A University of Sussex DPhil thesis

Available online via Sussex Research Online:

http://sro.sussex.ac.uk/

This thesis is protected by copyright which belongs to the author.

This thesis cannot be reproduced or quoted extensively from without first obtaining permission in writing from the Author

The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the Author

When referring to this work, full bibliographic details including the author, title, awarding institution and date of the thesis must be given

Please visit Sussex Research Online for more information and further details

OPTIMAL CONTROL OF A FLYWHEEL-BASED AUTOMOTIVE KINETIC ENERGY RECOVERY SYSTEM

by

LUIS ALEJANDRO PONCE CUSPINERA

SUBMITTED FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF SUSSEX
SCHOOL OF ENGINEERING AND INFORMATICS
DEPARTMENT OF ENGINEERING AND DESIGN

APRIL 2013

ABSTRACT

This thesis addresses the control issues surrounding flywheel-based Kinetic Energy Recovery Systems (KERS) for use in automotive vehicle applications. Particular emphasis is placed on optimal control of a KERS using a Continuously Variable Transmission (CVT) for volume car production, and a wholly simulation-based approach is adopted. Following consideration of the general control issues surrounding KERS operation, a simplified system model is adopted, and the scope for use of optimal control theory is explored. Both Pontryagin's Maximum Principle, and Dynamic Programming methods are examined, and the need for numerical implementation established. With Dynamic Programming seen as the most likely route to practical implementation for realistic nonlinear models, the thesis explores several new strategies for numerical implementation of Dynamic Programming, capable of being applied to KERS control of varying degrees of complexity. The best form of numerical implementation identified (in terms of accuracy and efficiency) is then used to establish via simulation, the benefits of optimal KERS control in comparison with a more conventional non-optimal strategy, showing clear benefits of using optimal control.

DECLARATION

This research has been carried out in the Department of Engineering and Design, School of Engineering and Informatics, at the University of Sussex. I hereby declare that this thesis has not been submitted, either in the same or different form, to this, or any other University for a degree.

Luis Alejandro Ponce Cuspinera

ACKNOWLEDGEMENT

I wish to express appreciation to those who contributed and supported me during this research project.

Special thanks to Dr. Julian Dunne for his guidance and support, for his patience and for encouraging me when most needed.

Thanks to my parents, Norma and Urbano, who have been a life example, always loving and supporting.

Thanks to my sister, and best friend, Ana Laura. Your unconditional love has always lifted me up. Thanks to my niece Valentina.

Thanks to “Consejo Nacional de Ciencia y Tecnología” (CONACYT) for providing me with financial support.

Thanks to my friends who contributed to the happy moments and for their most valuable companionship.

Thanks to my life mentors.

Thanks to God.

TABLE OF CONTENTS

NOMENCLATURE

T_w	traction wheel torque
T_{wf}	traction wheel friction torque
T_f	flywheel torque
T_{ff}	flywheel friction torque
J_w	wheel moment of inertia
J_f	flywheel moment of inertia
θ_w	wheel angular displacement
θ_f	flywheel angular displacement
G	gear ratio
$\bar{x}$	a vector of state variables
$\bar{u}$	a vector of control variables
m_v	vehicle mass
r_w	traction wheel radius
K_v	viscosity friction factor
K_{MB}	magnetic bearings friction factor
J	performance index
H	Hamiltonian
$\overline{\psi}$	co-state equations
f_0	cost associated to the states, control or time behaviour
f_i	state equations
c	damping coefficient
k	stiffness coefficient, discrete time steps
m	mass
α	weighting factor (cost related, control)
u_g	control gradient
ω_w	wheel rotational speed
q	least square evaluation function

β	weighting factor (cost related, states)
ς	weighting factor (cost related, control)
Kp	proportional gain
Kd	differential gain
Ki	integral gain

ABBREVIATIONS

KERS	Kinetic Energy Recovery System
CVT	Continuously Variable Transmission
EV	Electric Vehicle
HEV	Hybrid Electric Vehicle
P	Proportional
PI	Proportional-Integral
PID	Proportional-Integral-Derivative
TPBV	Two-point Boundary-value

1. INTRODUCTION

The increasing concerns about environmental protection, through the reduction of pollution, creates a need for more environmentally friendly devices. This has a great impact on the road transport sector, which in total currently represents around 25% of the carbon dioxide emissions throughout the world. With stricter legislations for lower CO_2 and harmful emissions, and an awareness that energy resources are finite, this gives increasing motivation for manufacturers, designers and researchers to improve and optimize power generation and transmission systems. In the automotive area this means the development and application of new materials, the optimization of the engine and powertrain systems, and the rapid evolution of hybrid and electric vehicles that could be introduced in the near future without any significant loss of vehicle performance. These necessities have given rise to the creation of several new devices, techniques and powertrain configurations. One of these devices is the Kinetic Energy Recovery System (KERS), which is intended to recover and store kinetic energy of a vehicle instead of wasting it as heat during braking. This is indeed a form of regenerative braking. KERS are not new, but with increasing interest and development in hybrid and electric vehicles, combined with the motivation to improve efficiency of combustion-engine-powered vehicles, the use of KERS is an area of growing focus. The recent inclusion for example of KERS in formula one cars not only shows a growing interest in KERS development but also makes its use more exciting. However the use of KERS in motor sport is largely for enhanced performance not for efficiency improvements. The ultimate benefit of KERS may be in volume car usage.

As part of this introduction, a description of KERS and a brief assessment is shortly presented. Different KERS possibilities are identified and the reasons for selecting a study of a Flywheel-based KERS with a Continuously Variable Transmission (CVT) are highlighted. Relevant information about both flywheels as storage systems, and CVTs as transmission systems, is presented. In the assessment of the state of the art of KERS, the importance of the control strategies used to transfer the energy to the storage device is manifest. For this reason, a particular interest is shown in the currently-used control strategies.

Since one of the major objectives in the regenerative process, is to maximise the energy recovered, the possibility of using an optimal control strategy to achieve this is worthy of detailed examination. Therefore, the use of optimal control in automotive systems and particularly in KERS is explored. Indeed as part of the literature review presented shortly in section 1.4, the most relevant publications in regenerative braking, flywheels as storage systems, CVT models and control, optimal control in automotive systems, and control of KERS are examined. With it, the motivation of using optimal control to improve performance and efficiency for KERS becomes evident.

The research addressed in this thesis is particularly concerned with the optimal control of a Flywheel-based KERS with a CVT. The modelling of the system plays an important role for control, therefore, suitable models for this application are evaluated and presented. The optimal control methodologies considered are the Classical tools of Pontryagin's Maximum Principle, and discrete Dynamic Programming. These methodologies are initially implemented via simulation and the results highlight the need for substantial development. It quickly becomes apparent that the approach to the improvement needed is the development of a modified version of Dynamic Programming.

In the next three subsections an overview is given of Kinetic Energy Recovery Systems in general, followed by focus on flywheel-based KERS. Of course flywheels and transmissions are not new, thus something of an overview is also given of these. The control issues, in particular the optimal control possibilities, are then examined leading into the review of the literature on all these topics.

1.1 Vehicle Kinetic Energy Recovery Systems (KERS)

A Kinetic Energy Recovery System (KERS) is a system capable of recovering the kinetic energy from a moving device by reducing its speed and storing the energy in a 'reservoir'. In an automotive application, the term KERS refers to a system that can recover the vehicle kinetic energy while braking, instead of wasting it as heat, this process is normally known as Regenerative Braking.

There are two main advantages of using a regenerative process: i) to extend the driving range (Cheng *et al*. 2011, Ye *et al*. 2010); and ii) to save energy consumption (Cheng 2011). These characteristics make a vehicle more efficient and increase its performance, which represents an improvement in fuel economy and a reduction in CO_2 emissions.

Although the concept of Regenerative Braking is relatively old it is only recently that it has captured more attention, especially as a result of increasingly strict vehicle emissions regulations, and the development of Hybrid Electric Vehicles (HEV) and Electric Vehicles (EV). This is why, the term Regenerative Braking is commonly associated with EV technology, and the term KERS or even ERS, is used as a more general energy management concept.

A KERS is formed by two subsystems: i) The transmission, normally consisting of a gear box, a Continuously Variable Transmission (CVT), electrical transmission or a motor-generator; and ii) the storage device (reservoir) which can be divided in two different groups: Electrical Storage Systems (ESS) and Mechanical Storage Systems (MSS).

The transmission is the connection between the wheels, engine and the storage system, also any additional devices between them. This has to be engaged at both ends to pump energy from the wheels and engine to the storage system. This subsystem may also be used to pump energy back to the vehicle, with the appropriate configuration, when needed. Three important characteristics of this subsystem are: i) high efficiency, ii) wide energy density range in order to store as much energy as possible; and iii) high power density, i.e. fast operation while braking or accelerating.

Turning to ESS, these include batteries, hydrogen fuel cells, and capacitor banks. They are good storage devices because they can store energy for a long time. The transmission subsystem for EES is not necessarily complex because for hybrid and electric vehicles, the motor is also used as a generator (Cheng *et al*. 2011, Inoue *et al*. 2010) and the vehicles transmission is the same for the KERS. Unfortunately they have a low operational range, in addition to

charge/discharge limitations. By contrast, MSS include springs, flywheels, and fluid reservoirs. They are lighter than ESS, have a relatively good charging rate and wider operational range, but they cannot generally store energy for long periods of time.

But the storage elements chosen for a KERS are closely-linked to the type of vehicle where the storage process is adopted depending on the different options available and the demands imposed. For example, HEVs and EVs normally use the main electric source as storage for the KERS; however for combustion engine powered vehicles, the use of an auxiliary source is needed. Lukic *et al*. (2008) and Van Mierlo *et al*. (2004) analyse the characteristics of different energy stores, including batteries, ultra capacitors, hydrogen fuel cells, and flywheels, when these are used in electric vehicles as the main energy store. From this analysis, it is clear that the lower cost, portability and storage capacity for long periods of time, batteries are the most prevalent energy storage system, especially as a main store. However, because of its limited charge/discharge rate, and the need for energy transformations that the regenerative process must undertake (Boretti 2010), using batteries is not the most efficient means of storage. Ultra capacitors and flywheels by contrast allow fast charge and discharge, making them more attractive secondary storage devices for use in high charge rates. In addition, reduced or non-energy transformations are required when using these two devices since electrical or mechanical energy (respectively) is stored.

Flywheels however are quite novel in automotive applications (Lukic *et al*. 2008), for this reason they have not been fully explored and are still very expensive. However, because they have a higher energy density and a higher energy capacity than Ultra Capacitors (Van Mierlo *et al*. 2004), they are becoming more popular in automotive applications.

Secondary storage means, and especially flywheels, require a transmission device to transfer the energy from the kinetic device. Therefore it is appropriate to consider the types of transmission available.

Transmission systems

Lechner *et al*. (1999) present very good and complete information about the different types of transmissions used in automotive engineering. They suggest that, transmissions can be categorized in many different ways. Two important categories however are: Geared transmissions, and Continuously Variable Transmissions. One of the motivations for the development of CVTs is the high efficiency they can achieve; but its main disadvantage, is the limited torque capacity they provide. But with high efficiency, and the huge effort being put into developing higher torque capacity CVTs, this makes it potentially a good future transmission system to be considered for KERS applications.

With the purpose of studying a KERS that is not exclusively for hybrid or electric vehicles, the next motivation is to explore technologies that are currently being considered for implementation in automotive applications, taking account of the benefits and possibilities that have been identified thus far. The selection of a Flywheel as a storage device and a CVT as a transmission system, as a subject of further study, has already been justified. But before fully reviewing the literature it is appropriate to focus on flywheel KERS, indeed on flywheels as energy stores in general.

1.2 Flywheel-based KERS

The concept of using flywheels as storage systems has been in existence for some years (Bolund *et al*. 2007), with the main applications being electric stabilizers in machines (Samineni *et al*. 2003), or electrical energy sources, or energy production units (Boukettaya *et al*. 2010). However, as Liu, H. *et al*. (2007) mention, flywheel characteristics have widened the possible applications into several areas such as in space, vehicle, or power sources. This is especially possible as a result of developments in materials and tribology, like the improvements in ball bearings and the development of magnetic bearings (Bolund *et al*. 2007).

It was mentioned earlier that HEVs and EVs use the main electric source to store the energy most of the time. However, advantages are accrued of using a flywheel combined with the electric sources, especially for rapid acceleration or

energy recovery high power density in charging. For example, Hua *et al*. (2009) use a flywheel as a second energy storage device in a hybrid vehicle.

In the case of combustion engine powered vehicles, flywheel-based KERS are being explored mainly in motor sport. The FIA regulations for the 2009 racing season allowed the use of regenerative braking systems. The company Flybrid Systems LLP, developed a mechanical KERS with a CVT that is suitable for both F1 vehicles and mainstream automotive applications (Cross *et al*. 2008). In addition to motorsport applications, Boretti (2010) presents a flywheel-based KERS with a CVT system in a compact car. Figure 1.1 shows a Flybrid system fitted to a production car.

Figure 1.1 Flybrid® 9013 hybrid system as fitted to the Jaguar XF demonstrator (http://www.flybridsystems.com/Roadcar.html)

Flywheel-based KERS are however not exclusive to automotive engineering. There are other applications where the kinetic energy is recovered and stored for later use, for example Ghedamsi *et al*. (2008) mention its use as an auxiliary source in a wind generator to store the extra electrical energy produced and also for improving the generator performance by working as a power regulator.

Flywheels

Whilst the subject of flywheels is being discussed it is appropriate to digress just to say something about flywheel materials and operating conditions. Liu, H. *et al*. (2007) and Bolund *et al*. (2007) state the importance of flywheel geometry and physical composition as a determining factor to its performance. Specific energy and energy density are two important features for this characterisation.

Flybrid Systems LLP (Cross *et al*. 2008) for example use a filament wound carbon composite flywheel. This composition provides higher energy densities and its shape produces higher specific energy levels, and allows it to reach high speeds up to 64 000 rpm. For efficiency and integrity reasons, the flywheel is enclosed in a vacuum housing that ensures containment in the event of failure. Boretti (2010) mentions that for series volume production cars, the speeds do not have to reach such levels because larger inertias can be used. Cross *et al*. (2008) also emphasize the importance of safety, and have performed various relevant crash tests to ensure the highest standards are met. An overview will now be given of the transmissions used with KERS.

