma}^+ \geq 0; \quad \lambda_T^{(+)} \geq 0 \quad \Rightarrow \quad \dot{\gamma}^+ = -M\lambda_T^{(+)} + z^- + q = 0; \quad \lambda_T^{(+)} = \frac{q}{M} \]  
\[ \dot{\gamma}^- \geq 0; \quad \lambda_T^{(-)} = 0 \quad \Rightarrow \quad \dot{\gamma}^- = -M\lambda_T^{(-)} - q = 0; \quad \lambda_T^{(-)} = 0 \]  
MODE 10: does not satisfy

\[ \lambda_T^{(0)} \geq 0; \quad z^- = 0 \quad \Rightarrow \quad \lambda_T^{(0)} = M_T \neq 0; \quad z^- = \dot{\gamma}^+ - M\lambda_T^{(+)} - q \]  
\[ \lambda_T^{(0)} \geq 0; \quad z^+ = 0 \quad \Rightarrow \quad \lambda_T^{(0)} = M_T - \lambda_T^{(+)}; \quad z^+ = \dot{\gamma}^- + M\lambda_T^{(+)} + q \]

\[ \dot{\gamma}^+ \geq 0; \quad \lambda_T^{(+)} \geq 0 \quad \Rightarrow \quad \dot{\gamma}^+ = -M\lambda_T^{(+)} + z^- + q = 0; \quad \lambda_T^{(+)} = \frac{q}{M} \]  
\[ \dot{\gamma}^- \geq 0; \quad \lambda_T^{(-)} = 0 \quad \Rightarrow \quad \dot{\gamma}^- = -M\lambda_T^{(-)} - q = 0; \quad \lambda_T^{(-)} = 0 \]  
MODE 11: It follows that in this mode, \( \dot{\gamma} = -M\lambda_T^{(+)} + q \) and \( \lambda_T = M_T \).
APPENDIX C. ANALYSIS OF INITIAL DECOMPOSITION CONTACT LAW

\[
\begin{align*}
\gamma^+ &= 0; \quad \lambda^{(+)}_T \geq 0 \quad \Rightarrow \quad \hat{\gamma}^+ = -M\lambda^{(+)}_T + z^- + q = 0; \quad \lambda^{(+)}_T = \frac{z^- + q}{M} & \quad \text{MODE 12: does not satisfy} \\
\hat{\gamma}^- &= 0; \quad \lambda^{(-)}_T \geq 0 \quad \Rightarrow \quad \hat{\gamma}^- = M\lambda^{(+)}_T - z^- - q; \quad \lambda^{(-)}_T = 0 \\
\lambda^{(+)}_T &= 0; \quad z^- \geq 0 \quad \Rightarrow \quad \lambda^{(+)}_T = M - \lambda^{(+)}_T = 0; \quad z^- = M\lambda^{(+)}_T - q \\
\lambda^{(-)}_T &= 0; \quad z^+ \geq 0 \quad \Rightarrow \quad \hat{\gamma}^+ = M_T \neq 0; \quad z^+ = \hat{\gamma}^- - M\lambda^{(+)}_T + q & \quad \text{MODE 13: It follows that in this mode, } \hat{\gamma} = 0 \text{ and } \lambda_T = \frac{q}{M}, \text{ with the condition } -M_T \leq \frac{q}{M} \leq M_T \\
\lambda^{(+)}_T &= 0; \quad \lambda^{(-)}_T \geq 0 \quad \Rightarrow \quad \hat{\gamma}^+ = -M\lambda^{(+)}_T + M\lambda^{(-)}_T + z^- + q = 0; \quad \lambda^{(+)}_T = \lambda^{(-)}_T + \frac{q}{M} & \quad \text{MODE 15: It follows that in this mode, } \hat{\gamma} = 0 \text{ and } \lambda_T = \frac{q}{M}, \text{ with the condition } 0 \leq \frac{q}{M} \leq M_T \\
\lambda^{(-)}_T &= 0; \quad z^+ \geq 0 \quad \Rightarrow \quad \hat{\gamma}^- = M\lambda^{(+)}_T - M\lambda^{(-)}_T + z^- - q = 0; \quad \lambda^{(-)}_T = \lambda^{(+)}_T - M_T & \quad \text{MODE 15: It follows that in this mode, } \hat{\gamma} = 0 \text{ and } \lambda_T = \frac{q}{M}, \text{ with the condition } 0 \leq \frac{q}{M} \leq M_T \\
\gamma^+ &= 0; \quad \lambda^{(+)}_T \geq 0 \quad \Rightarrow \quad \hat{\gamma}^+ = -M\lambda^{(+)}_T + M\lambda^{(-)}_T + z^- + q = 0; \quad \lambda^{(+)}_T = \lambda^{(-)}_T + \frac{q}{M} & \quad \text{MODE 16: The conditions are only met when } q = 0, \text{ which generally will not occur.} \\
\gamma^- &= 0; \quad \lambda^{(-)}_T \geq 0 \quad \Rightarrow \quad \hat{\gamma}^- = M\lambda^{(+)}_T - M\lambda^{(-)}_T + z^- - q = 0; \quad \lambda^{(-)}_T = \lambda^{(+)}_T - \frac{q}{M} & \quad \text{MODE 16: The conditions are only met when } q = 0, \text{ which generally will not occur.} \\
\lambda^{(+)}_T &= 0; \quad z^- \geq 0 \quad \Rightarrow \quad \lambda^{(+)}_T = M - \lambda^{(+)}_T = 0; \quad z^- = -q & \quad \text{MODE 16: The conditions are only met when } q = 0, \text{ which generally will not occur.} \\
\lambda^{(-)}_T &= 0; \quad z^+ \geq 0 \quad \Rightarrow \quad \lambda^{(-)}_T = M - \lambda^{(-)}_T = 0; \quad z^+ = q & \quad \text{MODE 16: The conditions are only met when } q = 0, \text{ which generally will not occur.}
\end{align*}
\]

To get some order in the above analysis the modes will be categorized:

- **Impossible**: These are modes that simply will never be a solution to the LCP, because the definitions of the different slack variables lead to conflicts in the complementarity conditions, and thereby render the mode impossible. These are the modes 2, 3, 4, 7, 8, 10 and 12.

- **Quiescency**: These modes only lead to a solution of the LCP if in equation (C.2), the vector \( b = 0 \). This means for instance that there must be a zero resultant external force working on the system. This applies for the modes, 1 and 16.

- **Stick**: These modes represent the solutions to the LCP with the result \( \hat{\gamma} = 0 \). As one can see in the above modes descriptions, the modes overlay each other. While mode 13 covers the whole area of \( \hat{\gamma} = 0 \), the modes 5, 9, 14 and 15 only cover half of the area either the side with positive force or with negative force and thus are abundant.

- **Slip**: These modes represent the solutions to the LCP where \( \hat{\gamma} \neq 0 \). These are modes 6, where \( \hat{\gamma} \geq 0 \) and mode 11, where \( \hat{\gamma} \leq 0 \).

The conclusion of the above is that modes 6, 11 and 13 cover the whole friction characteristic as can be seen in Figure C.1. A specific solution algorithm could be developed which only ‘tests’ these three modes, because they cover the whole solution space. The solution algorithm could work as follows: the value of \( \frac{q}{M} \) is calculated, if this value is between \(-M_T\) and \(M_T\) or equal, the contact...
is in mode 13 and thus stick, $\dot{\gamma} = 0$ and $\lambda_T = \frac{q}{M}$. If $\frac{q}{M} < -M_T$ or $\frac{q}{M} > M_T$ then the system can either fall into negative (mode 11) or positive (mode 6) slip, depending on the definition of $\dot{\gamma}$. This can be determined by testing the following inequalities: For mode 6 this condition is $MM_T \geq q$, if this is true the system is in mode 6. For mode 11 this condition is $MM_T \geq -q$, if this is true the system is in mode 11.
Appendix D

S-function: Event-Driven Integration Method with LCP

function [sys,x0,str,ts] = sfunLCPEVENT(t,x,u,flag)

switch flag,

%%%%%%%%%%%%%%%%%%%
% Initialization %
%%%%%%%%%%%%%%%%%%%

case 0,
    [sys,x0,str,ts]=mdlInitializeSizes;

%%%%%%%%%%%%%%%%%%%
% Derivatives %
%%%%%%%%%%%%%%%%%%%

case 1,
    sys=mdlDerivatives(t,x,u);

%%%%%%%%%%%%%%%%%%%
% Outputs %
%%%%%%%%%%%%%%%%%%%

case 3,
    sys=mdlOutputs(t,x,u);

case {2, 4, 9 }
    sys = []; % Unused flags

%%%%%%%%%%%%%%%%%%%
% Unexpected flags %
%%%%%%%%%%%%%%%%%%%

otherwise
    error(['Unhandled flag = ',num2str(flag)]);
end

%
function [sys,x0,str,ts]=mdlInitializeSizes

global M K rf rc rd mu_c mu_f z0 Wf Wc
Je=0.2; Jc=0.01; Jf=0.01; Jv=110; rf=0.6; rc=0.4; rd=0.2;
mu_c=1; mu_f=1;
k=6000;

M = [Je 0 0; 0 Jf+Jc*rf^2/rc^2 0; 0 0 Jv];
K = [0 0 0; -rf^2*rd^2*k rd*rf*k; 0 rf*rd*k -k];
Wf = [-1; 1; 0];
Wc = [-1 rf/rc 0];
z0=[ ];

sys = simsizes(sizes);

x0 = [0; 150; 0; 150; 0; 18];
str = [ ];
ts = [0 0];
% mdlDerivatives
% Return the derivatives for the continuous states.
%=============================================================================% function sys=mdlDerivatives(t,x,u)

global M K mu_c mu_f Wf Wc z0 rf rc rd

FN_f=u(1); FN_c=u(2); Te=u(3);

Tv=x(6)*20; % Motion Resistances

% LCPEVENT specific
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
h = [Te;0;-Tv];
phi=[x(1);x(3);x(5)]; % The rotation angles / distance of the six inertias

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Arranging all stick/slip contacts in appropriate sets %%%%%%%%%%%%%%%%%%%%%%%%
% is=0; it=0; Wh=[]; Ws=[];
if abs(x(4)-x(2)) < 1e-1 % Clutch F stick condition
    it=it+1;
    Wh(:,it)=Wf;
    fricsat(2*it-1,1)=mu_f*FN_f;
    fricsat(2*it,1)=mu_f*FN_f;
else
    is=is+1;
    Ws(:,is)=Wf;
    Ts(is,1)=sign(x(2)-x(4))*mu_f*FN_f;
end
if abs(x(4)*rf/rc-x(2)) < 1e-1 % Clutch C stick condition
    it=it+1;
    Wh(:,it)=Wc;
    fricsat(2*it-1,1)=mu_c*FN_c;
    fricsat(2*it,1)=mu_c*FN_c;
else
    is=is+1;
    Ws(:,is)=Wc;
    Ts(is,1)=sign(x(2)-x(4)*rf/rc)*mu_c*FN_c;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if isempty(Wh)==0 % Case that set of possibly sticking
    % contacts is nonzero
    % Evaluation of LCP only needed when
    % contacts are in possibly stick situation
    Wt=[Wh -Wh];
% Solving the LCP and determining qdot
if isempty(Ws)==0 % Case that set of sliding contacts is nonzero
    Mc = [Wt'*(M^-1)*Wt eye(2*it);-eye(2*it) zeros(2*it)];
    qc = [Wt'*(M^-1)*(h + K*phi + Ws*Ts); fricsat];
    [uc,err] = lemke(Mc,qc,z0);
    if err~=0
        [uc,err] = lemke(Mc,qc);
    end
    z0=uc;
    Tt=uc(1:max(size(uc))/2,1);
    qdot = (M^-1)*(h + K*phi + Ws*Ts + Wt*Tt);
else % Case that set of sliding contacts is zero
    Mc = [Wt'*(M^-1)*Wt eye(2*it);-eye(2*it) zeros(2*it)];
    qc = [Wt'*(M^-1)*(h + K*phi); fricsat];
    [uc,err,z0] = lemke(Mc,qc,z0);
    if err~=0
        [uc,err,z0] = lemke(Mc,qc);
    end
    Tt=uc(1:max(size(uc))/2,1);
    qdot = (M^-1)*(h + K*phi + Wt*Tt);
end
else % Case that set of possibly sticking contacts is zero
    qdot = (M^-1)*(h + K*phi + Ws*Ts);
end
sys = [x(2);
    qdot(1);
    x(4);
    qdot(2);
    x(6);
    qdot(3)];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% end LCPEVENT specific
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% mdlOutputs
% Return the block outputs.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
global rf rc rd
sys = [x(3)*rf*rd-x(5);x(2);x(4);x(4)*rf/rc;x(6)];
% end mdl0outputs
Appendix E

Lemke’s Algorithm

% LEMKE Solves standard linear complementarity problems (LCPs).
% An LCP solves
% \[ Mz + q \geq 0, \quad z \geq 0, \quad z'(Mz+q)=0. \]
% USAGE
% \[ [z, err] = lemke(M, q, z0); \]
% The input \( z0 \) defines a starting basis; it can be either
% an initial guess of the solution or a vector of zeros and ones
% with ones representing those \( z(i) \) thought to be non-zero in the
% solution. For example, passing \( z=[1.5;0;2.2] \) has the same
% effect as passing \( z=[1;0;1] \).
% If \( z0 \) is omitted the origin is used as a starting basis.
% ERR returns an error condition:
% 0: Solution found
% 1: Maximum iterations exceeded
% 2: Unbounded ray termination
% 3: Initial basis cannot be computed - try new value of \( z0 \)
% If NARGOUT==1, a warning message is displayed instead.
% 
% ALGORITHM
% Uses a modified Lemke’s algorithm (complementary pivoting)
% The algorithm is modified to allow a user defined initial basis.
% If no initial solution vector is supplied, a covering ray of ones
% is used.

% Copyright (c) 1997-2002, Paul L. Fackler & Mario J. Miranda
% paul_fackler@ncsu.edu, miranda.4@osu.edu

function [z, err] = lemke(M, q, z0)

n = length(q);

if nargin<4, ineqind=ones(n,1); end if nargin<3, z0=[]; end

if size(M)~=n n]
    error('Matrices are not compatible');
end

zer_tol = 1e-5; piv_tol = 1e-8; maxiter = min(1000,25*n); err=0;
% Trivial solution exists
if all(q>=0)
    z=zeros(n,1); return;
end

% Initializations
% (note: all variables are initialized to their appropriate sizes)
z = zeros(2*n,1); j=zeros(n,1); iter=0; theta=0; ratio=0;
leaving=0; Be=ones(n,1); x=q; bas=zeros(n,1); nonbas=zeros(n,1);
t = 2*n+1; % Artificial variable
entering=t; % is the first entering variable

% Determine initial basis
if isempty(z0)
    bas=[];
    nonbas=(1:n)';
else
    bas=find(z0>0);
    nonbas=find(z0<=0);
end
B=spalloc(n,n,nnz(M)); % allocate memory for B
B=-speye(n);

% Determine initial values
if ~isempty(bas)
    B=[M(:,bas) B(:,nonbas)];
    if condest(B)>1e16
        z=[]; err=3; return
    end
    x=-(B\q);
end

% Check if initial basis provides solution
if all(x>=0)
    z(bas)=x(1:length(bas));
    z=z(1:n);
    return
end

% Determine initial leaving variable
[tval,lvindex]=max(-x); bas=[bas;(n+nonbas)];
leaving=bas(lvindex);

bas(lvindex)=t; % pivot in the artificial variable

U=x<0;
%U=ones(n,1); % Alternative artificial vector
Be=-B*U; x=x+tval*U; x(lvindex)=tval; B(:,lvindex)=Be;

% Main iterations begin here
for iter=1:maxiter
% Check if done; if not, get new entering variable
if (leaving == t) break
elseif (leaving <= n)
    entering = n+leaving;
    Be=zeros(n,1); Be(leaving)=-1;
else
    entering = leaving - n;
    Be = M(:,entering);
end

d = B\Be;

% Find new leaving variable
j=find(d>piv_tol); % indices of d>0
if isempty(j) % no new pivots - ray termination
    err=2;
    break
end
theta=min((x(j)+zer_tol)./d(j)); % minimal ratios, d>0
j(find((x(j))./d(j))==theta)); % indices of minimal ratios, d>0
lvindex=find(bas(j)==t); % check if artificial among these
if ~isempty(lvindex) % Always use artificial if possible
    lvindex=j(lvindex);
else % otherwise pick among set of max d
    [theta,lvindex]=max(d(j));
    lvindex=find(d(j)==theta);
    lvindex=ceil(length(lvindex)*rand(1,1)); % if multiple choose randomly
    lvindex=j(lvindex);
end

leaving=bas(lvindex);

% Perform pivot
ratio=x(lvindex)./d(lvindex);
x = x - ratio*d;
x(lvindex) = ratio;
B(:,lvindex) = Be;
obas(lvindex) = entering;
end % end of iterations

if iter>=maxiter & leaving~=t err=1; end

z(bas) = x; z = z(1:n);

% Display warning messages if no error code is returned
if nargout<2 & err~=0
    s='Warning: solution not found - ';
    if err=1
        disp([s 'Iterations exceeded limit']);
    elseif err=2
        disp([s 'Unbounded ray']);
    elseif err=3
        disp([s 'Initial basis infeasible']);
    end
end
Appendix F

S-function: Time-Stepping Method with LCP

function [sys,x0,str,ts] = sfundualclutchLCPTS(t,x,u,flag)

switch flag,
    case 0
        [sys,x0,str,ts] = mdlInitializeSizes; % Initialization
    case 2
        sys = mdlUpdate(t,x,u); % Update discrete states
    case 3
        sys = mdlOutputs(t,x,u); % Calculate outputs
    case {1, 4, 9} % Unused flags
        sys = [];
    otherwise
        error(['unhandled flag = ',num2str(flag)]); % Error handling
end
% End of dualclutchTS.

%===============================================================================
% Initialization
%===============================================================================

function [sys,x0,str,ts] = mdlInitializeSizes

global M K mu_c mu_f z0 W R A dt rf rc rd
Je=0.2; Jc=0.01; Jf=0.01; Jv=110; rf=0.6; rc=0.4; rd=0.2;
mu_f=1; mu_c=1;
k=6000;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% LCPTS specific

\[ M = \begin{bmatrix} Je & 0 & 0 & Jf + Jc \cdot rf^2/\text{rc}^2 & 0 & 0 & Jv \end{bmatrix}; \quad W = \begin{bmatrix} -1 & -1 & \text{rf}/\text{rc} & 0 & 0 \end{bmatrix}; \]

\[ A = \begin{bmatrix} W^T \cdot M^{-1} \cdot W \cdot \text{eye}(2) & -\text{eye}(2) & \text{zeros}(2) \end{bmatrix}; \quad K = \begin{bmatrix} 0 & 0 & 0 & -\text{rf}^2 \cdot \text{rd}^2 \cdot k & \text{rd} \cdot \text{rf} \cdot k & 0 \cdot \text{rd} \cdot k \cdot -k \end{bmatrix}; \]

dt = 1e-4; % sample time

% Call simsizes for a sizes structure, fill it in, and convert it
% to a sizes array.

sizes = simsizes; sizes.NumContStates = 0; sizes.NumDiscStates = 6; sizes.NumOutputs = 5; sizes.NumInputs = 3; sizes.DirFeedthrough = 0; sizes.NumSampleTimes = 1; sys = simsizes(sizes);

x0 = [0;150;0;150;0;18]; % Initialize the discrete states;

str = []; % Set str to an empty matrix.

ts = [dt 0]; % sample time: [period, offset]
% End of mdlInitializeSizes.

%==============================================================
% Update the discrete states
%==============================================================

function sys = mdlUpdate(t,x,u)

\[ uA = [x(2);x(4);x(6)]; \quad qA = [x(1);x(3);x(5)]; \]

\[ qM = qA + 0.5 \cdot dt \cdot uA; \quad MuFn = [\mu_f \cdot Fn_f; \mu_c \cdot Fn_c]; \quad h = [Te;0;-Tv]; \]

\[ b = [-W^T \cdot (uA + M^{-1} \cdot (K \cdot qM + h) \cdot dt) - W^T \cdot M^{-1} \cdot W \cdot MuFn \cdot dt; 2 \cdot MuFn \cdot dt]; \]

[xlcp,err]=lemke(A,b,z0); if err~=0
    [xlcp,err] = lemke(A,b);
end z0 = xlcp; ylcp = A*xlcp+b;

T = 0.5*(ylcp(3:4)-xlcp(1:2)); gamma_c = xlcp(3:4)-ylcp(1:2);
uE = uA + M^-1*(K*qM*dt+h*dt+W*T); qE = qM + 0.5*dt*uE;

sys = [qE(1);uE(1);qE(2);uE(2);qE(3);uE(3)];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% end LCPTS specific
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% End of mdlUpdate.

%==============================================================================
% Calculate outputs
%==============================================================================
function sys = mdlOutputs(t,x,u)

global rf rc rd

sys = [(x(5)-rf*rd*x(3));x(2);x(4);x(4)*rf/rc;x(6)];
% End of mdlOutputs.
Appendix G

S-function: Time-Stepping Method with Augmented Lagrangian

function [sys,x0,str,ts] = sfundualclutchAUGLAGR(t,x,u,flag)

switch flag,
  case 0
    [sys,x0,str,ts] = mdlInitializeSizes; % Initialization
  case 2
    sys = mdlUpdate(t,x,u); % Update discrete states
  case 3
    sys = mdlOutputs(t,x,u); % Calculate outputs
  case {1, 4, 9} % Unused flags
    sys = [];
  otherwise
    error(['unhandled flag = ',num2str(flag)]); % Error handling
end
% End of dualclutchTS.