Transmissions

Flywheel-based KERS mainly use two types of transmission: i) A Motor-generator (Bolund *et al*. 2007, Ghedamsi *et al*. 2008), and ii) CVT (Boretti 2010, Cross *et al*. 2008). A motor-generator transmission is commonly used when the main vehicle power source is electrical; and CVTs are normally chosen when a transmission must be added to the vehicle. The main reason for this is to avoid the energy transformation losses.

The term CVT covers various types of transmissions, Lechner *et al*. (1999) mention the most common: Pulley transmission, Toroidal transmission (or friction gear), Hydrostatic transmission, and Electric transmission.

A pulley transmission consists of two variable diameter pulleys joined by a belt. The diameter variability creates a continuous range for the input/output ratio. The toroidal transmission consists of two discs, and a couple of rollers. The rollers work as contact point between the discs, and with its rotation, the contact

point change the centre distance, allowing a continuous range in the gear ratio. A hydrostatic transmission transfers the power through liquid, and this is controlled by a variable displacement pump which regulates the flow used by a hydrostatic motor. The Electric transmission regulates the current using different types of electric or power electronic components.

The first two types of transmission are entirely mechanical, the third one is hydraulic and the last one is electric. The application of the CVT has a great influence on which is the most convenient to be used, especially to avoid energy conversion losses. Flybrid Systems LLP (Cross 2008) use a full toroidal CVT, but for most electric vehicle applications the motor-generator is used (Bolund *et al*. 2007, Ghedamsi *et al*. 2008).

There is a great deal of interest in increasing efficiency and power range in transmission systems. For friction-based transmissions this means the development of new materials technologies and control techniques. For the electric based transmissions the focus is mainly in the energy management strategies used.

Control

Information about the control used in Flywheel-based KERS is actually very limited or non-specific. Cross *et al*. (2008) and Boretti (2010) do not mention any particular strategy used for control purposes, however, their control descriptions fit into classical control categories (e.g. trial and error approach). Elsewhere Ghedamsi *et al*. (2008) discuss the use of a PID control strategy for a wind generator; which, owing to its configuration, is represented as a MIMO (multiple input multiple output) control system.

Owing to the limited number of citations on Flywheel-based KERS, it is appropriate to consider and review the control systems used in other types of KERS. Ye *et al*. (2010) mention some of the strategies previously used in energy recovery systems, these are: Variable structure control, fuzzy control, intelligent control, and neural network control; and they present a H_2/H_∞ control strategy. Cheng *et al*. (2011) use a genetic algorithm (GA) based neural

network as a control strategy, and also mentions other strategies used in the past, including: fuzzy PID, H_∞ control, and robust sliding mode control. Hua al. (2009) implement two frequency conversion control methods: i) Constant Torque Control, and ii) Constant Power Control. Inoue *et al*. (2010) use a variational method as an optimal control strategy. But it is noticeable that only in the most recent years modern control strategies have been used for energy recovery purposes. This has opened up a whole new research topic namely in how to control KERS. An advantageous way to do this, if it were possible in practice, would be via optimal control. It is worth stating here the main approaches available and whether these have been used to control KERS.

1.3 Optimal Control

The classical control approach, which has been suitable for many different applications with acceptable results, finds its limitations with the modern systems characteristics and the additional control requirements. On the other hand, modern control strategies (such as optimal control) aim to fulfil these new demands (Ogata 2002, Kirk 1998). Kirk (1998) emphasizes the difference in the performance measurement between classical and optimal control theories. For classical control systems the performance is normally measured in terms of time and frequency characteristics: “rise time, settling time, peak overshoot, gain and phase margin, and bandwidth”. An optimal control objective is “to determine the control signals that will cause a process to satisfy the physical constraints and at the same time minimize (or maximize) some performance criterion” Kirk (1998). This is normally achieved in terms of time or energy. In a KERS application, where the objective is to maximize the amount of stored energy, and to satisfy other systems requirements, the use of optimal control theory therefore seems a very promising option.

Optimal Control Theory is usually applied using two independent methodologies: Pontryagin’s Maximum Principle, and Dynamic Programming. The first approach is based on the Calculus of Variations, and the second is based on Bellman’s principle of Optimality. In the case of Dynamic Programming, it can be implemented either in continuous or discrete time, the latter being more frequently used owing to the mathematical complexity that can

arise when working in continuous time. For this same reason use of Pontryagin's Maximum Principle's is uncommon for complex systems applications.

These two methods, and the general optimal control theory, are presented in more detail later in this thesis, where a brief contrast with the classical control theory is also made. However, it is important to note at the start the extent which optimal control theory has been applied to KERS. There is no evidence of using the optimal control strategies in a flywheel-based KERS with a CVT. There have been some applications in regenerative braking optimization, and also in the optimal control of CVTs, but not in KERS with a CVT.

In the case of regenerative braking optimization, Hoon *et al*. (2006) use a classical control strategy to follow an optimal operation line (OOL) combined with a regenerative braking algorithm; and Mukhitdinov *et al*. (2006) present four different strategies to maximize different parameters, where one of them includes maximization of the energy stored.

Optimal Control of CVTs has been implemented by Pfiffner *et al*. (2003) proposing a numerical solution approximation using the optimization package DIRCOL; and also by Liu, J *et al*. (2007) which use a genetic algorithm to solve the nonlinear problem presented by the system model (obtained by using the predictive model CARIMA). A different type of optimization is presented by Youmin *et al*. (2009) where they find the optimal PID parameters for a CVT control system through a faster experimentation strategy.

None of the optimal control strategies mentioned actually deal with the use of Pontryagin's Maximum Principle or Dynamic Programming. However, Pérez *et al*. (2007) evaluate the possibilities of using these strategies for a power split in a hybrid electric vehicle, where they highlight the difficulties of implementing both Pontryagin's Maximum Principle, and Dynamic Programming, and propose using a form of Nonlinear programming code.

Although optimization processes can become very complicated, especially when dealing with complex systems, there is interest in applying optimal control strategies in a range of relevant automotive engineering problems. First it is necessary to consider relevant and recent publications.

1.4 Literature Review

Here the most relevant literature related to Flywheel-based KERS is reviewed. First, the literature associated with regenerative braking process and KERS are reviewed, followed by a focus on flywheels as storage systems. For the subject it is also appropriate to assess the state of the art and apparent direction of CVTs. Optimal control applications (particularly in automotive systems) are then reported followed by the current control strategies used in KERS. The relevant literature that does not fall neatly into any of these five categories, (usually because it overlaps more than one), is allocated into the most appropriate one based on the particular research application.

Regenerative Braking/KERS

At the time of relatively early implementation of regenerative braking there was some scepticism about the actual benefits of implementing the strategy, as mentioned by Wicks *et al*. (1997). But they initiated the process of good system modelling and pointed out the benefits of regenerative braking. By making an idealized assumption of best case scenarios, and considering the driving cycle of a public transport bus, their analysis showed potential economy savings of up to 59% in a year of operation. A more realistic figure is actually presented by Cross *et al*. (2008), showing energy savings up to 30%. These figures for example, show the motivation for working with KERS and explains why, from then on, the interest in them has been increasing.

Electric Vehicles

EVs bring together various branches of engineering and are now-state-of-the-art in the sense that huge effort has been specifically focused on the improvement of battery systems, electric machines, power electronics and energy management. The development of EVs goes hand in hand with the global concerns about environmental protection and the shortage of non-

renewable energy sources (Yang *et al*. 2011). For electric vehicles, the regenerative process is considered almost compulsory since all of the elements needed to recover energy already exist in the vehicle. Yang *et al*. (2011) mention the advantage of using the driving motor as a generator. The energy management study of an electric scooter is examined for a system consisting of a DC motor, a lead acid battery bank, ultra capacitors, and an electronic gearshift. The objective is to determine the gear shifting points based on torque and efficiency curves. The results of a simulation study are compared with experimental results showing a good performance but with only 7.5% difference of the total energy regenerated.

Hybrid Electric Vehicles

Karden *et al*. (2006) analyse the future for HEVs, which use batteries as energy storage systems. This publication emphasizes the need for improvements in the power supply system comparing different hybrid configurations, and pointing out the use of regenerative braking as an important characteristic in HEVs development.

There are other types of Hybrid Vehicle that allow regenerative braking such as the air hybrid. Fazeli *et al*. (2011) present a double tank air hybrid engine vehicle, where the engine is used as compressor for the regenerative process. The upgrade from one conventional tank to two tanks (one of smaller size) is reported as improving the efficiency of the regenerative braking process. The particular types of vehicles described in this publication are relatively recent (1999) where the regenerative process was considered from the initial design stage.

Combustion engine powered vehicles

The work presented by Cross *et al*. (2008) is the most representative of all flywheel-based KERS. The system designed by Flybrid Systems LLP has been used in the F1 competitions and the implementation into mainstream vehicles is mentioned throughout. Despite having implemented a full design into real life applications, the need for new developments is clearly stated, especially in the areas of: i) failure modes of the rotating mass, ii) the method of control, and iii)

the transmission of energy to and from the flywheel. They emphasize the benefits of using a purely mechanical system to avoid energy conversions. A full toroidal CVT is used as the energy transfer medium. Two clutches (one connecting the driving shaft to the CVT, and one connecting the CVT to the flywheel) assist in achieving a wider gear ratio range. For this reason, a lot of the control is done with the clutches. The F1 KERS has an efficiency higher than 70% (round trip) and a dynamic response of 50 ms. Special attention is given to safety conditions which are a major concern, especially for mainstream vehicles.

Flywheel Storage Systems

The characteristics of a flywheel make it a great prospect as a storage system, especially when energy needs to be rapidly stored for short periods of time. This is because flywheels store energy in a kinetic form by rotational movement. Liu, H. *et al*. (2007) mention the most attractive characteristics i.e.: high efficiency, long cycle life, wide operating temperature range, freedom from depth-of-discharge effects, and high power and high energy density relative to the alternatives. In this publication, the key factors for delivering high performance are identified, namely: the material, geometry, length, and the type of bearings. Also the most important concerns are considered i.e.: safety, and energy losses. In a very similar publication, Bolund *et al*. (2007) include an extra consideration of a motor-generator that works as a converter/transmission system, stating the importance of power electronics in the matter.

In order to better visualize the main characteristics for the most common auxiliary storage systems, information has been summarized into Table 1.1. (Lukic et al. (2008), Fabien (2009), Dumé (2010)).

Device	Energy Density [Wh/kg]	Rates of Charge
Lead acid Battery	30	Very Low
Nickel-Cadmium Battery	31	Low
Supercapacitors	10-80	Very High
Li-ion Battery	200	Average
Composite Flywheels	100-1000	Very High

Table 1.1 – Energy densities for the most common auxiliary storage systems, and qualitative information about their rates of charge.

Flywheels have been used for a long time as stabilizers or regulators, in both mechanical and electrical applications. Such is the case presented by Smineni *et al*. (2003), where a flywheel is used to mitigate voltage sags. An induction machine is used as converter/transmission with a power converter interface and the model is given. A voltage reference is set up, where the flywheel operates as a source or storage depending on the negative torque command.

Although at the present time, the use of a flywheel as storage system in HEVs and EVs is not very common, the clear advantages of a flywheel's rapid response over batteries, make a combined storage device an attractive option to be considered. The hybrid vehicle presented by Hua *et al*. (2009) is an example of such system. In this publication, the energy recovered can be stored in a battery or a flywheel. When the recovered energy produces heavy current, the energy is stored in the flywheel, otherwise the energy is stored in the battery. The flywheel can operate under constant torque or constant power control depending on its rotational speed. The system is configured so that the battery works in the highest efficiency region. Combined storage systems like this make the most of the properties of storage devices, although it would not be practical to implement such a system for purely mechanical systems, because adding a set of batteries would significantly increase vehicle weight with a corresponding increase in carbon emissions.

Bearing friction

The Stribeck curve makes profound contribution to Tribology when introducing the curve of the coefficient of friction as a function of speed (see “The Stribeck curve”, Lu *et al*. 2005). Stribeck’s contributions were later extended to various lubrication regimes which led to the introduction of the dimensionless Sommerfeld number (Lu *et al*. 2005).

Stribeck curves show a nonlinear relationship between the friction coefficient and the rotational speed of bearings especially at lower speeds. In the literature, it can be observed that after a certain ‘breakaway point’ the system behaviour is almost linear (see for example Lu *et al*. 2005). However, it is important to consider that the flywheel reaches very high rotational speeds which is heavily influenced by the bearing friction behaviour. In this last publication, various experimental results for different journal bearing lubrication materials and loads are presented, as well as the temperature effects. Harnoy (2003) mentions the close relationship between ball bearings and journal bearings, stating that the main difference is the breakaway point, which happens earlier in rolling bearings due to a lower friction interaction.

The friction model for magnetic bearings is more complicated, Schweitzer (2002) shows various aspects that contribute to the friction forces acting on magnetic bearings. The most relevant losses are due to hysteresis, air, and eddy currents. The combination of these friction forces produce a nonlinear relationship between rotational speed and friction coefficient.

Continuously Variable Transmissions (CVT)

Most CVT research done so far is related to vehicle driveline where most of its developments are explored. CVT applications in KERS are very recent, in fact the state-of-the-art can be drawn mainly from the most recent standard application reported. There are two important areas to be considered for CVTs, one is CVT modelling, and the other one is the CVT control. Although for control purposes it is necessary to have an accurate model of the CVT, models are very much simplified to give a reasonable representation of system behaviour.

CVT Models

The simplest model for a CVT can be found in Powell *et al*. (2000), where a full hybrid electric vehicle with a CVT is modelled. The transient behaviour of the transmission is analysed and reported, where the model of the electric motor and the lead-acid battery are also given. (This model is actually found to be very useful when considering control applications, especially when advanced control techniques are used).

A different approach to a V-belt CVT is reported by Assadian *et al*. (2001), where the friction models consider the angle of contact at each pulley, as well as including the hydraulic system that controls the pulleys aperture. Stiffness and damping, in the shaft that connects the load, are also included. The full drive representation is obtained by assembling individual 'bond graphs' for the different parts of the system. The control analysis is limited to the hydraulic part of the system not to the overall behaviour of the system.