%==================================================================
% Initialization
%==================================================================

function [sys,x0,str] = mdlInitializeSizes

global M K mu_c mu_f z0 W A dt rf rc rd T p tol
Je=0.2; Jc=0.01; Jf=0.01; Jv=110; rf=0.6; rc=0.4; rd=0.2;
mu_f=1; mu_c=1;
k=6000;
APPENDIX G. S-FUNCTION: TIME-STEPPING METHOD WITH AUGMENTED LAGRANGIAN

AUGLAGR specific

\begin{align*}
M &= \begin{bmatrix} J_e & 0 & 0 \\ J_f + J_c \cdot r_f^2 / r_c^2 & 0 & 0 \\ 0 & 0 & J_v \end{bmatrix}; \quad W = \begin{bmatrix} -1 & -1 \\ r_f / r_c & 0 \\ 0 & 0 \end{bmatrix}; \\
A &= \begin{bmatrix} W' \cdot M^{-1} \cdot W & \text{eye}(2) \\ -\text{eye}(2) & zeros(2) \end{bmatrix}; \quad K = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -r_f^2 \cdot r_d^2 \cdot k & r_d \cdot r_f \cdot k \\ 0 & r_f \cdot r_d \cdot k & -k \end{bmatrix}; \\
dt &= 1e^{-3}; \quad \% \text{ sample time} \\
p &= 10; \quad \% \text{ steepness of regularization} \\
tol &= 1e^{-4}; \quad \% \text{ tolerance for convergence root finding algorithm}
\end{align*}

Call simsizes for a sizes structure, fill it in, and convert it to a sizes array.

sizes = simsizes; sizes.NumContStates = 0; sizes.NumDiscStates = 6; sizes.NumOutputs = 5; sizes.NumInputs = 3; sizes.DirFeedthrough = 0; sizes.NumSampleTimes = 1; sys = simsizes(sizes);

x0 = \begin{bmatrix} 0; 150; 0; 150; 0; 18 \end{bmatrix}; \quad \% \text{ Initialize the discrete states} \\
T = \begin{bmatrix} 0; 0 \end{bmatrix}; \quad \% \text{ Initialize regularization of } T

% Call mdlInitializeSizes.

%==============================================================================
% Update the discrete states
%==============================================================================

function sys = mdlUpdate(t,x,u)

global M K mu_c mu_f qA W A dt z0 T p tol

Fn_f = u(1); Fn_c = u(2); Te = u(3);
Tv = x(6) * 20;

uA = [x(2); x(4); x(6)]; qA = [x(1); x(3); x(5)];

qM = qA + 0.5 * dt * uA; MuFn = [mu_f * Fn_f; mu_c * Fn_c]; h = [Te; 0; -Tv];

converged = 0; while converged == 0
    uE = uA + M^{-1} * (K * qM * dt + h * dt + W * T * dt);

%==============================================================================
% AUGLAGR specific
%==============================================================================
gamma = W'*uE;
T_old = T;
T = prox_Ct(T-p*gamma,MuFn);
error = norm(T-T_old);
converged = error<tol;
end qE = qM + 0.5*dt*uE;

sys = [qE(1);uE(1);qE(2);uE(2);qE(3);uE(3)];

function T = prox_Ct(a,b) T=min(max(-b,a),b);
%%%%% End of AUGLAGR specific
%=====================================================================
% End of mdlUpdate.
%=====================================================================

%=====================================================================
% Calculate outputs
%=====================================================================
function sys = mdlOutputs(t,x,u)

global rf rc rd

sys = [(x(5)-rf*rd*x(3));x(2);x(4);x(4)*rf/rc;x(6)];
% End of mdlOutputs.
Appendix H

Dual-clutch Simulations

The evaluation of comparing the simulation results is presented in the tables H.1 and H.2. In the tables the following abbreviations are used:

- SWITCH: Switch Model,
- LCPEV: event-driven integration with LCP,
- LCPTS: time-stepping with LCP,
- AUGLAGR: time-stepping with augmented lagrange method,
- tolerance: the tolerance on the stick condition that the relative velocity is zero,
- Ts: sample time,
- c.t.: computation time,
- regul. tol.: for the augmented lagrange method the tolerance of the exact regularization,
- regul. par.: for the augmented lagrange method the steepness parameter of the regularization.
### APPENDIX H. DUAL-CLUTCH SIMULATIONS

<table>
<thead>
<tr>
<th>Fixed step; 10 sec simulated; ode 1 (Euler)</th>
<th>LCPEV (I)</th>
<th>LCPTS (I)</th>
<th>LCPTS (I)</th>
<th>SWITCH (I)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>↓ SWITCH (II)</td>
<td>↓ AUGLAGR (II)</td>
<td>↓ SWITCH (II)</td>
<td>AUGLAGR (II)</td>
</tr>
<tr>
<td>base case; Ts 1e-3; regul. tol. 1e-4; regul. par. 1</td>
<td>c.t. (I) 18sec; c.t. (II) 2sec; mean error 1e-10</td>
<td>c.t. (I) 17sec; c.t. (II) 58sec; mean error 1e-3</td>
<td>c.t. (I) 17sec; c.t. (II) 2sec; mean error 1e-1</td>
<td>c.t. (I) 2sec; c.t. (II) 58sec; mean error 1e-1</td>
</tr>
<tr>
<td>Ts 1e-4; regul. tol. 1e-4; regul. par. 1</td>
<td>c.t. (I) 3min; c.t. (II) 22sec; mean error 1e-10</td>
<td>c.t. (I) 3min; c.t. (II) 16min; mean error 1e-3</td>
<td>c.t. (I) 3min; c.t. (II) 22sec; mean error 1e-1</td>
<td>c.t. (I) 22sec; c.t. (II) 16min; mean error 1e-1</td>
</tr>
<tr>
<td>Ts 1e-3; regul. tol. 1e-8; regul. par. 1</td>
<td>n/a</td>
<td>c.t. (I) 17sec; c.t. (II) 3min; mean error 1e-7</td>
<td>n/a</td>
<td>c.t. (I) 2sec; c.t. (II) 3min; mean error 1e-1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fixed step; 10 sec simulated; ode 3</th>
<th>SWITCH(I)</th>
<th>AUGLAGR(I)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ts = 1e-3</td>
<td>c.t.(I) 6 sec; c.t.(II) 4 sec; mean error 1e-2</td>
<td>c.t.(I) 10 sec; c.t.(II) 4 sec; mean error 5e-2</td>
</tr>
<tr>
<td>Ts = 1e-4</td>
<td>c.t.(I) 60 sec; c.t.(II) 26 sec; mean error 1e-2</td>
<td>c.t.(I) 150 sec; c.t.(II) 26 sec; mean error 5e-3</td>
</tr>
</tbody>
</table>

Table H.1: Comparison of results fixed step simulations

Table H.2: Comparison of results between SimDriveline, Switch method and Augmented Lagrange method
Appendix I

Hybrid Control Clutch

I.1 HYSDEL model

```plaintext
SYSTEM hysclutch {
  INTERFACE {
    PARAMETER {
      REAL Je, Jv, mu, b, ts, reltol;
      REAL invJe = 1/Je;
      REAL invJv = 1/Jv;
      REAL invJeplusJv = 1/(Je+Jv);
      REAL wemin, wemax, gammamin, gammamax, Temin,
        Temax, Teincmin, Teincmax, Fnmin, Fnmax, Fnincmin, Fnincmax;
    }
    STATE {
      REAL Te [Temin,Temax];
      REAL Fn [Fnmin,Fnmax];
      REAL we [wemin,wemax];
      REAL gamma [gammamin,gammamax];
      REAL lc [-100,100];
    }
    INPUT {
      REAL Teinc [Teincmin,Teincmax];
      REAL Fninc [Fnincmin,Fnincmax];
    }
    OUTPUT {
      REAL y1, y2, y3;
    }
  }
  IMPLEMENTATION {
    AUX {
      BOOL stick, karnopp, neg_slip, lockupahead;
      REAL Tcl, Tcl_slip, lockupcost;
    }
    AD {
      neg_slip = gamma <= 0;
      stick = gamma <= reltol;
      karnopp = (Te*Jv + b*(we-gamma)*Je)*invJeplusJv <= mu*Fn;
      lockupahead = gamma <= 10*reltol;
    }
  }
}
```
APPENDIX I. HYBRID CONTROL CLUTCH

{ DA {
  lockupcost = {IF lockupahead & ~stick THEN ts*((Te - Tcl)*invJe - (Tcl - b*(we-gamma))*invJv) ELSE 0};
  Tcl_slip = {IF neg_slip THEN -mu*Fn ELSE mu*Fn};
  Tcl = {IF stick & karnopp THEN (Te*Jv + b*(we-gamma)*Je)*invJeplusJv ELSE Tcl_slip};
}
CONTINUOUS {
  Te = Te + Teinc;
  Fn = Fn + Fninc;
  we = we + ts*(Te - Tcl)*invJe;
  gamma = gamma + ts*((Te - Tcl)*invJe - (Tcl - b*(we-gamma))*invJv);
  lc = lockupcost;
}
OUTPUT {
  y1 = we;
  y2 = gamma;
  y3 = lc;
}
}

I.2 Hybrid Controller

clear all close all
Je = 0.2; % Engine Inertia
Jv = 0.8; % Vehicle-side Inertia
mu = 1; % Friction Coefficient
b = 0.67; % Load is dependent on velocity
ts = 1e-2; % Sample Time
reitol = 1; % Stick boundary
wemin = 80; % Engine Stall constraint (rad/s)
wemax = 200; % Engine Max. rotational velocity (rad/s)
gammamin = -1; % Min. relative velocity (rad/s)
gammamax = 200; % Max. relative velocity
Temin = 0; % Min. Engine Torque
Temax = 150; % Max. Engine Torque
Fnmin = 0; % Min. Clutch Normal Force
Fnmax = 180; % Max. Clutch Normal Force
Tegrad = 500; % max gradient engine torque per second
Fngrad = 800; % max gradient normal force clutch per second
Teincmin = -Tegrad*ts; % Min. increment Engine Torque
Teincmax = Tegrad*ts; % Max. increment Engine Torque
Fnincmin = -Fngrad*ts; % Min. increment Clutch Normal Force
Fnincmax = Fngrad*ts; % Max. increment Clutch Normal Force
I.2. HYBRID CONTROLLER

hybclutch = mld('hysclutch',ts)

gammamin = 0;

Q.y = diag([1 4 4]); % [omega_engine gamma lockupcost]
Q.u = [1 0;0 1]; % Q.x = [1 0;0 4]; % Q.z = [];
% Q.xN = [];
Q.rho = Inf;

Nc = 10;

% Constraints
umax=[0.99*Teincmax,0.99*Fnincmax]';
umin=[0.99*Teincmin,0.99*Fnincmin]';
xmin=[1.01*Temin,1.01*Fnmin,1.01*wemin,1.01*gammamin,-11]';
xmax=[0.99*Temax,0.99*Fnmax,0.99*wemax,0.99*gammamax,11]';
limits.umax=umax; limits.umin=umin; limits.xmin=xmin;
limits.xmax=xmax;

refs.y = [1 2 3];
%refs.u = [1 2];

hybclutchcon = hybcon(hybclutch,Q,Nc,limits,refs)

sim('hysclutchhybrid',[0 1])

load omshc.mat load inphc.mat

subplot(2,1,1);
plot(inputshybcon(1,:),inputshybcon(2,:),inputshybcon(1,:),inputshybcon(3,:))
legend('Te','Fn_c','Location','best') xlabel('Time (sec)') ylabel('Torque (Nm) and Force (N)') title('Engine Torque and Clutch Normal Force') grid on

subplot(2,1,2);
plot(omegasshybcon(1,:),omegasshybcon(2,:),omegasshybcon(1,:),omegasshybcon(3,:))
legend('Omega_e','Omega_v','Location','best') xlabel('Time (sec)') ylabel('Rotational Velocities (rad/sec)') title('Engine rotational velocity and Vehicle-side rotational velocity') grid on

% refymin = [-1;-1];
% refymax = [1;1];
%
% range.xmin = xmin;
% range.xmax = xmax;
% range.umin = umin;
% range.umax = umax;
% range.refymin = refymin;
% range.refymax = refymax;

%expC = expcon(hybclutchcon,range)
Figure I.1: Simulink simulation model of hybrid controller
Appendix J

Planetary Gear Set

A schematic drawing of the planetary gear set can be seen in Figure J.1. In the derivation of the theoretical model we have simply assigned the torques $T_a$, $T_s$ and $T_p$ to the three branches of the planetary gear, respectively the annulus, the sun and the planet carrier. These torques are not external, but tied up by the other external forces because of kinematic constraints and the law of power conservation, since the planetary gear is a mere torque splitter,

$$\omega_s = (z + 1)\omega_p - z\omega_a, \quad (J.1)$$

$$T_a\omega_a + T_s\omega_s - T_p\omega_p = 0. \quad (J.2)$$

with

$$z = \frac{d_a}{d_s} \quad (J.3)$$

For all $\omega_a$, $\omega_s$ and $\omega_p$ the equations (J.1) and (J.2) result in,

$$T_a = zT_s,$$

$$T_p = (z + 1)T_s. \quad (J.4)$$

By raising equation (J.1) to the acceleration level one gets,

$$\dot{\omega}_s = (z + 1)\dot{\omega}_p - z\dot{\omega}_a. \quad (J.5)$$
Appendix K

Hybrid Control B-IST

K.1 HYSDEL model

K.1.1 Part 1

```plaintext
SYSTEM BST23part1code {
  INTERFACE {
    PARAMETER {
      REAL ts;
      REAL rgb = 1/1.913;
      REAL ratio3 = 1/1.220;
      REAL rd = 1/3.737;
      REAL rgn = 0.7;
      REAL z = 2.1155;
      REAL rp = (z+1)*rgn/z;
      REAL Jic_e = 0.12;
      REAL Jcl_e = 0.03;
      REAL Je = Jic_e + Jcl_e;
      REAL Jf = 0.08;
      REAL Jc1_p = 0.003;
      REAL Jtr_p = 0.0027;
      REAL Jp = Jc1_p + Jtr_p;
      REAL Js = 0.0028;
      REAL Jv = 110;
      REAL Jps = Js + Jp/(rgb^2);
      REAL alfa = (Jf*Je + rgn^2*Jps*Jf + (rgn/z)^2*Jps*Je);
      REAL wemin = 84;
      REAL weymax = 700;
      REAL dphimin = -1;
      REAL dphimax = 1;
      REAL Fnmin = 0;
      REAL Fnbmax = 200;
      REAL Fnclmin = 0;
      REAL Fnclmax = 200;
      REAL Temin = 0;
      REAL Temax = 145;
      REAL Fhincmin = -500;
      REAL Fhincmax = 800;
    }
  }
}
```

REAL Teincmin = -400;
REAL Teincmax = 500;
REAL kds = 6000;
REAL bv = 10;
REAL mu = 1;
}

STATE {
REAL Te [Temin,Temax];
REAL Fnb [Fnbmin,Fnbmax];
REAL Fncl [Fnclmin,Fnclmax];
REAL dphi [dphimin,dphimax]; /* Angle deflection */
REAL we [wemin,wemax]; /* Engine-flywheel velocity */
REAL ws [rgb*wemin,rgb*wemax]; /* Secondary shaft velocity */
REAL wv [rd*rgb*wemin,rd*rgb*wemax]; /* Vehicle-side velocity */
}

INPUT {
REAL Teinc [Teincmin,Teincmax]; /* Engine Torque */
REAL Fnbinc [Fnbincmin,Fnbincmax]; /* First clutch normal force */
REAL Fnclinc [Fnclincmin,Fnclincmax]; /* Second clutch normal force */
}

OUTPUT {
REAL y1,y2,y3;
}

IMPLEMENTATION {
AUX {
REAL Tb, Tcl; /* Clutch torques */
REAL Tds, Tv;
BOOL negslip;
BOOL dwe1, dwe2, dwe3, dTe1, dTe2, dTe3;
/*
  BOOL Fnclzero, Fnclinczero;/*
}

AD{
  negslip = we-rgb*ws <= 0;
dwe1 = we <= 105;
dTe1 = Te <= 105-(105-we)*45/21;
dwe2 = we <= 419;
dTe2 = Te <= 145-(419-we)*40/314;
dwe3 = we <= 628;
dTe3 = Te <= 127+(628-we)*18/209;
  /*
  Fnclzero = Fncl <=1;
  Fnclinczero = Fnclinc <=100;*/
}

DA{
  Tcl = (IF negslip THEN -mu*Fncl ELSE mu*Fncl);
}

LINEAR{
  Tb = mu*Fnb;
  Tds = kds*dphi + (kds/1000)*(rd*ws-wv);
  Tv = bv*wv;
}

CONTINUOUS {
  Te = Te + ts*Teinc;
  Fnb = Fnb + ts*Fnbinc;
}
K.1. HYSDEL MODEL

\[
F_{ncl} = F_{ncl} + ts \cdot F_{nclinc};
\]

\[
d\phi = d\phi + ts \cdot (rd \cdot ws - wv); \quad /* \text{Angle deflection} */
\]

\[
we = we + ts \cdot (((\text{rgn} \cdot \text{rgn} / (z \cdot z)) \cdot J_{ps} + J_{f}) \cdot T_e - \text{rgn} \cdot \text{rgn} / z \cdot J_{ps} \cdot T_b - \text{rgn} \cdot \text{rd} \cdot \text{J}_{f} \cdot T_{ds}) / \alpha);
\]

\[
ws = ws + ts \cdot ((\text{rgn} \cdot \text{J}_{f} \cdot T_e - \text{rgn} / \text{z} \cdot \text{J}_{ps} \cdot T_{ds} + \text{rgn} / \text{rgb} \cdot \text{J}_{f} \cdot T_{cl}) + \text{rgn} / z \cdot \text{J}_{f} \cdot T_{b}) / \alpha;
\]

\[
w_v = w_v + ts \cdot (T_{ds} - T_v) / J_v; \quad /* \text{Vehicle side velocity} */
\]

\}

OUTPUT{

\[
y_1 = (z+1) / \text{rp} \cdot ws - z \cdot we;
\]

\[
y_2 = we - ws / \text{ratio3};
\]

\[
y_3 = kds \cdot d\phi + (kds / 1000) \cdot (rd \cdot ws - w_v);
\]

}\n
MUST{

\[
dwe1 \rightarrow dTe1; \quad (dwe2 \& \sim dwe1) \rightarrow dTe2; \quad (dwe3 \& \sim dwe2) \rightarrow dTe3;
\]

/*Fnclzero \rightarrow Fnclinczero;*/

}\n
K.1.2 Part 2

SYSTEM BST23part2code {

INTERFACE {

PARAMETER {

REAL ts;
REAL rgb = 1/1.220;
REAL rd = 1/3.737;
REAL rgn = 0.7;
REAL z = 2.1155;
REAL rp = (z+1) * rgn / z;
REAL Jic_e = 0.12;
REAL Jcl_e = 0.034;
REAL Je = Jic_e + Jcl_e;
REAL Jf = 0.08;
REAL Jcl_p = 0.003;
REAL Jtr_p = 0.0027;
REAL Jp = Jcl_p + Jtr_p;
REAL Js = 0.0028;
REAL Jv = 110;
REAL Jps = Js + Jp / (rgb^2);
REAL \alpha = (Jf \cdot Je + rgn^2 \cdot Jps \cdot Jf + (rgn / z)^2 \cdot Jps \cdot Je);
REAL wemin = 84;
REAL wemax = 700;
REAL dphimin = -1;
REAL dphimax = 1;
REAL FnBmin = 0;
REAL FnBmax = 200;
REAL Fnclmin = 0;
REAL Fnclmax = 200;
REAL Temin = 0;
}
APPENDIX K. HYBRID CONTROL B-IST

REAL Temax = 145;
REAL Fnincmin = -500;
REAL Fnincmax = 800;
REAL Teincmin = -400;
REAL Teincmax = 500;
REAL kds = 6000;
REAL bv = 10;
REAL mu = 1;