Pfiffner *et al*. (2003), and Setlur *et al*. (2003), present similar CVT models which are more detailed than the one presented by Powell *et al*. (2000). Pfiffner *et al*. (2003) focus more on the optimal control approach and do not follow the model in much detail. A numerical optimization approach is proposed and the solution is obtained with the help of the optimization package DIRCOL. The difficulty of solving for the optimal control online is clearly highlighted, and for this reason the optimization is done offline. However, they show that, from a simplified optimal control strategy, this can be tracked by using a classical control strategy. Indeed an improvement in the fuel consumption was achieved. Setlur *et al*. (2003) include an engine model into the full simplified representation of the system. The full model representation differs from the earlier ones, because here, the CVT is used to split the power where both CVT ends are connected to a planetary gear train and the output, giving more flexibility to the engine operation without producing changes in wheel speed. The control objective is to drive the vehicle wheels to follow a desired trajectory using the 'backstepping' strategy.

A more complete CVT model that could still be used for control purposes is presented by Müller *et al* (2001). There are two main contributions over the previous models. The first one is that it mentions coupling between engine speed and the CVT output torque, where they propose compensation of the CVT torque by changing the engine torque. The second one is the inclusion of the electrohydraulic valve that controls the pulley aperture. The paper shows a nonlinear relationship between the gear ratio and the axial displacement of the pulley. A general regression neural network is used to identify the overall nonlinearity of the system.

There is a good deal of interest in CVT analysis and design, which has considerably improved the development of the CVT. In the case of the pulley-based CVT, research has focused on the friction contact between the belt and the pulleys. An example of this is the multi-band layered belts, which are examined by Kong *et al*. (2008). A complex friction model is shown, where this model considers various factors like the viscosity of the lubricant, the thickness of the oil-film, the band width, and the relative sliding speed. The governing equations for the layers are shown, and using steady state conditions, a solution using a boundary value problem solver is found. The results show the distribution of the traction force depends on the position of the pulley. Redistribution of the load using layered-belts, improves the performance of a CVT. Carbone *et al*. (2007), and Srivastava *et al*. (2008), present more detailed models, where they include various factors like the contact angle, the sliding velocity, and the geometrical relations. It can be seen from these publications that that the complete mathematical representation of a CVT is very complex, even when the CVT is operating in a steady state condition.

A more elaborate version of a pulley CVT is the ‘infinitely variable’ transmission (IVT). With the assistance of a planetary gear set, the CVT is capable of reaching an infinite gear ratio range. This type of transmission is analysed and compared with a CVT by Mangialardi *et al*. (1999). In this publication the kinematic analysis of an IVT is shown and explained. It explores two different power flows for two different IVT configurations (i.e. both series and parallel), and shows the parameters determination. Both power flows are compared with

a standard CVT. The results show that for different torques, different power flows have a higher efficiency. In this publication the power flow series system No. 1, is the most efficient provided the operational range stays at higher gear ratios.

Most of the research has focused on pulley-based CVTs, however there is an increasing interest in the toroidal transmission since they potentially offer greater torque capacity with fast ratio change (Osumi *et al*. 2002). In this publication, a half-toroidal CVT is explored, and a geared-neutral system is developed. A combined simplified half-toroidal CVT model and hydraulic piston model is presented. The model is validated by installing a torque converter into the vehicle. The geared neutral system allows shifting between two different modes (low and high speeds) and it shows improvements over a more conventional system.

Mensler *et al*. (2006) use a continuous time identification method to model a toroidal CVT. The main advantage of using this method is that the model is obtained in continuous time. The results obtained using a simulation model and real vehicle experiments show very good agreement. This demonstrates that when the system model structure is known, a very efficient approach to complete the model is the use of parameter or system identification methods. This is also very convenient when a control strategy is to be applied to complex, or difficult to model, systems.

Control

There are different approaches to solve the control problem for a nonlinear model that describes CVT transient behaviour. However, most solution methods are concerned with optimal operation. Guzzela *et al*. (1995) present a feedback linearization strategy to represent a CVT model where a linear “kick-down” controller is used to achieve time optimal control. The resulting control has a fast response and keeps the system within the desired bounds. Gauthier *et al*. (2012) also use a feedback linearization design. They describe, that despite the fact of having two inputs to adjust the gear ratio, it can be simplified to just one input (assuming that only one of the pulleys moves). A PI (Proportional Integral)

compensator is used to eliminate the steady-state error, and a real time wear adaptation is achieved. The adaptation is implemented as a lookup table based on the use of experimentally derived maps.

As mentioned earlier, Pfiffner *et al*. (2003) and Liu, J. *et al*. (2007) use optimal control on a CVT. Pfiffner *et al*. (2003) use the mathematical model obtained from mechanical analysis, whereas Lui *et al*. (2007) use a predictive model strategy to obtain the model, and a genetic algorithm (GA) to solve the control problem. One of the motivations of using a genetic algorithm is because the solution search starts from many points simultaneously, and also because a GA is less susceptible to getting stuck in a local minimum. The results show an improvement over a standard PID controller.

A parallel hybrid vehicle with a CVT is considered by Won *et al*. (2005), mainly for energy management purposes. The energy management task considers two issues: Torque distribution, and Charge maintenance. It is mentioned that the implementation of optimal control is promising, however difficult to generate an appropriate solution “given the unknown nature of the driving situation”. A torque representation of the system is given, and then approximation and linearization applied. In order to maintain the current, the state of charge is kept within a range. An important highlighted issue is the difficulty of doing on-line optimization, for this reason the CVT gear ratio is set to follow the optimal path along the efficiency curves for the engine. The resultant ratio is used to determine the torque distribution.

Optimal Control

There is a great deal of interest in finding optimal control solutions to practical engineering problems. But many approaches adopted do not necessarily involve using conventional optimal control strategies. However, some of the most relevant literature associated with the particular powertrain problem of interest is mentioned below, which mainly includes optimal control in automotive systems, or problems where conventional optimal control strategies have in fact been applied.

An optimal control approach for a turbocharged SI engine using an automated wet clutch is proposed by Frei *et al*. (2006). First, an analysis of the possible variables to be controlled is undertaken, where the spark and clutch commands are suitably selected. Because of the nonlinear characteristics of the model, the use of the Minimum Principle and other analytical methods are discarded, and instead a nonlinear programming approach is selected. It is stated that the results obtained might not be the optimal solution but that improvements can be achieved. Interestingly the time to reach the torque specifications was actually reduced considerably in most cases.

Haj-Fraj *et al*. (2001) present an optimal control of gear shift operation using Dynamic Programming. The automatic transmission is modelled and the nonlinear characteristic (from the gearshifts) is simplified by linearization methods. The system is discretized and the optimal problem formulation is stated. Simulation results show smoothness in the acceleration, and the system is found to be robust with respect to varying the gear shifting time.

A Nonlinear Programming optimization for a power split in a hybrid vehicle is presented by Pérez *et al*. (2007). The hybrid system consists of an internal combustion engine and a bank of ultra-capacitors as the storage system. As mentioned earlier, conventional optimal control strategies are discarded owing to the non-linearity of the system. The objective function is to minimize the power losses and the discrete problem is solved using MINOS (a Fortran-based computer system for nonlinear applications). The results are compared with a Dynamic Programming solution proposed by Pérez *et al*. (2006). The Nonlinear Programming solution shows an improvement over Dynamic Programming.

A recent publication by Song *et al*. (2011) shows a customized Dynamic Programming solution to a clutch fill control problem. The authors state that in conventional Dynamic Programming some of the discretised states are not reachable at all. For this reason, they propose a customized version where an accurate solution with a reduced computational burden is achieved. This version inverts the discretized dynamic model making the control variable a

function of the system states, with this eliminating the inaccuracy introduced by approximations in the conventional version which uses linear interpolation.

KERS control

As mentioned earlier, there is not much literature related to flywheel-based KERS control. However, Ghedamsi *et al*. (2007) present a multiple-input-multiple-output (MIMO) PID-based Torque control for a wind generator. The flywheel is connected to an induction machine which converts the excess electrical energy into mechanical energy. The flywheel works as a regulator to improve electrical power quality. The mathematical representation of the Induction Machine is presented and a simulation is undertaken to validate the proposed control strategy in terms of steady-state and transient responses.

Cikanek *et al*. (2002) discuss the regenerative braking process for an HEV. A motor-generator is used to transfer the energy back and forth (from mechanical to electric energy), and a proportioning ratio control strategy is used to optimize the regenerative process. The algorithm is developed using MATRIXx (engineering software developed by National Instruments which provides solution for dynamic control systems). The code was tested in a development vehicle.

An Optimal Torque control using the variational method is presented by Inoue *et al*. (2010). In this publication, the modelling and control for an induction motor is presented. The variational method is used to find the optimal torque, and appropriate electronic circuits are included in order to maintain the optimal torque found. The strategy is verified via simulation using MATLAB/Simulink and the SimPowerSystems block set.

Ye *et al*. (2010) present an electronic configuration connecting the main and auxiliary sources of an electric vehicle in order to increase system efficiency by controlling the regenerative current with an H_2/H_∞ control strategy. The software program is designed in Matlab (and rewritten in C for on-board memory implementation). The results show efficiency improvements of 3% in soft braking, and 6% in emergency braking. Another robust control solution for an

electric vehicle is shown by Cao *et al* (2005), the objective being to improve the regenerative process without damaging the energy storage system. The natural robustness of standard PID is mentioned but the operational requirements require a more effective method. The results show an improvement in the charging currents over use of a PI control.

Cheng *et al*. (2011) propose a genetic algorithm neural network control for a recovery system of an electric motorcycle. It is mentioned that the advantage of using neural networks is in handling non-linear or highly correlated models. The motivation to use a genetic algorithm is the searching efficiency. Neural networks are capable of learning governing behaviour, and in finding a relationship between input/output parameters. The designed algorithm does not require complicated computations, which makes it easy to implement. The results show more stability in the energy recovery of the battery which increases its life-cycles. Cao *et al*. (2007) also use neural network control, in this case for an electric vehicle. The implementation of two networks, one that determines the PID parameters, and one that estimates the nonlinear prediction model, form a full closed-loop self-adaptive PID control strategy. The importance of off-line training of the neural networks is stated in order to reduce on-line computing time. Results for driving and regenerative braking experiments are shown and compared with standard PID control, with considerable improvement in the systems time response.

Having reviewed the literature, and with specific interest in the optimal control of a flywheel-based KERS with a CVT, it is evident from the literature that there is an important need for development of tools that can be used to implement conventional optimal control theory. The application here being focused on energy efficiencies for volume-produced vehicles but still with interest in the possibility of using optimal control for performance enhancement in motor sport. Until these computational tools are available, the full benefit of optimal control theory will not be realised. Apart from the issues of real-time implementation in practice, the key requirement is adequate treatment of a KERS taking account of bearing friction and adoption of even the simplest model of a CVT. Therefore

with a range of unsolved problems to be addressed, a position has been reached where justifiable objectives can be set.

1.5 Objectives of the thesis

The aim of this thesis is to develop a suitable optimal control strategy for a flywheel-based KERS with a CVT. The intention is to propose a technique that can be used to optimize the energy stored in the system and with appropriate development can be implemented in both motorsport and mainstream production vehicles.

The specific objectives are as follows:

- To find a suitable and simple, yet representative, model of the flywheel-based KERS with a CVT for control purposes.

- To implement the conventional optimal control strategies, and to find, develop and determine an appropriate optimal control methodology for this application.

- To compare the developed optimal control strategy with a non-optimal strategy in terms of performance and robustness.

Two main contributions have been made in this thesis, namely: i) it is the first time that the use of Runge Kutta, and Modified Euler integration methods has been recorded as part of inverting the integration process in discrete Dynamic Programming. In addition, the assumption and use of piece-wise linear control as part of this procedure has also not been reported before; and ii) this is the first implementation of optimal control via discrete Dynamic Programming, and Pontryagin's Maximum Principle applied to a flywheel-based Kinetic Energy Recovery System.

1.6 Layout of the thesis

The layout of the thesis can be given as follows:

Chapter 2 presents the modelling of a flywheel-based KERS with a CVT. Initially a simplified frictionless model is presented, and later friction is included in the model with two different variants. A previously-published extended model is included which includes a hydraulic actuator for a pulley-based CVT. The model is included to contrast a simplified model with a more realistic model.

Chapter 3 covers Optimal Control theory including the implementation of both Pontryagin's Maximum Principle, and Dynamic Programming. Application of Pontryagin's Maximum Principle to the optimal control of a linear oscillator is used to understand the computational problem associated with a corresponding Dynamic Programming solution and the likely problems for flywheel-based KERS with a CVT implementation. It shows the advantages and disadvantages of these methodologies, where major difficulty with Dynamic Programming is encountered. This difficulty then leads to the development of a new improved modified computational version of conventional Dynamic Programming. The purpose of this, is the development of an accurate and efficient optimal control strategy which can be implemented on a flywheel-based KERS with a CVT. The improved version of discrete Dynamic Programming is based on control parameter finding, which is done by inverting the integration process. To achieve this, both the Euler numerical integration method, and Runge Kutta numerical integration method are explored and considered, where Runge Kutta is selected as the appropriate method to be used.

Chapter 4 shows the implementation of the Pontryagin's Maximum Principle, and the newly developed improved version (from Chapter 3) of a modified Dynamic Programming approach. This applies to on the flywheel-based KERS models with a CVT, presented in Chapter 2 (for both the frictionless and simple friction options). Further development is undertaken to take account of problems not previously seen with the linear oscillator. A suitable new improved, methodology in the form of a modified Dynamic Programming, this time based on the modified Euler numerical integration method, is fully implemented for the

flywheel-based KERS models with a CVT. This implementation of optimal control strategies explores two possibilities: i) control variable optimization, and ii) energy optimization. An example which emphasises the potential of using optimal control theory for energy optimization in flywheel-based KERS with a CVT is presented. For this example, the simplified model is combined with an additional control input which can be related to friction brakes.

Chapter 5 presents a comparison between the implemented optimal control strategy and conventional classical control. Both, the simplified and more realistic models are compared with proportional (P) and proportional-integral (PI) controllers. The implementation of the classical control strategy is achieved using Matlab-Simulink, where the model is verified and later, control is applied.

Chapter 6 draws appropriate conclusions of the study and identifies further research.