}\ STATE {
    REAL Te [Temin,Temax];
    REAL Fnb [Fnbnmin,Fnbmax];
    REAL Fncl [Fnclmin,Fnclmax];
    REAL dphi [dphimin,dphimax]; /* Angle deflection */
    REAL we [wemin,wemax]; /* Engine-flywheel velocity */
    REAL ws [rgb*wemin,rgb*wemax]; /* Secondary shaft velocity */
    REAL wv [rd*rgb*wemin,rd*rgb*wemax]; /* Vehicle-side velocity */
}\ INPUT {
    REAL Teinc [Teincmin,Teincmax]; /* Engine Torque */
    REAL Fnbininc [Fnincmin,Fnincmax]; /* First clutch normal force */
    REAL Fnnclinc [Fnincmin,Fnincmax]; /* Second clutch normal force */
}\ OUTPUT {
    REAL y1,y2,y3;
}\ IMPLEMENTATION {
\ AUX {
    REAL Tb, Tcl; /* Clutch torques */
    REAL Tds, Tv;
    BOOL negslip, brakezero, flywheelneg;
    BOOL dwe1, dwe2, dwe3, dTe1, dTe2, dTe3;
}\ AD{
    negslip = we-rgb*ws <= 0;
    brakezero = Fnb <= 0.01;
    flywheelneg = (z+1)/rp*ws-z*we <= -5;
    dwe1 = we <= 105;
    dTe1 = Te <= 105-(105-we)*45/21;
    dwe2 = we <= 419;
    dTe2 = Te <= 145-(419-we)*40/314;
    dwe3 = we <= 628;
    dTe3 = Te <= 127+(628-we)*18/209;
}\ DA{
    Tcl = {IF negslip THEN -mu*Fncl ELSE mu*Fncl};
}\ LINEAR{
    Tb = mu*Fnb;
    Tds = kds*dphi + (kds/1000)*(rd*ws-wv);
    Tv = bv*wv;
}\ CONTINUOUS {
K.2. HYBRID CONTROLLER B-IST 2-3 UPSHIFT

Te = Te + ts*Teinc;
Fnb = Fnb + ts*Fnbinc;
Fnc1 = Fnc1 + ts*Fnc1inc;
dphi = dphi + ts*(rd*ws-wv); /* Angle deflection */
we = we + ts*(((rgn*rgn/(z*z)))*Jps+Jf)*Te /* Engine-flywheel velocity */
    - (((rgn*rgn/(z*z)))*Jps+Jf-rgn/rgb*Jf)*Tcl
    - rgn*rgn/z*Jps*Tb - rgn*rd*Jf*Tds)/alfa);
ws = ws + ts*((rgn*Jf*Te /* Secondary shaft velocity */
    + (rgn*rgn/(z*z*rgb)*Je+(rgn/rgb-1)*rgn*Jf)*Tcl
    + rgn/z*Je*Tb - (rd*rgn*rgn*Jf+rd*(rgn*rgn/(z*z))*Je)*Tds)/alfa);
    wv = wv + ts*(Tds-Tv)/Jv;
}

OUTPUT{
    y1 = (z+1)/rp*ws-z*we;
    y2 = we-ws/rgb;
    y3 = kds*dphi + (kds/1000)*(rd*ws-wv);
}

MUST{
    brakezero | flywheelneg;
    dwe1 -> dTe1; (dwe2 & ~dwe1) -> dTe2; (dwe3 & ~dwe2) -> dTe3;
}

K.2 Hybrid Controller B-IST 2-3 Upshift

K.2.1 Case Constant Driveshaft Torque

close all
clear all

    ts = 1e-2;
t1 = [0 0.5];
t2 = [0.5 1.25];

    ratio2 = 1/1.913;
    ratio3 = 1/1.220;
    rd = 1/3.737;
    rgn = 0.7;
    z = 2.1155;
    rp = (z+1)*rgn/z;
    Jic_e = 0.12;
    Jcl_e = 0.034;
    Je = Jic_e + Jcl_e;
    Jf = 0.08;
    Jcl_p = 0.003;
    Jtr_p = 0.0027;
    Jp = Jcl_p + Jtr_p;
    Js = 0.0028;
    Jv = 110;
    Jps2 = Js + Jp/(ratio2^2);
    Jps3 = Js + Jp/(ratio3^2);
k = 6000;
\[d = \frac{k}{1000};\]
\[bv = 10;\]
\[mub = 1;\]
\[mucl = 1;\]
\[Tds\text{\_init1} = 600;\]
\[Te\text{\_init1} = \frac{Tds\text{\_init1}}{ratio2*rd};\]
\[Tb\text{\_init1} = 0;\]
\[dphi\text{\_init1} = \frac{Tds\text{\_init1}}{k};\]
\[Tcl\text{\_init1} = \frac{((-Jf*Te\text{\_init1} + rgn*rd*Jf*Tds\text{\_init1} + rgn*Jf*Te\text{\_init1})) * z^2 + (rgn*Je*Tb\text{\_init1} + rgn*Jps2*Tb\text{\_init1}) * z - Te\text{\_init1} * rgn*Jps2 - rgn*Jf*Tds\text{\_init1} * Je)}{((-ratio2*Jf + rgn*Jf + rgn*ratio2*Jf - rgn^2*Jf) * z^2 - rgn*Jps2*ratio2 - rgn^2*Je) * ratio2};\]
\[wvevenwicht1 = 75;\]
\[x0\_1 = [Te\text{\_init1};Tb\text{\_init1};Tcl\text{\_init1};dphi\text{\_init1};wvevenwicht1/(ratio2*rd);wvevenwicht1/rd; wvevenwicht1];\]
\[wf\text{\_init1} = (z+1)/rp*x0\_1(6)-z*x0\_1(5);\]
\[BST23\_flatTds\_part1sys = mld('BST23\_flatTds\_part1',ts)\]
\[clear COSTS\]
\[COSTS.y = diag([3 50]);\]
\[COSTS.x = diag([1 100]);\]
\[N = 10;\]
\[wemin = 1.01*84;\]
\[wemax = 0.99*700;\]
\[dphimin = 0.99*-1;\]
\[dphimax = 0.99*1;\]
\[Fnbmin = 0.01;\]
\[Fnbmax = 0.99*200;\]
\[Fnclmin = 0.01;\]
\[Fnclmax = 0.99*200;\]
\[Temin = 0.01;\]
\[Temax = 0.99*145;\]
\[Fnincmin = 0.99*-500;\]
\[Fnincmax = 0.99*800;\]
\[Teincmin = 0.99*-400;\]
\[Teincmax = 0.99*500;\]
\[wfmin = -300;\]
\[wfmax = -5;\]
\[gammamin = 0;\]
\[gammamax = wemax*(ratio3-ratio2);\]
\[Tdsmin = 0;\]
\[Tdsmax = 2000;\]
\[clear LIMITS\]
\[LIMITS.umin = [Teincmin,Fnincmin,Fnincmin];\]
\[LIMITS.umax = [Teincmax,Fnincmax,Fnincmax];\]
\[LIMITS.xmin = [Temin,Fnbmin,Fnclmin,dphimin,wemin,ratio2*rd*wemin,ratio2*rd*wmwemin];\]
\[LIMITS.xmax = [Temax,Fnbmax,Fnclmax,dphimax,wemax,ratio3*rd*wemax,ratio3*rd*wmwemax];\]
\[LIMITS.ymin = [wfmin,gammamin,Tdsmin];\]
\[LIMITS.ymax = [wfmax,gammamax,Tdsmax];\]
clear REFSIGNALS
REFSIGNS.y = [1 3]; %[flywheel_vel driveshaft_torque]
REFSIGNS.x = [2 3]; %[brake_force clutch_force]

ContrBST23_flatTds_part1 = hybcon(BST23_flatTds_part1sys,COSTS,N,LIMITS,REFSIGNS)
try
    sim('SIMBST23_flatTds_part1',t1)
catch
    errmsg1 = lasterr
end

load BST23_flatTds_part1inputs
load BST23_flatTds_part1deltaphi
load BST23_flatTds_part1omegas
load BST23_flatTds_part1DriveTorque

timepart1 = BST23_flatTds_p1inp(1,:);

inputspart1 = BST23_flatTds_p1inp(2:4,:);
figure
subplot(3,2,[1 2]), plot(timepart1,inputspart1)
title('Inputs for upshift 2 - 3 flat Tds PART I')
legend('Engine Torque (Nm)','Normal Force Brake (N)','Normal Force Clutch (N)',
'Location','best')
xlabel('Time (sec)')
grid on

omegaspart1 = BST23_flatTds_p1oms([2 3],:);
subplot(3,2,3), plot(timepart1,omegaspart1)
title('Rotational Velocities for upshift 2 - 3 flat Tds PART I')
legend('Engine-side velocity', 'Primary Shaft velocity','Location','best')
ylabel('Rotational Speed (rad/sec)')
xlabel('Time (sec)')
grid on

wfpart1 = [BST23_flatTds_p1oms(7,:)];
subplot(3,2,4), plot(timepart1,wfpart1)
title('Flywheel velocity for upshift 2 - 3 flat Tds PART I')
ylabel('Rotational Speed (rad/sec)')
xlabel('Time (sec)')
grid on

DriveTorquepart1 = BST23_flatTds_p1Tds(2,:);
subplot(3,2,[5 6]), plot(timepart1,DriveTorquepart1)
title('Driveshaft Torque for upshift 2 - 3 flat Tds PART I')
ylabel('Torque (Nm)')
xlabel('Time (sec)')
grid on

if ~exist('errmsg1','var')
    Te_init2 = BST23_flatTds_p1inp(2,end);
end
APPENDIX K. HYBRID CONTROL B-IST

\[
\begin{align*}
\text{Tb\textsubscript{init}2} &= \text{BST23\_flatTds\_plinp}(3,\text{end}); \\
\text{Tds\textsubscript{init}2} &= \text{BST23\_flatTds\_p1Tds}(2,\text{end}); \\
\text{dphi\textsubscript{init}2} &= \text{BST23\_flatTds\_p1dp}(2,\text{end}); \\
\text{Tcl\textsubscript{init}2} &= \text{BST23\_flatTds\_plinp}(4,\text{end}); \\
\text{we2} &= \text{BST23\_flatTds\_p1oms}(2,\text{end}); \\
\text{ws2} &= \text{BST23\_flatTds\_p1oms}(5,\text{end}); \\
\text{wv2} &= \text{BST23\_flatTds\_p1oms}(6,\text{end}); \\
\text{gamma\textsubscript{init}2} &= \text{we2} - \frac{\text{ws2}}{\text{ratio3}}; \\
\text{x0\_2} &= [\text{Te\textsubscript{init}2};\text{Tb\textsubscript{init}2};\text{Tcl\textsubscript{init}2};\text{dphi\textsubscript{init}2};\text{we2};\text{ws2};\text{wv2}]; \\
\text{BST23\_flatTds\_part2sys} &= \text{mld('BST23\_flatTds\_part2',ts)} \\
\text{clear COSTS} \\
\text{COSTS.y} &= \text{diag([3 100]);} \quad %[\text{relative\_vel driveshaft\_torque}] \\
\text{COSTS.x} &= \text{diag([1]);} \quad %[\text{brake\_force}] \\
%\text{COSTS.u} &= \text{diag([1 1 1]);} \\
\text{N} &= 10; \\
\text{wemin} &= 1.01\times84; \\
\text{wemax} &= 0.99\times700; \\
\text{dphimin} &= 0.99\times-1; \\
\text{dphimax} &= 0.99\times1; \\
\text{Fnbmin} &= 0.01; \\
\text{Fnbmax} &= 0.99\times200; \\
\text{Fnclmin} &= 0.01; \\
\text{Fnclmax} &= 0.99\times200; \\
\text{Temin} &= 0.01; \\
\text{Temax} &= 0.99\times145; \\
\text{Fnincmin} &= 0.99\times-500; \\
\text{Fnincmax} &= 0.99\times800; \\
\text{Teincmin} &= 0.99\times-400; \\
\text{Teincmax} &= 0.99\times500; \\
\text{wfmin} &= -300; \\
\text{wfmax} &= 300; \\
\text{gammamin} &= 0; \\
\text{gammamax} &= \text{wemax}(\text{ratio3}-\text{ratio2}); \\
\text{Tdsmin} &= 0; \\
\text{Tdsmax} &= 2000; \\
\text{clear LIMITS} \\
\text{LIMITS.umin} &= [\text{Teincmin}\text{Fnincmin}\text{Fnincmin}]; \\
\text{LIMITS.umax} &= [\text{Teincmax}\text{Fnincmax}\text{Fnincmax}]; \\
\text{LIMITS.xmin} &= [\text{Temin}\text{Fnbmin}\text{Fnclmin}\text{dphimin}\text{wemin}\text{ratio3}\times\text{wemin}\text{ratio3}\times\text{rd}\times\text{wemin}]; \\
\text{LIMITS.xmax} &= [\text{Temax}\text{Fnbmax}\text{Fnclmax}\text{dphimax}\text{wemax}\text{ratio3}\times\text{wemax}\text{ratio3}\times\text{rd}\times\text{wemax}]; \\
\text{LIMITS.ymin} &= [\text{wfmin}\text{gammamin}\text{Tdsmin}]; \\
\text{LIMITS.ymax} &= [\text{wfmax}\text{gammamax}\text{Tdsmax}]; \\
\text{clear REFSIGNALS} \\
\text{REFSIGNALS.y} &= [2 3]; \quad %[\text{relative\_vel driveshaft\_torque}] \\
\text{REFSIGNALS.x} &= [2]; \quad %[\text{brake\_force}] \\
\text{Contr\_BST23\_flatTds\_part2} &= \text{hybcon(BST23\_flatTds\_part2sys,COSTS,N,LIMITS,REFSIGNALS)}; \\
\text{try}
K.2. HYBRID CONTROLLER B-IST 2-3 UPSHIFT

```matlab
K.2.2 Case Decreasing Driveshaft Torque

close all
clear all

ts = 1e-2;
t1 = [0 0.5];
t2 = [0.5 1.5];
```
ratio2 = 1/1.913;
ratio3 = 1/1.220;
rd = 1/3.737;
grn = 0.7;
z = 2.1155;
rp = (z+1)*grn/z;
Jic_e = 0.12;
Jcl_e = 0.034;
Je = Jic_e + Jcl_e;
Jf = 0.08;
Jcl_p = 0.003;
Jtr_p = 0.0027;
Jp = Jcl_p + Jtr_p;
Js = 0.0028;
Jv = 110;
Jps2 = Js + Jp/(ratio2^2);
Jps3 = Js + Jp/(ratio3^2);
k = 6000;
d = k/1000;
bv = 10;
mub = 1;
muc1 = 1;

Tds_init1 = 800;
Tds_3 = 600;
Te_init1 = Tds_init1*ratio2*rd;
Tb_init1 = 0;
dphi_init1 = Tds_init1/k;
Tcl_init1 = ((-Jf*Te_init1+grn*rd*Jf*Tds_init1+grn*Je*Tb_init1+grn*Jps2*Tb_init1)*z^2
 + (grn*Je*Te_init1+grn*Jf*Tds_init1)*z-Te_init1*grn^2*Jps2
 - grn^2*rd*Tds_init1*Je)/((-ratio2*Jf+grn*Jf+grn*ratio2*Jf-rgrn^2*Jf)*z^2
 - grn^2*Jps2*ratio2-grn^2*Je)*ratio2;

wvevenwicht1 = 75;

x0_1 = [Te_init1;Tb_init1;Tcl_init1;dphi_init1;wvevenwicht1/(ratio2*rd);wvevenwicht1/rd;
        wvevenwicht1];
wf_init1 = (z+1)/rp*x0_1(6)-z*x0_1(5);

BST23_kd_part1sys = mld('BST23kd_part1',ts)
clear COSTS
COSTS.y = diag([3 50]);  %[Flywheel_speed Driveshaft_Torque]
COSTS.x = diag([1 100]);  %[Brake_Force Clutch_Force]
N = 10;

wemin = 1.01*84;
wemax = 0.99*700;
dphimin = 0.99*-1;
dphimax = 0.99*1;
Fnbmmin = 0.01;
Fnbmax = 0.99*200;
Fnc1min = 0.01;
Fnc1max = 0.99*200;
K.2. HYBRID CONTROLLER B-IST 2-3 UPSHIFT

Temin = 0.01;
Temax = 0.99*145;
Fnincmin = 0.99*-500;
Fnincmax = 0.99*800;
Teincmin = 0.99*-400;
Teincmax = 0.99*500;
wfmin = -300;
wfmax = -5;
gammamin = 0;
gammamax = wemax*(ratio3-ratio2);
Tdsmin = 0;
Tdsmax = 2000;
clear LIMITS
LIMITS.umin = [Teincmin,Fnincmin,Fnincmin];
LIMITS.umax = [Teincmax,Fnincmax,Fnincmax];
LIMITS.xmin = [Temin,Fnbmin,Fnclmin,dphimin,wemin,ratio2*wemin,ratio2*rd*wemin];
LIMITS.xmax = [Temax,Fnbmax,Fnclmax,dphimax,wemax,ratio3*wemax,ratio3*rd*wemax];
LIMITS.ymin = [wfmin,gammamin,Tdsmin];
LIMITS.ymax = [wfmax,gammamax,Tdsmax];
clear REFSIGNALS
REFSIGNALS.y = [1 3]; %[Flywheel_speed Driveshaft_Torque]
REFSIGNALS.x = [2 3]; %[Brake_Force Clutch_Force]

ContrBST23_kd_part1 = hybcon(BST23_kd_part1sys,COSTS,N,LIMITS,REFSIGNALS)
try
    sim('SIMBST23_kd_part1',t1)
catch
    errmsg1 = lasterr
end

load BST23_kd_part1inputs
load BST23_kd_part1deltaphi
load BST23_kd_part1omegas
load BST23_kd_part1DriveTorque
timepart1 = BST23_kd_part1inp(1,:);
inputspart1 = BST23_kd_part1inp(2:4,:);
figure
subplot(3,2,[1 2]), plot(timepart1,inputspart1)
title('Inputs for upshift 2 - 3 kickdown PART I')
legend('Engine Torque (Nm)','Normal Force Brake (N)','Normal Force Clutch (N)',
       'Location','Best')
xlabel('Time (sec)')
grid on

omegaspart1 = BST23_kd_part1oms([2 3,:]);
subplot(3,2,3), plot(timepart1,omegaspart1)
title('Rotational Velocities for upshift 2 - 3 kickdown PART I')
legend('Engine-side velocity', 'Primary Shaft velocity','Location','Best')
ylabel('Rotational Speed (rad/sec)')
xlabel('Time (sec)')
grid on
wfpart1 = BST23_kd_part1oms(7,:);
subplot(3,2,4), plot(timepart1,wfpart1)
title('Flywheel Velocity for upshift 2 - 3 kickdown PART I')
ylabel('Rotational Speed (rad/sec)')
xlabel('Time (sec)')
grid on

DriveTorquepart1 = BST23_kd_part1Tds(2,:);
subplot(3,2,[5 6]), plot(timepart1,DriveTorquepart1)
title('Driveshaft Torque for upshift 2 - 3 kickdown PART I')
ylabel('Torque (Nm)')
xlabel('Time (sec)')
grid on

if ~exist('errmsg1','var')
    Te_init2 = BST23_kd_part1inp(2,end);
    Tb_init2 = BST23_kd_part1inp(3,end);
    Tds_init2 = BST23_kd_part1Tds(2,end);
    dphi_init2 = BST23_kd_part1dphi(2,end);
    Tcl_init2 = BST23_kd_part1inp(4,end);
    Tds_3 = 600;
    Te_3 = Tds_3*ratio3*rd;
    we2 = BST23_kd_part1oms(2,end);
    ws2 = BST23_kd_part1oms(5,end);
    wv2 = BST23_kd_part1oms(6,end);
    gamma_init2 = we2 - ws2/ratio3;

    x0_2 = [Te_init2;Tb_init2;Tcl_init2;dphi_init2;we2;ws2;wv2];

    BST23_kd_part2sys = mld('BST23_kd_part2',ts)
    clear COSTS
    COSTS.y = diag([5 50]);  %[Relative_Velocity Driveshaft_Torque]
    COSTS.x = diag([2]);    %[Brake_Force]
    %COSTS.u = diag([1 1 1]);
    N = 10;

    wemin = 1.01*84;
    wemax = 0.99*700;
    dphimin = 0.99*-1;
    dphimax = 0.99*1;
    Fnbnmin = 0.01;
    Fnbnmax = 0.99*200;
    Fnc1min = 0.01;
    Fnc1max = 0.99*200;
    Temin = 0.01;
    Temax = 0.99*145;
    Fnincmin = 0.99*-500;
    Fnincmax = 0.99*800;
    Teincmin = 0.99*-400;
    Teincmax = 0.99*500;
K.2. HYBRID CONTROLLER B-IST 2-3 UPSHIFT