2. MODELS OF A FLYWHEEL-BASED KERS WITH A CVT

A very important step in the use of control theory is dynamic system modelling. Therefore, an understanding of the dynamics of the system to be controlled is usually essential. For this reason some sort of modelling has to be done as a first step. This chapter presents (for control purposes) the kinematics and dynamic modelling of a flywheel-based KERS with a CVT. First a simple model of a KERS is shown, and later a more complicated version is given. The appropriate adaptations and representations of the system for control purposes are then explained. The more realistic model is extended to include a hydraulic piston, which works as actuator for the pulleys (assuming a pulley-based CVT transmission is used). First considering the dynamic model for a simple KERS.

2.1 Obtaining a simple KERS model

A KERS can be represented in its simplest form by three elements: i) The device containing the kinetic energy (in this case a vehicle's wheel), ii) the transmission device (in this case a pulley-based CVT), and iii) a storage device (in this case a flywheel). A free body diagram representation of this model is shown in figure 2.1.

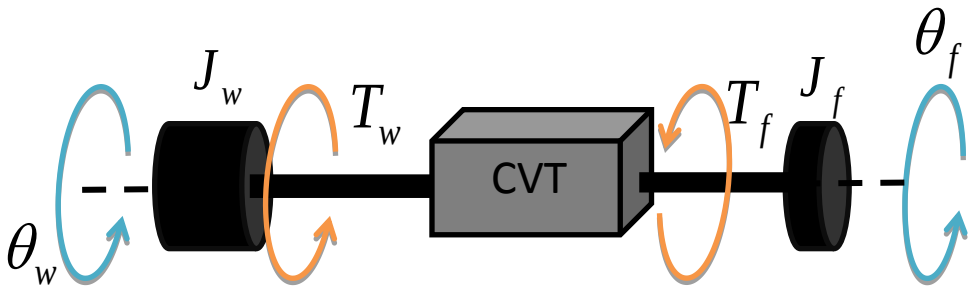

Figure 2.1 Representation of a simple KERS for an automotive application

In order to construct a model that can be operated for control applications a kinematic analysis and dynamic representation of these elements has to be performed where the gear ratio is allowed to vary as a function of time. The wheel and flywheel are represented as two rotating discs where the equations of motion are given as follows:

$$T_f(t) = J_f \ddot{\theta}_f(t) \tag{2.1}$$

and

$$T_w(t) = J_w \ddot{\theta}_w(t) \tag{2.2}$$

where $T(t)$ is the total torque, J is the moment of inertia, and $\theta(t)$ is the angular displacement for the flywheel and drive wheel respectively.

A CVT consists of two pulleys, with a V-shape that can be adjusted, connected by a metal belt. The V-shape of the pulleys allows adjustment of the gear ratio at each end. The forces and moments associated with the pulley are given as follows:

$$T_f(t) = Fr_{tf}(t) \tag{2.3}$$

and

$$T_w(t) = -Fr_{tw}(t) \tag{2.4}$$

where $r_t(t)$ represents the distance from the connecting shaft to the belt and F is the force inside the belt. Now, defining the gear ratio $G(t)$ as the ratio of the input radius to the output it radius i.e.:

$$G(t) = \frac{r_{tw}(t)}{r_{tf}(t)} \tag{2.5}$$

and, by combining equations (2.3) to (2.5), the relationship between the input and output torques becomes:

$$-T_f(t)G(t) = T_w(t) \tag{2.6}$$

In order to find the overall relationship between the input and output of the system for control purposes, the dynamics model for the CVT has to be included. The CVT speed relationship for the transmission can be defined as:

$$\dot{\theta}_f(t) = G(t)\dot{\theta}_w(t) \tag{2.7}$$

and, as mentioned by Powell *et al*. (2000), the main difference between a fixed-ratio transmission, and a variable transmission, is found while finding the derivative of the speed. This is given as:

$$\frac{d}{dt}\left(\dot{\theta}_f(t) = G(t)\dot{\theta}_w(t)\right) \tag{2.8}$$

and

$$\ddot{\theta}_f(t) = \dot{G}(t)\dot{\theta}_w(t) + G(t)\ddot{\theta}_w(t) \tag{2.9}$$

Now working with equations (2.1) to (2.9), the overall dynamic model becomes:

$$-J_f G(t)\dot{G}(t)\dot{\theta}_w(t) = (J_f G^2(t) + J_w)\ddot{\theta}_w(t) \tag{2.10}$$

This is a time-varying 2nd order differential equation and its representation in state space form is:

$$x_1(t) = \theta_w(t) \tag{2.11}$$

and

$$\dot{x}_1(t) = \dot{\theta}_w(t) = x_2(t) \tag{2.12}$$

and

$$\dot{x}_2(t) = \frac{-J_f G(t)\dot{G}(t)x_2(t)}{J_w + J_f G^2(t)} \tag{2.13}$$

where equations (2.12) and (2.13) are the state differential equations.

It is important at this point to define a control variable. As seen in equation (2.13), there are two elements present that are neither constant nor states. Therefore appropriate selection of a control variable is needed (and appropriate modifications must be made). The gear ratio cannot be chosen as a control variable because its derivative is also found in equation (2.13) and, as clearly emphasised by Brogan (1991), the system input must be the actual physical input, but when the input is specified, there is no freedom left in specifying its

derivative. Pfiffner *et al*. (2003) and Setlur *et al*. (2003) select the derivative of the gear ratio as the system input. In doing this, another state has to be added so that the system has three state variables and is represented by three first order differential equations. This modification gives the following equations:

$$x_3(t) = G^2(t) \tag{2.14}$$

and, defining the control variable as:

$$\dot{G}(t) = u(t) \tag{2.15}$$

and

$$\dot{x}_2(t) = \frac{-J_f u(t) x_2(t) x_3^{0.5}(t)}{J_w + J_f x_3(t)} \tag{2.16}$$

and

$$\dot{x}_3(t) = 2u(t) x_3^{0.5}(t) \tag{2.17}$$

It can now be seen that the simplest state space representation of a controllable KERS presented here is in the form given by equations (2.12), (2.16) and (2.17). This model is used in the thesis to explore simple control possibilities. But it can also be seen that the simplest state space representation of a controllable system is a set of non-linear first order differential equations (even though the individual components have linear characteristics).

2.2 Obtaining a more realistic KERS model

Owing to its simplification the system presented in the previous section does not include representative characteristics of the system, like friction; which can have a large impact on the system behaviour. These features can be crucial for a control analysis especially when dealing with energy optimization.

Kiencke *et al*. (2005) show more realistic models for diverse vehicle subsystems. The model in the driveline section of this publication has been used as a guide in order to make the KERS model more realistic. A free body diagram, to represent a more realistic KERS model for an automotive application, is shown in figure 2.2.

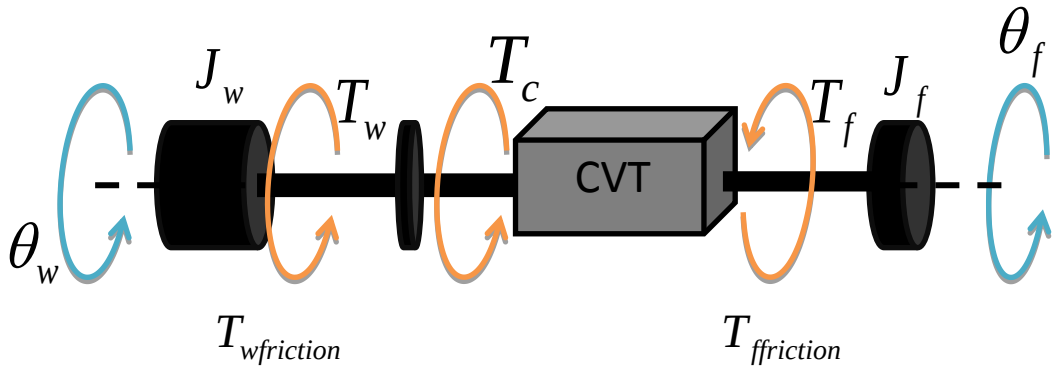

Figure 2.2 Representation of a more realistic KERS for an automotive application

From figure 2.2, the dynamic analysis has to be performed to obtain the equations of motion and with this, construction of the mathematical representation of the whole system has been achieved. For the vehicle and the wheels, the subsystem can be represented as follows:

$$T_w(t) - F_v(t) r_w - T_{wf}(t) = J_w \ddot{\theta}_w(t) \tag{2.18}$$

where $F_v(t)$ represents the traction forces acting on the vehicle, r_w is the radius of the wheels, and $T_{wf}(t)$ is the friction torque in the wheels, mainly present at the bearings. Kiencke *et al*. (2005) give a detailed formulation of the forces acting in the vehicle and these include: rolling resistance, wind friction, the acceleration of the mass, and the resolution of the vehicle weight on an inclined plane. However it is only the acceleration of the vehicle that is included in this more realistic model and therefore it can be written as:

$$F_v(t) = m_v a_v(t) \tag{2.19}$$

where m_v and a_v are vehicle mass and acceleration respectively. By assuming no tyre slip equation (2.19) can be expressed as:

$$F_v(t) = m_v \ddot{\theta}_w(t) r_w \tag{2.20}$$

The friction torque will actually be described in more detail later on together with the flywheel's friction torque.

The second block in figure 2.2 represents a friction clutch which when fully engaged, and assuming that it does not have any additional friction losses, the input and output torques remain the same, expressed as:

$$T_w(t) = T_c(t) \tag{2.21}$$

The CVT model like the one described in the previous section remains unchanged, therefore the ratio between the input and output stays the same; and for the flywheel block the introduction of friction is needed so it is now represented by:

$$T_f(t) - T_{ff}(t) = J_f \ddot{\theta}_f(t) \tag{2.22}$$

where $T_{ff}(t)$ is the rotational friction for the flywheel.

The main rotational friction for both the wheel and the flywheel occurs at the bearings. Special interest is focused on the friction in the flywheel since the rotational speeds are very high and of course because it is energy store. As mentioned in Chapter 1, there are two types of bearings used in flywheel systems: Ball bearings, and Magnetic bearings (see Bolund *et al*. 2007, Liu, H. *et al*. 2007).

Considering ball bearings, a linear relationship between the coefficient of friction and the rotational speed is assumed, due to the fluid viscosity properties of the lubricants. In this case, a simplified rotational friction torque characteristic for the flywheel is given as:

$$T_{ff}(t) = K_v \dot{\theta}_f(t) \tag{2.23}$$

where K_v is a viscosity related constant factor. For the friction losses in the wheels, the same simplification can be used, thus the relationship is given as:

$$T_{wf}(t) = K_v \dot{\theta}_w(t) \tag{2.24}$$

The friction model for magnetic bearings by contrast has a nonlinear relationship with respect to the rotational speed (Schweitzer 2002). This can be simplified to a quadratic relationship, which is given as:

$$T_{ff}(t) = K_{MB} \dot{\theta}_f^{\,2}(t) \tag{2.25}$$

where K_{MB} is a constant factor.

To construct the full model of the system, equations (2.18), (2.21), (2.22) and (2.5) are combined to give the final relationship:

$$-J_f G(t)\dot{G}(t)\dot{\theta}_w(t) - T_{ff}(t)G(t) - T_{wf}(t) - F_v(t)r_w = (J_f G^2(t) + J_w)\ddot{\theta}_w(t) \tag{2.26}$$

But before substituting all the variables, equations (2.23) and (2.25) are combined with equation (2.7) in order to formulate the final relationship as function of the angular displacement of the wheel. These expressions are given as:

$$T_{ff}(t) = K_v G(t)\dot{\theta}_w(t) \tag{2.27}$$

for ball bearings, and for magnetic bearings:

$$T_{ff}(t) = K_{MB} G^2(t)\dot{\theta}_w^{\,2}(t) \tag{2.28}$$

Now equations (2.20), (2.27) or (2.28), and equation (2.24) can be used in equation (2.26), to give two relationships:

$$-J_f G(t)\dot{G}(t)\dot{\theta}_w(t) - K_v\dot{\theta}_w(t)G^2(t) - K_v\dot{\theta}_w(t) - m_v\ddot{\theta}_w(t)r_w{}^2 = (J_f G^2(t) + J_w)\ddot{\theta}_w(t) \quad (2.29)$$

for ball bearings, and

$$-J_f G(t)\dot{G}(t)\dot{\theta}_w(t) - K_v\dot{\theta}_w{}^2(t)G^3(t) - K_v\dot{\theta}_w(t) - m_v\ddot{\theta}_w(t)r_w{}^2 = (J_f G^2(t) + J_w)\ddot{\theta}_w(t) \quad (2.30)$$

for magnetic bearings.

From equation (2.29), it can be seen that only the second state has been modified with respect to the simplified model. Therefore including friction from ball bearings the second state equation changes to:

$$\dot{x}_2(t) = \frac{-J_f u(t)x_2(t)x_3{}^{0.5}(t) - K_v x_2(t)x_3(t) - K_v x_2(t)}{J_w + J_f x_3(t) + m_v r_w{}^2} \quad (2.31)$$

and considering magnetic bearings the second state equation becomes:

$$\dot{x}_2(t) = \frac{-J_f u(t)x_2(t)x_3{}^{0.5}(t) - K_v x_2{}^2(t)x_3{}^{1.5}(t) - K_v x_2(t)}{J_w + J_f x_3(t) + m_v r_w{}^2} \quad (2.32)$$

A more realistic model of the Flywheel-based KERS is now found in equations (2.11), (2.12), (2.14), (2.15), (2.31) or (2.32), and (2.17). The more realistic model represents a greater challenge in terms of control theory, especially when considering the magnetic bearings. Cross *et al*. (2008) and Van Mierlo *et al*. (2004), represent the rotational friction forces with a constant term in order to simplify the problem.

2.3 Obtaining a more realistic KERS model with a hydraulic piston

In the literature review, it was mentioned that Müller *et al*. (2001) included the representation of a hydraulic system in their model. This is mainly done because the position of the hydraulic system and the gear ratio have a non-linear relationship. In this section, an extension of the previous dynamic models

is made to include a hydraulic system. This model is shown just to give an indication of an even more realistic KERS.

The model is constructed assuming a non-linear (quadratic) relationship between the gear ratio $G(t)$ and the hydraulic piston position $x(t)$, in the form:

$$G(t) = K_a x^2(t) \tag{2.33}$$

where K_a is the constant of proportionality.

Finding the derivative with respect to time from equation (2.33), gives a relationship for the gear ratio rate of change as:

$$\dot{G}(t) = 2K_a \dot{x}(t)x(t) \tag{2.34}$$

Considering that a hydraulic piston can be represented by a second order oscillator, the dynamic equation is given as:

$$\ddot{x}(t) + \frac{c}{m}\dot{x}(t) + \frac{k}{m}x(t) = \frac{F(t)}{m} \tag{2.35}$$

Setting the external force as the control input, the state space is now given as:

$$x_1(t) = x(t) \tag{2.36}$$

and

$$f_1(t) = \dot{x}_1(t) = \dot{x}(t) = x_2(t) \tag{2.37}$$

and

$$f_2(t) = \dot{x}_2(t) = \frac{u(t) - cx_2(t) - kx_1(t)}{m} \tag{2.38}$$

Considering the simple CVT model, a third state can be formulated as:

$$\dot{x}_3(t) = \frac{-J_f G(t)\dot{G}(t)x_3(t)}{J_w + J_f G^2(t)} \tag{2.39}$$

Using equations (2.33) and (2.34) in equation (2.39), the final relationship for the extended state is given as:

$$\dot{x}_3(t) = \frac{-2J_f {K_a}^2 {x_1}^3(t)x_2(t)x_3(t)}{J_w + J_f {K_a}^2 {x_1}^4(t)} \tag{2.40}$$

The state space model represented in equations (2.37), (2.38) and (2.40) is a frictionless representation of the CVT considering the hydraulic piston. The input force is the actual force applied to the actuator, and $x_3(t)$ represents the speed of the wheels. If friction is added to the model, $\dot{x}_3(t)$ changes to:

$$\dot{x}_3(t) = \frac{-2J_f {K_a}^2 x_2(t){x_1}^3(t) - K_v {K_a}^2 {x_1}^4(t)x_3(t) - K_v x_3(t)}{J_w + J_f {K_a}^2 {x_1}^4(t) + m_v {r_w}^2} \tag{2.41}$$

The upgraded state space model is a more realistic representation. However, this representation assumes that only one of the pulleys can be manipulated to vary the gear ratio.

It can be seen that including the actuator substantially complicates the nature of the CVT transient dynamics giving a highly nonlinear system. This model is not actually used for control purposes in this thesis but is included to show the type of model that is ultimately needed as KERS system models become more realistic.

2.4 Conclusions of the findings in Chapter 2

In this chapter the state space model representation of a flywheel-based KERS with a CVT system is obtained through kinematic and dynamic analysis of pulley-based CVT transient behaviour. Initially a simplified frictionless model is developed, and later on, friction is included in the model, considering two

different bearing types operating with the flywheel. CVT dynamics are governed by a nonlinear relationship which represents a degree of complexity for control purposes. For control purposes where a clutch is used this is assumed to be fully engaged at all times. An extended model including a hydraulic actuator acting on a pulley-based CVT is described, to demonstrate its highly nonlinear characteristic only to suggest an extremely complicated control design were it to be used. In this chapter, a pulley-based CVT is used to obtain the models, however the transient behaviour presented could be considered for other friction CVTs (i.e. toroidal) as it is evident that the same degree of complexity would be present. The simplified CVT with friction models will be used in the following chapters for control purposes.

3. OPTIMAL CONTROL METHODOLOGIES – THE POTENTIAL FOR KERS CONTROL

Optimal Control theory differs in various ways from Classical control theory, therefore, it is important to know the differences between them. It is therefore helpful to understand the underlying concepts behind optimal control theory, in order to implement it (even for people with a reasonable understanding of classical control). With this intention this chapter first presents a general description of optimal control theory, particularly Pontryagin's Maximum Principle, and the Dynamic Programming strategy. It then moves to the application of these strategies to a linear oscillator problem, as an academic example. This is done with the intention of building up a methodology that can be tested and then applied to more complex systems (in later Chapters). The KERS is more complex because it has nonlinear characteristics (as shown in Chapter 2). The first methodology applied is Pontryagin's Maximum Principle; and the second is Dynamic Programming.

After initial implementation of Dynamic Programming, results show some evident deficiencies, therefore a new strategy for Dynamic Programming is explored and developed. This is then applied to a combination of forward integration approaches needed for numerical implementation of discrete time of Dynamic Programming, culminating to form a modified version of Dynamic Programming. The results show the need for this modified version of Dynamic Programming which has particular advantages over the use of direct and standard applications of Dynamic Programming. This strategy is later applied to a KERS model in Chapter 4. Before drawing conclusions at the end of the Chapter, a summarizing table with the different versions of Dynamic Programming is presented.

3.1 Optimal Control Theory

Classical control theory is mainly a trial and error design methodology (Kirk 1998). Although it is very useful and often reaches acceptable behaviour, the control and specification of modern systems, demands more accurate control strategies, like optimal control. Also, whereas in classical control theory, time

and frequency domain characteristics determine the quantitative performance of the controlled system, in optimal control theory, the objective is to minimize or maximize a certain performance criterion. This criterion can be defined in terms of different indicators, such as time or energy.

Ogata (2002) emphasizes the use of the frequency-response and the root-locus methods as the fundaments of classical control theory. These methods are most useful when the system is represented in the frequency domain, which is achieved by time domain transformation of the mathematical expressions using Laplace Transform techniques (i.e. differential equations describing the system plant).

Much of linear control theory has been based on the frequency domain representation. It is not only suitable for determining the frequency response (which is best represented by Bode diagrams) but also for determining control stability, and finding the steady state error. These characteristics are key factors for measuring the performance of classical control systems.

Classical control theory measures the system response to a given input (commonly a step input). For a closed loop control, the response is fed back to be compared with the desired input (reference). The difference is used to generate the new system input (effectively a trial and error approach). The design of a controller is based on this error detection process, where the error is adjusted by a gain and compensating factor. The compensating factors are based on the system frequency response (leading or lagging characteristics), derivative compensators respond based on an anticipation of the systems response (error prediction) and integral compensators based on the previous systems response (error accumulation). The most common controller designs use all the three characteristics and are known as proportional-integral-derivative (PID) controllers, however the proportional controller can stand alone or use any of the compensators.

In order to emphasize the main characteristics of classical control theory, relative advantages and disadvantages are presented in table 3.1.

Advantages	Disadvantages
• Satisfies more or less arbitrary performance requirements • Robust (insensitive to disturbances) • No error (steady state) • Stability consideration	• Acceptable but not optimal • Difficult for MIMO (Multiple Input Multiple Output) systems • Primarily used with linear constant coefficient systems

Table 3.1 Advantages and Disadvantages of classical control theory

The representation of a system for Optimal Control by contrast is given in state space form (a time domain representation), which, to use the words of Pinch (1993), “The state variables are the ones that define what the system is doing at specific time t and the control variables are the ones that modify the behaviour of the system”. The state space representation is mainly used in modern control theory; however classical control theory problems can be approached using the time domain representation but much of the frequency domain understanding is compromised.

Optimal Control theory is not based on frequency response characteristics of the system, and the controller design is not built specifically to eliminate the systems error in the steady state response given desired time response characteristics. Optimal control design is based on meeting specific system requirements and on minimizing (or maximizing) specific performance criteria. For this reason, it is more convenient to work in the time domain.

An essential part of the process of constructing an optimal solution is the construction of the dynamic model which includes the definition of so called state variable which must satisfy quite strict conditions.

The state variables are commonly represented in vector form by variable $\bar{x}$, and the control variable is normally represented by vector $\bar{u}$. The standard notation of a system model in state space is as follows:

$$\dot{\bar{x}}(t) = f\left(\bar{x}(t), \bar{u}(t), t\right) \tag{3.1}$$

In order to construct an optimal control solution, both Brogan (1991) and Kirk (1998), emphasise three components:

i. A mathematical description of the system (model) to be controlled
ii. A description of the system constraints including the task to be performed
iii. The specification of a performance criterion

The mathematical model of the system involves the representation of the system in state space form.

The system constraints are normally physical restrictions or limitations imposed on the system; these constraints establish the boundaries for the control problem and they play a very important role in order to find an appropriate solution.

The performance measurement, commonly represented by variable J and also called performance cost, is the quantitative evaluation of the system performance based on a specific criterion which is chosen by the designer. The cost can be evaluated at any stage and its representation is usually given as follows:

$$J(t) = \int_{t_o}^{t_f} f_0(\bar{x}(t), \bar{u}(t), t)dt + h(\bar{x}(t_f), t_f) \tag{3.2}$$

Where f_0 represents the cost associated to the states, control variables or time behaviour and h represents the cost associated with the final conditions of the system in terms of the states or time.

The two most common strategies to approach an optimal control problem are the Pontryagin's Maximum Principle and Dynamic Programming. The Pontryagin's Maximum Principle is essentially a mathematical strategy, whereas Dynamic Programming is a computational strategy. Both are implemented in

order to verify the accuracy of the results, to establish the most appropriate method for KERS development.

3.2 Optimal Control Using Pontryagin's Maximum Principle

Pontryagin's Maximum Principle, is very well described in Pinch (1993), and is a method based on the calculus of variations. This approach can be described as the optimization process of functionals (i.e. "A transformation from a function to a number", Bellman et al (1965)) in which the functions themselves satisfy differential equations. The calculus of variations cannot be implemented directly in optimal control problems since control systems are described by differential equations (instead of integrals), and also owing to the form the constraints can take. It was for this reason that Pontryagin developed the Maximum Principle suitable for work with differential equations.

When a system, like the state space model given by equation (3.1) is subject to the performance criterion given by equation (3.2), it is said to be an optimal control problem. But before moving to the process of solving this problem, there is an important question that needs to be asked (Brogan (1991), Kirk (1998) and Pinch (1993)): Is the system controllable? In most cases it is assumed that there is one control that will successfully drive the system and meet the constraint criteria. Therefore instead of just determining the existence of the optimal control, the challenge in practice is to find it. While using Pontryagin's Maximum Principle to find such control, the solution found will certainly satisfy certain necessary conditions. This means that the system will meet all the specific requirements in following an optimal trajectory; however this may not be the global optimal solution to the problem.

Two concepts, that are needed to solve an optimal control problem using the Pontryagin's Maximum Principle, have to be considered first. In order to minimize the cost function in equation (3.2) using variational and optimization theory, it is necessary to introduce the concept of Lagrange multipliers. By doing this, a new function (which includes both the states and the integral part of the cost function) is formed. This function is referred as the Hamiltonian H.

In order to define the Hamiltonian it is necessary to revise briefly the concept behind the calculus of variations. From optimization theory it is known that the minimum (or maximum) points are reached when the partial derivatives of the function of interest are equal to zero. In the case of the calculus of variations the optimization of J is found when the variations around the optimal trajectory are equal zero, the mathematical expression for this is given by:

$$\delta J = \int_{t_0}^{t_1} \left\{ \sum_{i=1}^{n} \frac{\partial f_0}{\partial x_i} \partial x_i + \sum_{j=i}^{m} \frac{\partial f_0}{\partial u_m} \partial u_m \right\} dt + f_0(t_1) \partial t \tag{3.3}$$

where n represents the number of states and m represents the number of control variables. However the control variables $\bar{u}$ and the states $\bar{x}$ are dependent on the state equations, and for this reason it is necessary to introduce the Lagrange multipliers $\psi_i(t)$. Therefore, in order to complete the minimization process it is necessary to introduce the following set of integrals:

$$\Phi_i = \int_{t_0}^{t_1} \psi_i(t) \left(\dot{x}_i - f_i(\bar{x}, \bar{u}) \right) dt \tag{3.4}$$

which, as it was done for J, their variations have to be equal zero; therefore a new condition can be expressed as:

$$\delta J + \sum_{i=1}^{n} \delta \Phi_i = 0 \tag{3.5}$$

giving as a result the Hamiltonian H, expressed as:

$$H(t) = -f_0(t) + \sum_{i=1}^{n} \psi_i(t) f_i(t) \tag{3.6}$$

where f_0 is the function evaluated in the performance criterion, n represents the number of states, f_i are the state equations, and ψ_i are the Lagrange multipliers which satisfy the necessary conditions:

$$\dot{\psi}_i(t) = -\frac{\partial H(t)}{\partial x_i(t)} \tag{3.7}$$

The set of equations formed by (3.7) are commonly known as the co-state functions. The condition for optimality is then given by:

$$\frac{\partial H(t)}{\partial \bar{u}(t)} = 0 \tag{3.8}$$

Up to this point, the optimal control problem does not deal with any constraints on the system, neither physical nor imposed, but just with the optimization procedure subjected to a specific criterion. If a problem of this nature should be solved, it is just a matter of mathematically solving these set of equations for the unknown variable $\bar{u}$. For simple linear systems this represents solving a set of integrals but the variational approach normally leads to a nonlinear two-point boundary-value problem, which becomes extremely difficult to be solved analytically, and as Kirk (1998) mentions, it is more appropriate to use numerical methods to find the solution to this type of problems.

The inclusion of the constraints is analysed in detail by Kirk (1998) (Chapter 5), which normally does not increase the difficulty of the problem. It is important to mention however that these constraints have to be chosen carefully in a very well represented mathematical way. Kirk (1998) presents the eight most common boundary conditions for control systems. This will be shown shortly and explained in more detail when Pontryagin's Maximum Principle is applied to some relevant test cases.

Before moving forward to implement Pontryagin's Maximum Principle, it is appropriate to give a brief description of the alternative optimal strategy: Dynamic Programming. With this, the discussion and analysis of the results will be easier to understand.

3.3 Optimal Control using Dynamic Programming

Dynamic Programming is a multistage sequential decision-making process based on Bellman's principle of optimality, described in his own words: "An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision" (Bellman 1962).

Brogan (1991) describes one of the main advantages of the principle of optimality by saying that it replaces a decision between all alternatives by a sequence of decisions between fewer alternatives. The principle of optimality fulfils both necessary and sufficient conditions, which means that the solution found is a global minimum within the specification of the problem.

For example, consider a decision process to select the optimal path between two points a and z, where two different trajectories can be chosen: i) One through b and c, and ii) another one through b and d. If a cost is incurred for each trajectory, the optimal cost can be expressed as:

$$J^* = \min\left[J_{a-b-c-z} \quad J_{a-b-d-z}\right] \tag{3.9}$$

where the cost for each trajectory is the sum of the costs from their segments i.e.:

$$J_{a-b-c-z} = J_{a-b} + J_{b-c} + J_{c-z} \tag{3.10}$$

for the first trajectory, and

$$J_{a-b-c-z} = J_{a-b} + J_{b-d} + J_{d-z} \tag{3.11}$$

for the second trajectory.