\[
\begin{align*}
wf_{\text{min}} &= -300; \\
wf_{\text{max}} &= 300; \\
\gamma_{\text{min}} &= -100; \\
\gamma_{\text{max}} &= w_{\text{emax}}(\text{ratio}_3 - \text{ratio}_2); \\
T_{ds_{\text{min}}} &= 0; \\
T_{ds_{\text{max}}} &= 2000; \\
\text{clear LIMITS} \\
\text{LIMITS}.\text{umin} &= [\text{Teinc}_{\text{min}}, \text{Fninc}_{\text{min}}, \text{Fninc}_{\text{min}}]; \\
\text{LIMITS}.\text{umax} &= [\text{Teinc}_{\text{max}}, \text{Fninc}_{\text{max}}, \text{Fninc}_{\text{max}}]; \\
\text{LIMITS}.\text{xmin} &= [\text{Tem}_{\text{min}}, \text{Fnb}_{\text{min}}, \text{Fncl}_{\text{min}}, \phi_{\text{min}}, w_{\text{emin}}, \text{ratio}_3 w_{\text{emin}}, \text{ratio}_3 \text{rd} w_{\text{emin}}]; \\
\text{LIMITS}.\text{xmax} &= [\text{Tem}_{\text{max}}, \text{Fnb}_{\text{max}}, \text{Fncl}_{\text{max}}, \phi_{\text{max}}, w_{\text{emax}}, \text{ratio}_3 w_{\text{emax}}, \text{ratio}_3 \text{rd} w_{\text{emax}}]; \\
\text{LIMITS}.\text{ymin} &= [wf_{\text{min}}, \gamma_{\text{min}}, T_{ds_{\text{min}}}] \\
\text{LIMITS}.\text{ymax} &= [wf_{\text{max}}, \gamma_{\text{max}}, T_{ds_{\text{max}}}] \\
\text{clear REFSIGNALS} \\
\text{REFSIGNALS}.y &= [2, 3]; \quad %[\text{Relative} \_ \text{Velocity} \text{ Driveshaft} \_ \text{Torque}] \\
\text{REFSIGNALS}.x &= [2]; \quad %[\text{Brake} \_ \text{Force}] \\
\text{Contr}BST23\_kd\_part2 &= \text{hybcon}([\text{BST23}_{\text{kd}}\_\text{part2sys}, \text{COSTS}, \text{N}, \text{LIMITS}, \text{REFSIGNALS}]) \\
\text{try} \\
\text{sim}(\text{SIMBST23}_{\text{kd}}\_\text{part2}', \text{t2}) \\
\text{catch} \\
\text{errmsg2} &= \text{lasterr} \\
\text{end} \\
\text{load} \text{ BST23}_{\text{kd}}\_\text{part2inputs} \\
\text{load} \text{ BST23}_{\text{kd}}\_\text{part2deltaphi} \\
\text{load} \text{ BST23}_{\text{kd}}\_\text{part2omegas} \\
\text{load} \text{ BST23}_{\text{kd}}\_\text{part2DriveTorque} \\
\text{close all} \\
\text{time} &= [\text{BST23}_{\text{kd}}\_\text{part1inp}(1,:) \text{ BST23}_{\text{kd}}\_\text{part2inp}(1,:)]; \\
\text{inputs} &= [\text{BST23}_{\text{kd}}\_\text{part1inp}(2:4,:) \text{ BST23}_{\text{kd}}\_\text{part2inp}(2:4,:)]; \\
\text{figure} \\
\text{subplot}(3,2,[1 2]), \text{plot}(\text{time}, \text{inputs}) \\
\text{title}('\text{Inputs for upshift 2 - 3 kickdown}') \\
\text{legend}('\text{Engine Torque (Nm)}', '\text{Normal Force Brake (N)}', '\text{Normal Force Clutch (N)}', \\
'\text{Location}', '\text{best}') \\
\text{xlabel}('\text{Time (sec)}') \\
\text{grid on} \\
\text{omegas} &= [\text{BST23}_{\text{kd}}\_\text{part1oms}(\text{[2 3]},:) \text{ BST23}_{\text{kd}}\_\text{part2oms}(\text{[2 4]},:)]; \\
\text{subplot}(3,2,3), \text{plot}(\text{time}, \text{omegas}) \\
\text{title}('\text{Rotational Velocities for upshift 2 - 3 kickdown}') \\
\text{legend}('\text{Engine-side velocity}', '\text{Primary Shaft velocity}', '\text{Location}', '\text{best}') \\
\text{ylabel}('\text{Rotational Speed (rad/sec)}') \\
\text{xlabel}('\text{Time (sec)}') \\
\text{grid on} \\
\text{wf} &= [\text{BST23}_{\text{kd}}\_\text{part1oms}(7,:) \text{ BST23}_{\text{kd}}\_\text{part2oms}(7,:)]; \\
\text{subplot}(3,2,4), \text{plot}(\text{time}, \text{wf}) \\
\text{title}('\text{Flywheel Velocity for upshift 2 - 3 kickdown}') \\
\text{ylabel}('\text{Rotational Speed (rad/sec)}')
K.3 Hybrid Controller B-IST 1-2 Upshift

K.3.1 Case Decreasing Driveshaft Torque

close all
clear all

ts = 1e-2;
t1 = [0 0.5];
t2 = [0.5 1.5];

ratio1 = 1/3.308;
ratio2 = 1/1.913;
rd = 1/3.737;
rgn = 0.7;
z = 2.1155;
rp = (z+1)*rgn/z;
Jic_e = 0.12;
Jcl_e = 0.034;
Je = Jic_e + Jcl_e;
Jf = 0.08;
Jcl_p = 0.003;
Jtr_p = 0.0027;
Jp = Jcl_p + Jtr_p;
Js = 0.0028;
Jv = 110;
Jps1 = Js + Jp/(ratio1^2);
Jps2 = Js + Jp/(ratio2^2);
k = 6000;
d = k/1000;
bv = 10;
mub = 1;
muc1 = 1;

Tds_init1 = 1600;
Tds_2 = 850;
Te_init1 = Tds_init1*ratio1*rd;
Tb_init1 = 0.1;
dphi_init1 = Tds_init1/k;
Tc1_init1 = ((-Jf*Te_init1+rgn*rd*Jf*Tds_init1+rgn*Jf*Te_init1-rgn^2*rd*Jf*Tds_init1)*z^2
+rgn*Je*Tb_init1+rgn^2*Jps1*Tb_init1)*z-Te_init1*rgn^2*Jps1
K.3. HYBRID CONTROLLER B-IST 1-2 UPSHIFT

\[-rgn^2*rd*Tds_init1*Je)/((-ratio2*Jf+rgn*Jf+rgn*ratio2*Jf-rgn^2*Jf)*z^2
\]
\[-rgn^2*Jps1*ratio2-rgn^2*Je)*ratio2;\]

\[wv_init1 = 40;\]

\[x0_1 = [Te_init1;Tb_init1;Tcl_init1;dphi_init1;wv_init1/(ratio1*rd);wv_init1/rd;wv_init1];\]

\[wf_init1 = (z+1)/rp*x0_1(6)-z*x0_1(5);\]

\[BST12_kd_part1sys = mld('BST12_kd_part1',ts)\]

\[clear COSTS\]

\[COSTS.y = diag([50]); %[Driveshaft_Torque]\]

\[COSTS.x = diag([1 100]); %[Brake_Force Clutch_Force]\]

\[N = 10;\]

\[wemin = 1.01*84;\]

\[wemax = 0.99*700;\]

\[dphimin = 0.99*-1;\]

\[dphimax = 0.99*1;\]

\[Fnbmin = 0;\]

\[Fnbmax = 0.99*250;\]

\[Fnclmin = 0.01;\]

\[Fnclmax = 0.99*250;\]

\[Temin = 0.01;\]

\[Temax = 0.99*145;\]

\[Fnincmin = 0.99*-500;\]

\[Fnincmax = 0.99*800;\]

\[Teincmin = 0.99*-400;\]

\[Teincmax = 0.99*500;\]

\[wfmin = -1000;\]

\[wfmax = -200;\]

\[gammamax = 1000;\]

\[gammamin = -1000;\]

\[Tdsmin = 0;\]

\[Tdsmax = 2000;\]

\[clear LIMITS\]

\[LIMITS.umin = [Teincmin,Fnincmin,Fnincmin];\]

\[LIMITS.umax = [Teincmax,Fnincmax,Fnincmax];\]

\[LIMITS.xmin = [Temin,Fnbmin,Fnclmin,dphimin,wemin,ratio1*wemin,ratio1*rd*wemin];\]

\[LIMITS.xmax = [Temax,Fnbmax,Fnclmax,dphimax,wemax,ratio2*wemax,ratio2*rd*wemax];\]

\[LIMITS.ymin = [wfmin,gammamin,Tdsmin];\]

\[LIMITS.ymax = [wfmax,gammamax,Tdsmax];\]

\[clear REFSIGNALS\]

\[REFSIGNALS.y = [3]; %[Driveshaft_Torque]\]

\[REFSIGNALS.x = [2 3]; %[Brake_Force Clutch_Force]\]

\[ContrBST12_kd_part1 = hybcon(BST12_kd_part1sys,COSTS,N,LIMITS,REFSIGNALS)\]

\[try\]
\[sim('SIMBST12_kd_part1',t1)\]

\[catch\]
\[errmsg1 = lasterr\]

\[end\]

\[load BST12_kd_part1inputs\]
load BST12_kd_part1deltaphi
load BST12_kd_part1omegas
load BST12_kd_part1DriveTorque

timepart1 = BST12_kd_part1inp(1,:);

inputspart1 = BST12_kd_part1inp(2:4,:);
figure
subplot(3,2,[1 2]), plot(timepart1,inputspart1)
title('Inputs for upshift 1 - 2 kickdown PART I')
legend('Engine Torque (Nm)','Normal Force Brake (N)','Normal Force Clutch (N)', 'Location', 'Best')
xlabel('Time (sec)')
grid on

omegaspart1 = BST12_kd_part1oms([2 3],:);
subplot(3,2,3), plot(timepart1,omegaspart1)
title('Rotational Velocities for upshift 1 - 2 kickdown PART I')
legend('Engine-side velocity','Primary Shaft velocity', 'Location', 'Best')
ylabel('Rotational Speed (rad/sec)')
xlabel('Time (sec)')
grid on

wfpart1 = BST12_kd_part1oms(7,:);
subplot(3,2,4), plot(timepart1,wfpart1)
title('Flywheel Velocity for upshift 1 - 2 kickdown PART I')
ylabel('Rotational Speed (rad/sec)')
xlabel('Time (sec)')
grid on

DriveTorquepart1 = BST12_kd_part1Tds(2,:);
subplot(3,2,[5 6]), plot(timepart1,DriveTorquepart1)
title('Driveshaft Torque for upshift 1 - 2 kickdown PART I')
ylabel('Torque (Nm)')
xlabel('Time (sec)')
grid on

if ~exist('errmsg1','var')
    Te_init2 = BST12_kd_part1inp(2,end);
    Tb_init2 = BST12_kd_part1inp(3,end);
    Tds_init2 = BST12_kd_part1Tds(2,end);
    dphi_init2 = BST12_kd_part1dphi(2,end);
    Tcl_init2 = BST12_kd_part1inp(4,end);
    Tds_2 = 850;
    Te_2 = Tds_2*ratio2*rd;
    we2 = BST12_kd_part1oms(2,end);
    ws2 = BST12_kd_part1oms(5,end);
    wv2 = BST12_kd_part1oms(6,end);
    gamma_init2 = we2 - ws2/ratio2;
    x0_2 = [Te_init2;Tb_init2;Tcl_init2;dphi_init2;we2;ws2;wv2];
    BST12_kd_part2sys = mld('BST12_kd_part2',ts)
clear COSTS
COSTS.y = diag([1 50]);  %[Relative_Velocity Driveshaft_Torque]
COSTS.x = diag([20]);   %[Brake_Force]
COSTS.u = diag([1 1 1]);
N = 10;

wemin = 1.01*84;
wemax = 0.99*700;
dphimin = 0.99*-1;
dphimax = 0.99*1;
Fnbmin = 0.01;
Fnbmax = 0.99*250;
Fnclmin = 0.01;
Fnclmax = 0.99*200;
Temin = 0.01;
Temax = 0.99*145;
Fnincmin = 0.99*-500;
Fnincmax = 0.99*800;
Teincmin = 0.99*-400;
Teincmax = 0.99*500;
wfmin = -500;
wfmax = 0;
gammamin = -100;
gammamax = wemax*(ratio2-ratio1);
Tdsmin = 0;
Tdsmax = 2000;
clear LIMITS
LIMITS.umin = [Teincmin,Fnincmin,Fnincmin];
LIMITS.umax = [Teincmax,Fnincmax,Fnincmax];
LIMITS.xmin = [Temin,Fnbmin,Fnclmin,dphimin,wemin,ratio2*wemin,ratio2*rd*wemin];
LIMITS.xmax = [Temax,Fnbmax,Fnclmax,dphimax,wemax,ratio2*wemax,ratio2*rd*wemax];
LIMITS.ymin = [wfmin,gammamin,Tdsmin];
LIMITS.ymax = [wfmax,gammamax,Tdsmax];
clear REFSIGNALS
REFSIGNALS.y = [2 3];  %[Relative_Velocity Driveshaft_Torque]
REFSIGNALS.x = [2];    %[Brake_Force]