Assuming that the optimal trajectory from b to z is given by the path $b-c-z$, it can be said that:

$$J_{b-c} + J_{c-z} < J_{b-d} + J_{d-z} \tag{3.12}$$

For this reason, the optimal path from a to z is found as:

$$J^* = \min\left[J_{a-b}\right] + J_{b-c-z} \tag{3.13}$$

Making clear that when finding the optimal path between a and b, the path $b-d-z$ should not be considered any longer as an option for optimality; the global minimum found it is used for further decision taking. From this example, it can be seen that the Principle of Optimality implies an initial decision based on comparing the last trajectories (‘backward’ run).

Dynamic Programming can be implemented for both continuous and discrete time systems. Continuous time Dynamic Programming deals with the solution of the Hamilton-Jacobi-Bellman equation (Kirk 1998), a nonlinear partial differential equation based on the principle of optimality that can be solved analytically or with numerical techniques. Kirk (1998) mentions that this equation works as a bridge between Dynamic Programming and variational methods (i.e. Pontryagin’s Maximum Principle). The discrete time Dynamic Programming method uses a combinatorial strategy and compares different possible trajectories, which are chosen and reduced by using the principle of optimality. Following a ‘backward’ run, the procedure selects the best trajectory by using an appropriate performance measurement. This strategy is relatively easy to implement numerically.

For a more detailed picture of how discrete Dynamic Programming works consider the following, if there is a two state system with state variable $\bar{x} = \left[\bar{x}_1 \quad \bar{x}_2\right]^T$, at any given time, there are an infinite number of values of $\bar{x}_1$ and $\bar{x}_2$ forming a two dimensional mesh. A discrete state mesh is found at each time step, for which different trajectories can be found. This trajectories are given by an initial state $\bar{x}(t) = \left[x_1(t) \quad x_2(t)\right]^T$, where t can only take discrete

values, and by applying an admissible control $\bar{u}$, giving a following state $\bar{x}(t+1) = [x_1(t+1) \quad x_2(t+1)]^T$. However in order to determine which route is optimal (and meets the specification) all combinations need to be compared. Although these combinations are considerably reduced given the principle of optimality, the problem easily becomes huge. Furthermore when more states are involved the mesh becomes multidimensional leading to a problem of growing complexity commonly known as the 'curse of dimensionality'.

For these reasons it is crucial to define the working limits to create the different combinations. A range for admissible values must be set for both state and control variables. The step size for these variables is very important and plays an important role not just in the accuracy of the solution but also in the efficiency of the strategy. But reducing the time-step, in order to make better paths available, easily leads to a computational problem. Setting the step size to be small enough so that the optimal control and trajectory could not be missed, would however be ideal.

Even with a good set of state and control variables when the system is taken from a specific point of the mesh at t_m by using a control u (chosen from a set of admissible controls), the chances that one of the admissible values of the mesh at t_{m+1} is reached, are not very good. For this problem the Dynamic Programming standard solution suggests interpolating the control u obtained so that the closest point can be achieved. For example, in the case of a linear system, if the admissible state at t_{m+1} is exactly half way between two non-admissible states that are reached by the admissible controls u_1 and u_2, then the control to reach the admissible state can be found by linear interpolation as follows: $u = \frac{u_1 + u_2}{2}$. Unfortunately for either high order systems or nonlinear systems, interpolation is generally not very successful in terms of accuracy.

Owing to the complexity of the system of interest, discrete Dynamic Programming is used in this application despite some of the difficulties with standard implementation. The reason for this is that the nonlinearity of the

system would make the Hamilton-Jacobi-Bellman equation hard to handle; therefore only discrete Dynamic Programming and the Pontryagin's Maximum Principle are considered further.

3.4 Application of Optimal Control to a Linear Oscillator

As mentioned earlier the application of optimal control differs from classical control. Therefore before attempting to implement an optimal control strategy for the system of interest (i.e. a KERS), it is useful to explore the most common methodologies on a simpler problem. Therefore a second order oscillator, formed by a mass attached to a spring and a damper, is used to build up an optimal control solution. It is chosen because of its relatively simple characteristics, i.e. a linear system formed of two states (two differential equations), and also because it is a well-known system. Thus it can be used to obtain an understanding of the numerical implementation of all the adopted methods.

Following the steps in section 3.1, to construct an optimal control solution, the first step is to construct the state space model of the system. The dynamic equation for a linear oscillator is given as:

$$\ddot{x}(t)+\frac{c}{m}\dot{x}(t)+\frac{k}{m}x(t)=\frac{F(t)}{m} \tag{3.14}$$

where m is the mass, k is the spring stiffness coefficient, c is the damping coefficient, F is an external force applied (or the desired control variable u), and $x(t)$ is the position of the mass with respect to its steady state point.

The state vector is defined as $\bar{x}=\begin{bmatrix}x_1 & x_2\end{bmatrix}^T$, where for the linear oscillator the state space form is represented as:

$$x_1(t)=x(t) \tag{3.15}$$

and

$$f_1(t)=\dot{x}_1(t)=\dot{x}(t)=x_2(t) \tag{3.16}$$

and

$$f_2(t) = \dot{x}_2(t) = \frac{u(t) - cx_2(t) - kx_1(t)}{m} \tag{3.17}$$

The system is in state space form, the following step is to determine the constraints on the system, and to specify the desired task to be performed. In this case the objective is to drive the system from a rest position with initial speed of zero (i.e. $x_1(0) = 0 \quad x_2(0) = 0$) to a final position and speed at a specified final time (i.e. $x_1(t_f) = x_{1f} \quad x_2(t_f) = x_{2f}$).

The last step is to define a performance measurement, in this case the optimization criterion involves the control variable. Its mathematical expression from equation (3.2) is shown below:

$$J(t) = \int_0^{t_f} \alpha u(t)^2 dt \tag{3.18}$$

This is a quadratic cost criterion which is referred to as a minimum 'energy' performance index when J in equation (3.18) is to be minimised.

Now that the general description of the optimal control problem is complete, it is appropriate to discuss the adoption of the strategies described in the previous sections. To keep the same order as they were presented the Pontryagin's Maximum Principle is applied first.

Application of Pontryagin's Maximum Principle

Turning to the use of the Maximum Principle to formulate the optimal control problem the procedure described in section 3.2 is followed. The Hamiltonian for a second order system is given as:

$$H(t) = -f_0(t) + \psi_1(t) f_1(t) + \psi_2(t) f_2(t) \tag{3.19}$$

By substituting equations (3.16), (3.17), and (3.18) into equation (3.19), H is obtained as:

$$H(t) = \alpha u^2(t)\psi_0(t) + x_2(t)\psi_1(t) + \frac{u(t) - cx_2(t) - kx_1(t)}{m}\psi_2(t) \tag{3.20}$$

where the "Lagrange multiplier" (ψ_0) related to the cost-function has been introduced, however from the definition of the Hamiltonian (equations (3.3) to (3.6)) it is known that its value is equal to -1, for this reason the co-state ($\dot{\psi}_0$) is always zero.

Using equations (3.7) and (3.8), from the Hamiltonian obtained in equation (3.20) the co-state equations can be found as follows:

$$\dot{\psi}_0(t) = 0 \tag{3.21}$$

and

$$\dot{\psi}_1(t) = \frac{k\psi_2(t)}{m} \tag{3.22}$$

and

$$\dot{\psi}_2(t) = -\psi_1(t) + \frac{c\psi_2(t)}{m} \tag{3.23}$$

And the condition for optimality is:

$$\frac{\partial H(t)}{\partial u(t)} = 2\alpha u(t)\psi_0(t) + \frac{\psi_2(t)}{m} = 0 \tag{3.24}$$

From equation (3.24) the control variable is given as:

$$u(t) = \frac{-\psi_2(t)}{2\alpha m\psi_0(t)} \tag{3.25}$$

The set of equations (3.20) to (3.25) formulate the minimum 'energy' optimal control problem for the linear oscillator using Pontryagin's Maximum Principle. In order to find the optimal control solution it is necessary to solve these equations. Simple linear problems can be solved in closed form; however for more complex problems an analytical solution is no longer available. The state and co-state equations have to be then solved numerically in the form of a two-point boundary-value problem.

Optimal Control solution for the Oscillator via Pontryagin's Maximum Principle

In order to find the optimal control solution analytically, the set of differential equations formed by the state and co-state variables, equations (3.16), (3.17), and equations (3.21) to (3.23), have to be solved. The condition for optimality given by equation (3.25) is used to relate the state and co-state variables, and the value of the constants of integration that emerge from the process can be found by using the boundary conditions.

Since the Hamiltonian is never a function of the variable x_0, equation (3.21) is always equals zero which implies that its solution is a constant value. Therefore any negative value, in order to meet the requirements of the Pontryagin's Maximum Principle theory, can be chosen (Pinch 1993). The standard solution for equation (3.21) is given as:

$$\psi_0(t) = -1 \tag{3.26}$$

To obtain the analytical solution for equations (3.22) and (3.23) it is necessary to do some algebraic manipulation. Therefore finding the derivative of equation (3.22) with respect to time the following is obtained:

$$\ddot{\psi}_1(t) = \frac{k\dot{\psi}_2(t)}{m} \tag{3.27}$$

And using equations (3.27) and (3.23), a final expression for ψ_1 is found as:

$$\ddot{\psi}_1(t) - \frac{c}{m}\dot{\psi}_1(t) + \frac{k}{m}\psi_1(t) = 0 \tag{3.28}$$

The second order differential equation given by (3.28) is actually a common expression whose solution is given by the method of undetermined coefficients, which is given as:

$$\psi_1(t) = C_1 e^{\theta t} + C_2 e^{\phi t} \tag{3.29}$$

where the coefficients for the exponential functions are determined by:

$$\theta = \frac{\frac{c}{m} + \sqrt{\frac{c^2}{m^2} - 4\frac{k}{m}}}{2} \tag{3.30}$$

and

$$\phi = \frac{\frac{c}{m} - \sqrt{\frac{c^2}{m^2} - 4\frac{k}{m}}}{2} \tag{3.31}$$

The constants C_1, C_2 can be determined by using the boundary conditions specified in the optimal control problem. With the solution given in equation (3.29), and using equation (3.22), a final expression for ψ_2 is given as:

$$\psi_2(t) = \frac{\left(C_1\theta e^{\theta t} + C_2\phi e^{\phi t}\right)m}{k} \tag{3.32}$$

Therefore in order to find the missing constant values (C_1, C_2), by substituting the solutions given in equations (3.32) and (3.26) into equation (3.25), the control variable can is expressed by:

$$u(t) = \frac{C_1\theta e^{\theta t} + C_2\phi e^{\phi t}}{2\alpha k} \tag{3.33}$$

Equation (3.33) can be substituted in equation (3.17) where it can be written in terms of the state variable x_1 as follows:

$$\ddot{x}_1(t)+\frac{c}{m}\dot{x}_1(t)+\frac{k}{m}x_1(t)=\frac{C_1\theta e^{\theta t}+C_2\phi e^{\phi t}}{2\alpha k m} \tag{3.34}$$

Equation (3.34) is a second order differential equation which has both a homogenous (complementary function) and a particular integral solution. The homogenous solution is found by using the method of undetermined coefficients and is given as:

$$x_{1H}(t)=C_3e^{\chi t}+C_4e^{\sigma t} \tag{3.35}$$

where, the coefficients of the exponential functions are:

$$\chi=\frac{-\frac{c}{m}+\sqrt{\frac{c^2}{m^2}-4\frac{k}{m}}}{2} \tag{3.36}$$

and

$$\sigma=\frac{-\frac{c}{m}-\sqrt{\frac{c^2}{m^2}-4\frac{k}{m}}}{2} \tag{3.37}$$

And its particular solution is given as:

$$x_{1P}(t)=\frac{C_1\theta e^{\theta t}}{2\alpha k\left(k+\theta c+\theta^2 m\right)}+\frac{C_2\phi e^{\phi t}}{2\alpha k\left(k+\phi c+\phi^2 m\right)} \tag{3.38}$$

By adding both solutions it is found that:

$$x_1(t)=C_3e^{\chi t}+C_4e^{\sigma t}+\frac{C_1\theta e^{\theta t}}{2\alpha k\left(k+\theta c+\theta^2 m\right)}+\frac{C_2\phi e^{\phi t}}{2\alpha k\left(k+\phi c+\phi^2 m\right)} \tag{3.39}$$

And using equation (3.16), the expression for the state variable x_2 is given as:

$$x_2(t) = C_3\chi e^{\chi t} + C_4\sigma e^{\sigma t} + \frac{C_1\theta^2 e^{\theta t}}{2\alpha k\left(k + \theta c + \theta^2 m\right)} + \frac{C_2\phi^2 e^{\phi t}}{2\alpha k\left(k + \phi c + \phi^2 m\right)} \qquad (3.40)$$

Since both initial and final states are known from the boundary conditions, it is possible to find the four constants (C_1, C_2, C_3, C_4) of integration by solving the algebraic relations for four equations and four unknowns.

In an example the values in Table 3.1 are used as specifications of the second order oscillator and the values in Table 3.2 are the desired boundary conditions.

Second Order Oscillator	
Parameters	
m [kg]	1
k [N/m]	1
c [Ns/m]	0.1

Table 3.1 Second order oscillator parameter values

Boundary Conditions		
	to	tf
time [s]	0	10
x1 (t) [m]	0	0.6
x2 (t) [m/s]	0	1

Table 3.2 Boundary conditions for control of the oscillator

Using the values in Tables 3.1 and 3.2, and for a weighting factor in the cost function of $\alpha = 10$, the constants of integration for this example are given as follows:

$$C_1 = -0.0154 + 2.1348i$$
$$C_2 = -0.0154 - 2.1348i$$
$$C_3 = 0.0039 - 0.5341i$$
$$C_4 = 0.0039 + 0.5341i$$

Using these constants of integration in equations (3.39), (3.40) and (3.33), the optimal control solution is finally obtained in closed form. The results are shown in figures 3.1 to 3.3.

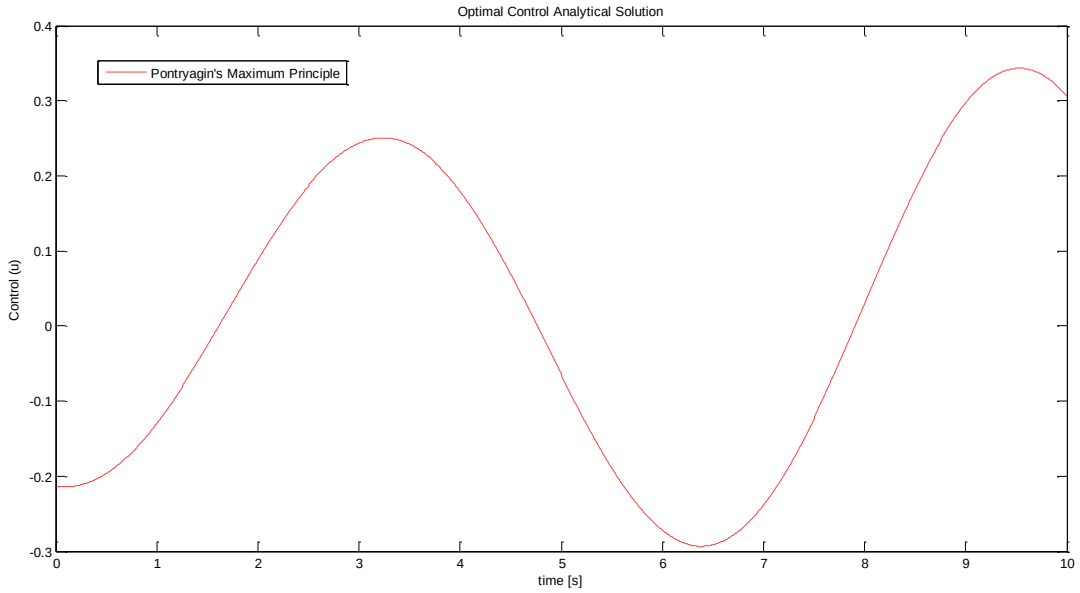

Figure 3.1 Optimal control u for the linear oscillator problem using Pontryagin Maximum Principle analytical solution

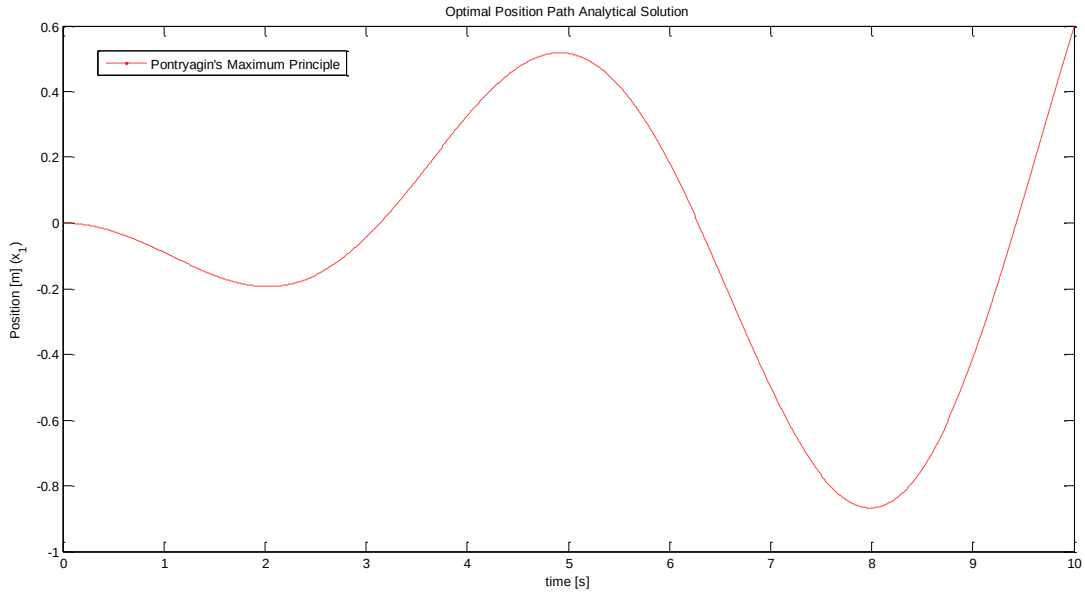

Figure 3.2 Optimal trajectory x_1 (position) for the linear oscillator problem using Pontryagin Maximum Principle analytical solution

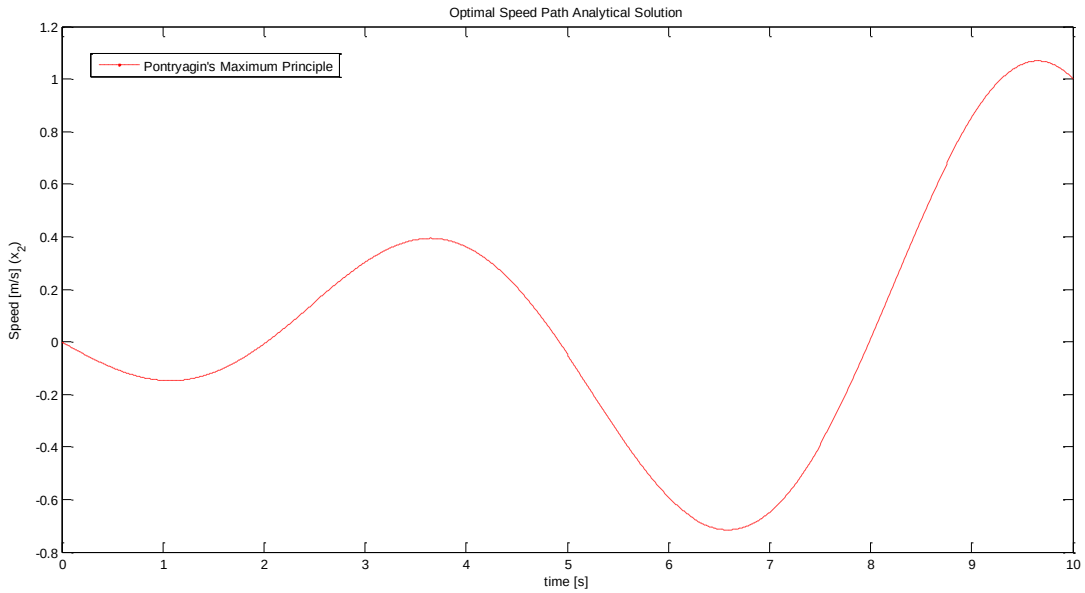

Figure 3.3 Optimal trajectory in x_2 (speed) for the linear oscillator problem using Pontryagin Maximum Principle analytical solution

Figure 3.1 shows the control u which drives the system optimally under the specifications given. Figures 3.2 and 3.3 show that the boundary conditions for the states are satisfied. With the results obtained, equation (3.18) is used to calculate the performance measure which gives the value $J = 4.19$ ('energy' dimensions).

These results are verified next by using a two-point boundary-value solver and later in the chapter by using the Dynamic Programming methodology.

Optimal Control solution for the linear Oscillator via Pontryagin Maximum Principle and Two-Point Boundary-Value solution

As mentioned earlier, the Pontryagin Maximum Principle analytical solution for more difficult problems, represents a huge challenge. Therefore the two-point boundary-value problem is normally solved using numerical methods or computational tools, like Matlab. Since for this example, the closed form solution has been found a numerical solution is used to verify the results and to gain experience when no analytical solution will be available (such as for the KERS problem). Therefore the Matlab function (bvp4c) is used to solve the boundary-value problem; the results are shown in figures 3.4 to 3.6.

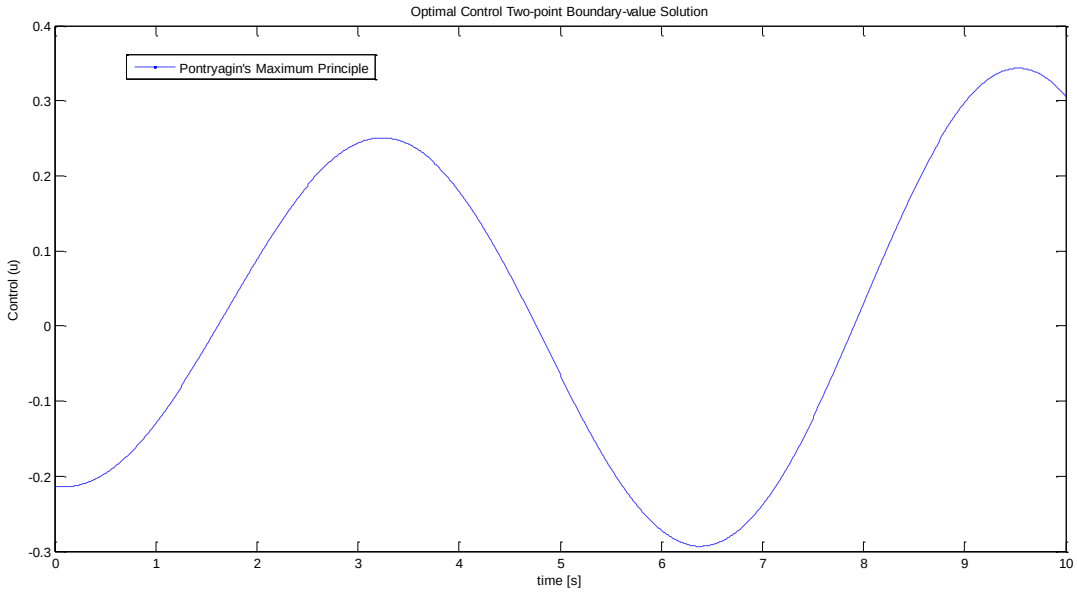

Figure 3.4 Optimal control u for the linear oscillator problem using Pontryagin Maximum Principle numerical solution of the TPBV

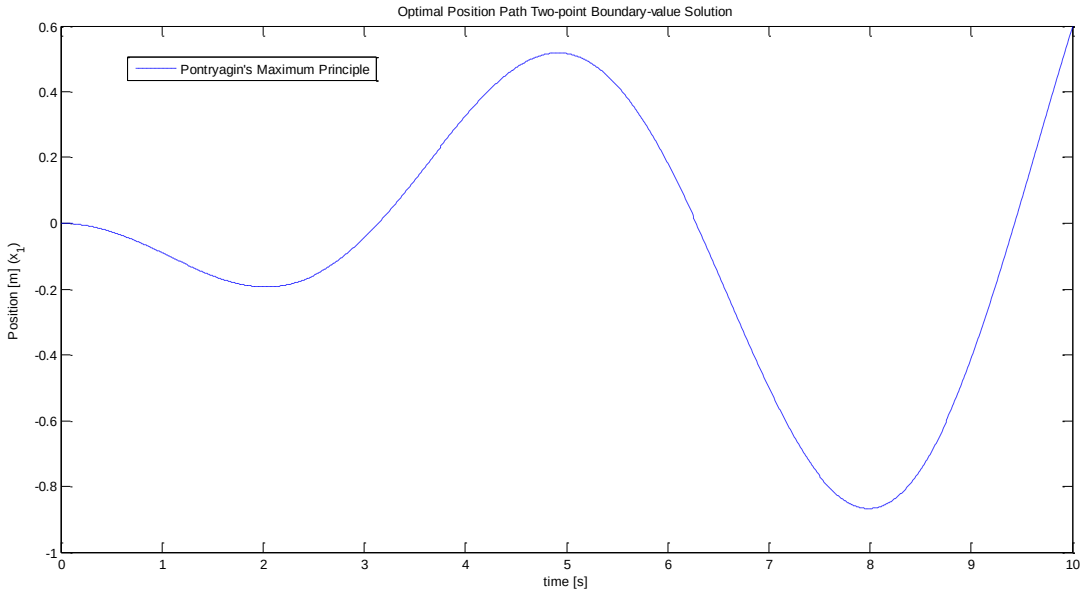

Figure 3.5 Optimal trajectory x_1 (position) for the linear oscillator problem using Pontryagin Maximum Principle numerical solution of the TPBV

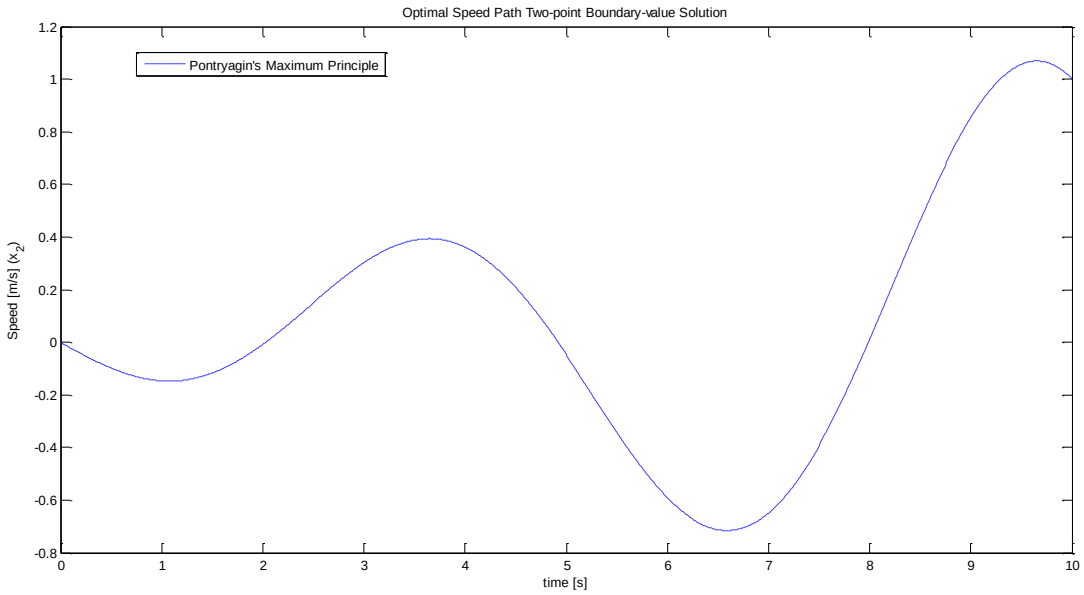

Figure 3.6 Optimal trajectory in x_2 (speed) for the linear oscillator problem using Pontryagin Maximum Principle numerical solution of the TPBV

Figure 3.4 shows the optimal control found, and figures 3.5 and 3.6 show the trajectories for the states, which meet the specified boundary conditions; and evaluating equation (3.18) the performance measure is $J = 4.196$ ('energy' units).