ContrBST12_kd_part2 = hybcon(BST12_kd_part2sys,COSTS,N,LIMITS,REFSIGNALS)
try
    sim('SIMBST12_kd_part2',t2)
catch
    errmsg2 = lasterr
end
load BST12_kd_part2inputs
load BST12_kd_part2deltaphi
load BST12_kd_part2omegas
load BST12_kd_part2DriveTorque

close all
time = [BST12_kd_part1inp(1,:) BST12_kd_part2inp(1,:)];
APPENDIX K. HYBRID CONTROL B-IST

\[
\text{inputs} = \begin{bmatrix} \text{BST12\_kd\_part1inp}(2:4,:) \\ \text{BST12\_kd\_part2inp}(2:4,:) \end{bmatrix};
\]
\begin{figure}
\begin{align*}
\text{subplot}(3,2,[1 2]), & \quad \text{plot}(\text{time},\text{inputs}) \\
\text{title}(& \quad \text{Inputs for upshift 1 - 2 kickdown}) \\
\text{legend}(& \quad \text{'Engine Torque (Nm)', 'Normal Force Brake (N)', 'Normal Force Clutch (N)'}, \\
& \quad \text{'Location', 'best'}) \\
\text{xlabel}(& \quad \text{'Time (sec)'}) \\
\text{grid} & \quad \text{on}
\end{align*}
\end{figure}

\[
\text{omegas} = \begin{bmatrix} \text{BST12\_kd\_part1oms}(\{2 3\},:) \\ \text{BST12\_kd\_part2oms}(\{2 4\},:) \end{bmatrix};
\]
\begin{figure}
\begin{align*}
\text{subplot}(3,2,3), & \quad \text{plot}(\text{time},\text{omegas}) \\
\text{title}(& \quad \text{Rotational Velocities for upshift 1 - 2 kickdown}) \\
\text{legend}(& \quad \text{'Engine-side velocity', 'Primary Shaft velocity', 'Location', 'best'}) \\
\text{ylabel}(& \quad \text{'Rotational Speed (rad/sec)'}) \\
\text{xlabel}(& \quad \text{'Time (sec)'}) \\
\text{grid} & \quad \text{on}
\end{align*}
\end{figure}

\[
\text{wf} = \begin{bmatrix} \text{BST12\_kd\_part1oms}(7,:) \\ \text{BST12\_kd\_part2oms}(7,:) \end{bmatrix};
\]
\begin{figure}
\begin{align*}
\text{subplot}(3,2,4), & \quad \text{plot}(\text{time},\text{wf}) \\
\text{title}(& \quad \text{Flywheel Velocity for upshift 1 - 2 kickdown}) \\
\text{ylabel}(& \quad \text{'Rotational Speed (rad/sec)'}) \\
\text{xlabel}(& \quad \text{'Time (sec)'}) \\
\text{grid} & \quad \text{on}
\end{align*}
\end{figure}

\[
\text{DriveTorque} = \begin{bmatrix} \text{BST12\_kd\_part1Tds}(2,:) \\ \text{BST12\_kd\_part2Tds}(2,:) \end{bmatrix};
\]
\begin{figure}
\begin{align*}
\text{subplot}(3,2,[5 6]), & \quad \text{plot}(\text{time},\text{DriveTorque}) \\
\text{title}(& \quad \text{Driveshaft Torque for upshift 1 - 2 kickdown}) \\
\text{ylabel}(& \quad \text{'Torque (Nm)'}) \\
\text{xlabel}(& \quad \text{'Time (sec)'}) \\
\text{grid} & \quad \text{on}
\end{align*}
\end{figure}

end
Appendix L

Explicit Controller for Linear Clutch Model

L.1 M-file

% Dry Clutch Engagement
% (C) 2000-2003 by A. Bemporad
% edited 2005 by J. Kusters

close all clear all

Je=0.2; % Engine inertia
Jv=0.8; % Vehicle-side inertia
bv=0.67; % Vehicle-load coefficient
be=0.03; % Engine-load coefficient
mu=1; % Friction coefficient

delta_Fn_max = 800; % Maximal Gradient of Clutch Normal Force
dt = 1e-2; ts = 0; te = 1;

% MPC Prediction Model

% Unlocked model
% Je*we_dot = Te-Tc-be*we
% Jv*wv_dot = Tc-bv*wv
% Tc = mu*Fn

% The engine torque T_e is modeled as a step disturbance
%
% states = [we, we-wv, Te]'
% inputs = [Fn]'
% outputs = [we, we-wv]'

A=[-be/Je, 0, 1/Je;
   -be/Je+bv/Jv, -bv/Jv, 1/Je;]
APPENDIX L. EXPLICIT CONTROLLER FOR LINEAR CLUTCH MODEL

\[ \begin{bmatrix} 0,0,0 \end{bmatrix}; \]
\[ \begin{bmatrix} -\mu/Je; \\
-\mu*(1/Jv+1/Je); \\
0 \end{bmatrix}; \]
\[ \begin{bmatrix} C= [1, 0, 0,; \\
0, 1, 0] \end{bmatrix}; \]
\[ D=\text{zeros}(2,1); \]
\[ \text{sys}=\text{ss}(A,B,C,D); \text{linclutch}=\text{c2d}(\text{sys}, dt); \]
\[ % \begin{bmatrix} A,B,C,D \end{bmatrix}=\text{ssdata}(\text{model}); \]

% Constraints
\[ \text{dumin}=-\text{delta}_Fn_{\text{max}}*dt; \text{dumax} = \text{delta}_Fn_{\text{max}}*dt; \]
\[ \text{umax} = [180]'; \text{umin} = [0]'; \text{ymin} = [80,0]'; \text{ymax} = [150,150]'; \text{clear limits limits.dumin=umin; limits.dumax=dumax; limits.umax=umax; limits.umin=umin; limits.ymin=ymin; limits.ymax=ymax}; \]
\[ \text{clear interval cost cost.T} = \text{diag([0]); cost.} \]
\[ \text{S}=\text{diag([1 20]); \%[deviation of engine velocity, deviation of relative velocity]} \]
\[ \text{interval.N}=10; \text{interval.Nu}=1; \text{interval.Ncy}=2; \text{interval.Ncu}=1; \]
\[ %\text{cost.rho}=\text{inf}; \]
\[ \text{linclutchcon} = \text{lincon(linclutch, 'track', cost, interval, limits)} \]
\[ \text{range.xmax} = [150;150;150]; \text{range.xmin} = [80;0;50]; \text{range.umax} = \]
\[ [180]; \text{range.umin} = [0]; \text{range.refymin} = [90;0]; \text{range.refymax} = \]
\[ [140;1]; \]
\[ %\text{options.uniteeps}=1e-5; \]
\[ %\text{options.fixref.x}=[]; \]
\[ %\text{options.fixref.u}=[]; \]
\[ %\text{options.fixref.y}=[1 2]; \]
\[ %\text{options.valueref.y}=[100;0]; \]
\[ \% \text{figure} \]
\[ % \text{load omslc.mat} \]
\[ % \text{load inplc.mat} \]
\[ % \text{subplot}(2,1,1), \text{plot(omegaslincon(1,:)),omegaslincon(2,:),omegaslincon(1,:),omegaslincon(3,:))} \]
\[ % \text{grid on} \]
\[ % \text{legend('Engine','Vehicle-side')} \]
\[ % \text{xlabel('Time (sec)')} \]
\[ % \text{ylabel('Rotational velocity (rad/sec)')} \]
\[ % \text{title('State variables')} \]
\[ % \text{subplot}(2,1,2), \text{plot(inputslincon(1,:),inputslincon(2,:))} \]
\[ % \text{grid on} \]
\[ % \text{xlabel('Time (sec)')} \]
\[ % \text{ylabel('Clutch Normal force (N)')} \]
\[ % \text{title('Optimal input')} \]
L.2. **Explicit Form**

\[ \text{ExpC} = \text{expcon(linl clutchcon, range)}; \%
\text{ options) } \]

% Write solution to latex file
texfile='dry clutch'; latex(ExpC, texfile);

Tin0=120; [h,E1]=plotsection(ExpC,[3 4 5 6],[Tin0,0,120,0],[1,0],0);

% Now E1 is defined over the state space (we, we-vv). Change
% the partition to (we,vv)
M=[1 0; 1 -1]; % Transformation matrix [we,vv]=M*[we,we-vv]
E1.H=E1.H*inv(M); % H*[we,we-vv]=H*inv(M)*[we,vv]
plot(E1);
title(sprintf('Section with T_e=5.2f, ref=[120,0]',...
              Tin0))
axis([75 160 0 160]); xlabel('\omega_e');
ylabel('\omega_v');

L.2  Explicit Form

\[
\begin{align*}
\text{ExpC} &= \text{expcon(linl clutchcon, range)}; \%
\text{ options) } \\
\text{Texfile} &= 'dry clutch'; \text{ latex(ExpC, texfile); } \\
\text{Tin0} &= 120; [h,E1]=\text{plotsection(ExpC,[3 4 5 6],[Tin0,0,120,0],[1,0],0);} \\
\text{M} &= [1 0; 1 -1]; \text{ Transformation matrix [we,vv]=M*[we,we-vv]} \\
E1.H &= E1.H*\text{inv(M); } \%
H*[we,we-vv]=H*\text{inv(M)*[we,vv]} \\
\text{plot(E1); title} = \text{sprintf('Section with T_e=5.2f, ref=[120,0]',...} \\
\text{Tin0)) axis([75 160 0 160]); \text{ xlabel('\omega_e'); ylabel('\omega_v');} \\
\end{align*}
\]