The results presented in figures 3.4, 3.5 and 3.6 show complete consistency with the solutions presented in figures 3.1, 3.2 and 3.3. The performance measurement is the same for both cases. The comparison is presented in figures 3.4b, 3.5b and 3.6b.

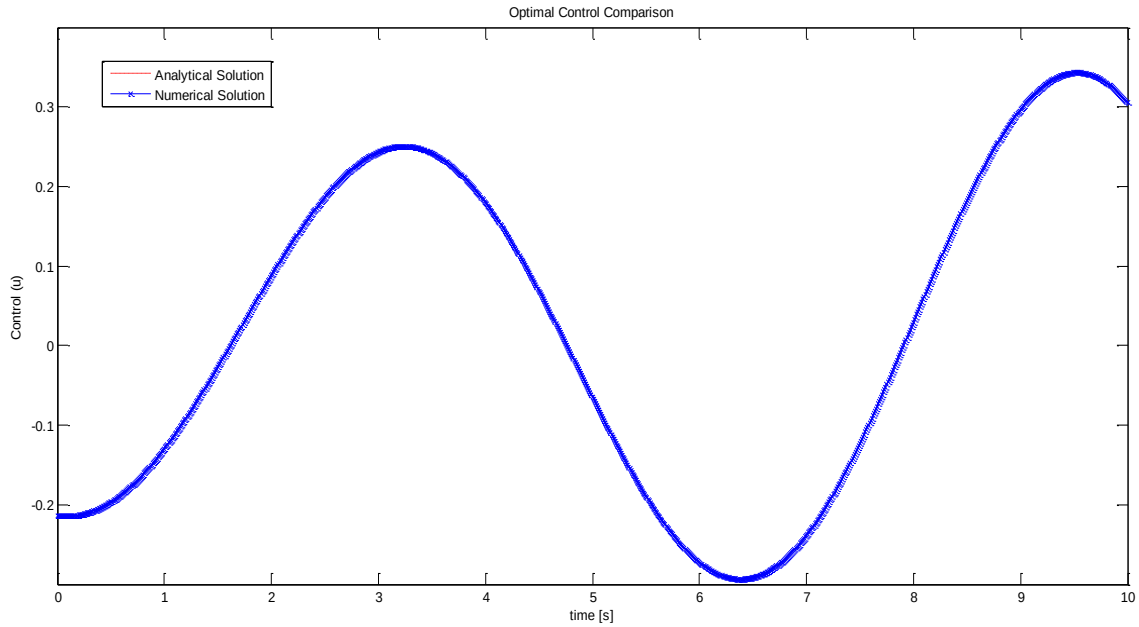

Figure 3.4b Optimal control u for the linear oscillator problem comparison between analytical and numerical solutions

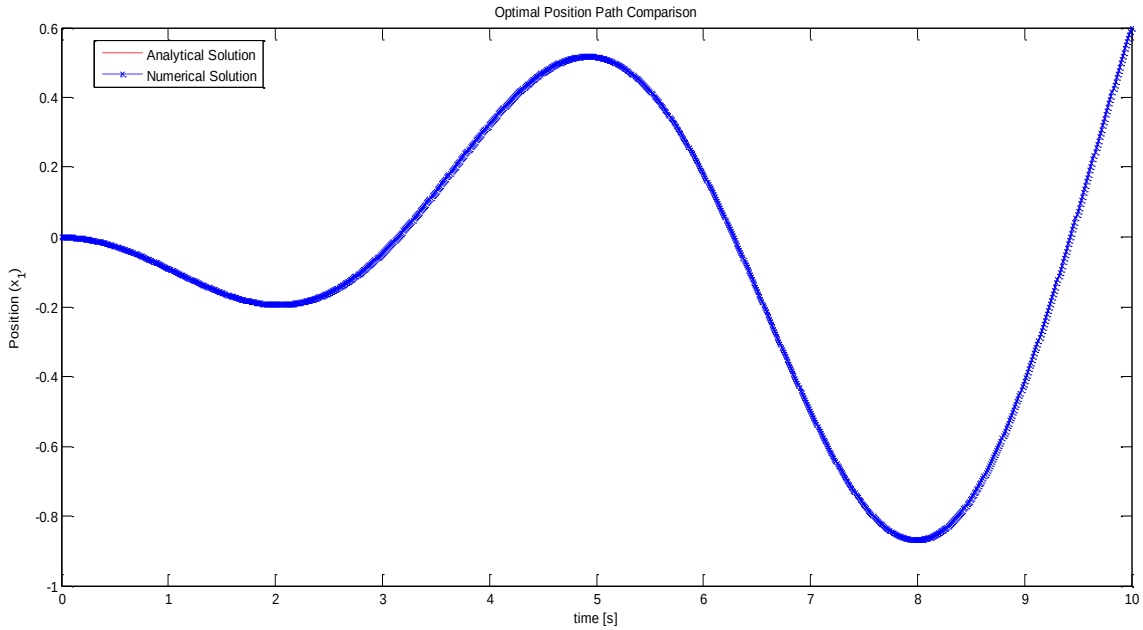

Figure 3.5b Optimal trajectory x_1 (position) for the linear oscillator problem comparison between analytical and numerical solutions

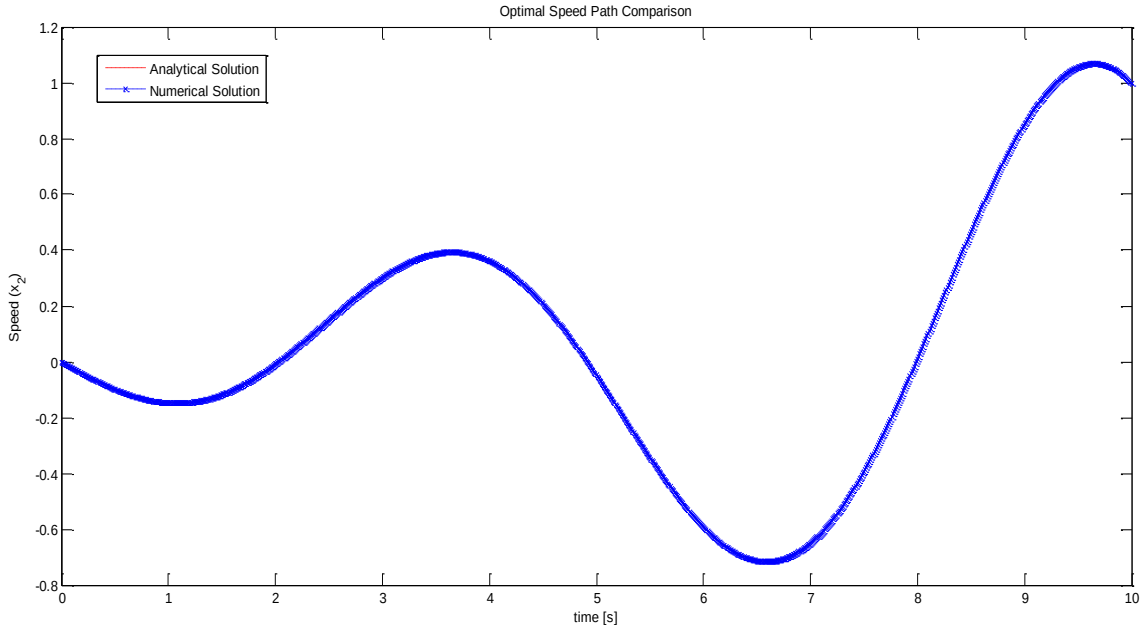

Figure 3.6 Optimal trajectory in x_2 (speed) for the linear oscillator problem comparison between analytical and numerical solutions

For a linear system, Pontryagin's Maximum Principle is not difficult to implement, but it can be seen in the analytical solution that the problem can easily become a mathematical challenge. Moreover, when considering nonlinear problems, analytical solutions are no longer available and numerical methods have to be used. Also, as mentioned in section 3.2, since Pontryagin's Maximum Principle only deals with necessary conditions, there is a degree of uncertainty about the global optimum being achieved. For these reasons, the implementation of another methodology, like Dynamic Programming, is found to be wholly appropriate.

Application of Dynamic Programming

Optimal Control of the Linear Oscillator via Standard Dynamic Programming

The implementation of a Dynamic Programming strategy is now considered. A standard Dynamic Programming solution has been developed, for the same oscillator control problem. The importance of using interpolation or another method to reduce the error will be shown while applying what will be called standard Dynamic Programming.

The Computer Code

In this example, the procedure described in section 3.3 is followed to create a Dynamic Programming solution numerically. The computer code is given a set of admissible state values and controls, and it uses the Principle of Optimality to select the control with the lowest performance measure. For each discrete time step, starting from the last one, the system takes an admissible state $[x_1(t) \quad x_2(t)]$, applies an admissible control u_1, and evaluates the state where the system is driven $[x_1(t+1) \quad x_2(t+1)]$ (discarding any results that take the system outside the admissible range). This result is then compared with the admissible states, and once it finds the closer one, the performance measure for that step is calculated (using the admissible control u_1), adding the accumulative performance measure up to that point. After applying all the admissible controls, the code selects the best option and stores the total performance measurement to be used later as the accumulated cost.

To set up the ranges for the state and control admissible variables, the solution from Pontryagin's Maximum Principle will be used in order to avoid an exploratory run that might be needed if the system is completely unknown. The input values are shown in Table 3.3.

Admissible Values for Dynamic Programming (case 1)	
x1 (position) [m]	[-0.9:0.02:0.8]
x2 (speed) [m/s]	[-1:0.02:1.3]
u (control)	[-0.5:0.02:0.5]
k (number of steps)	40

Table 3.3 Admissible values used in the Dynamic Programming optimal control solution of the linear oscillator

The admissible values contained in table 3.3 can be read like this: a vector of values is created from the minimum to the maximum value in increments of 0.02. The systems specifications and boundary conditions are taken from Tables 3.1 and 3.2. The results and the comparison with Pontryagin's Maximum Principle, are shown in figures 3.7 to 3.9.

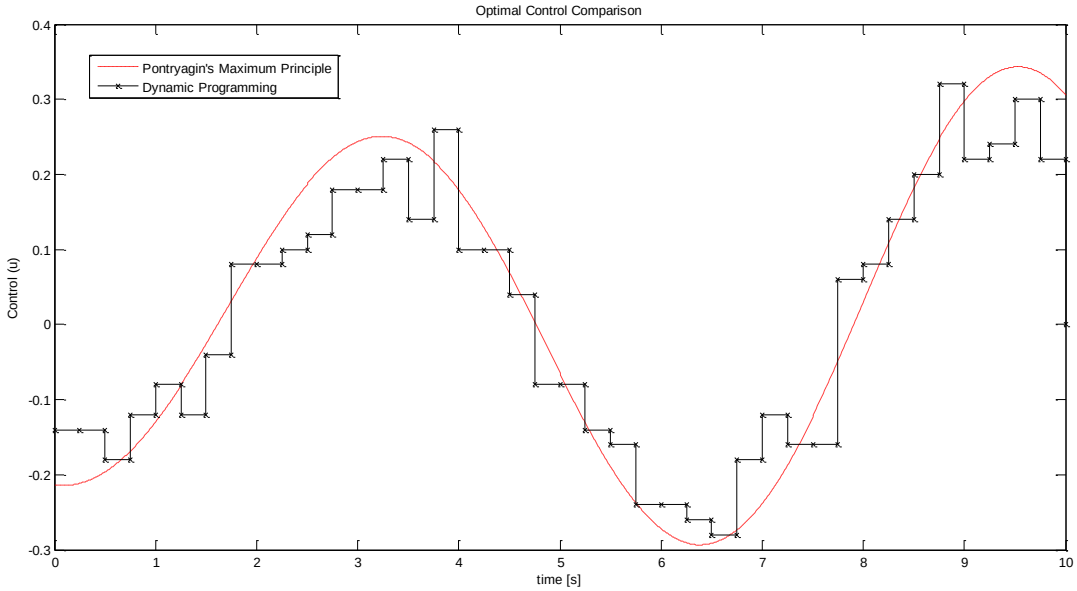

Figure 3.7 Optimal control comparison between Dynamic Programming and Pontryagin Maximum Principle

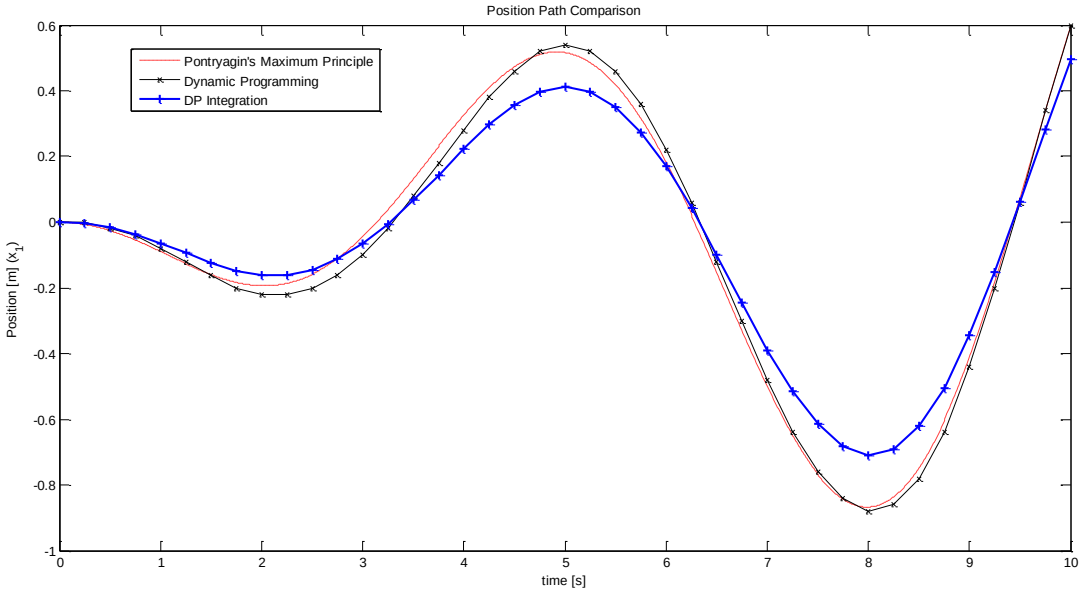

Figure 3.8 Position path comparison between both methods, and the integration of the control obtained using Dynamic Programming

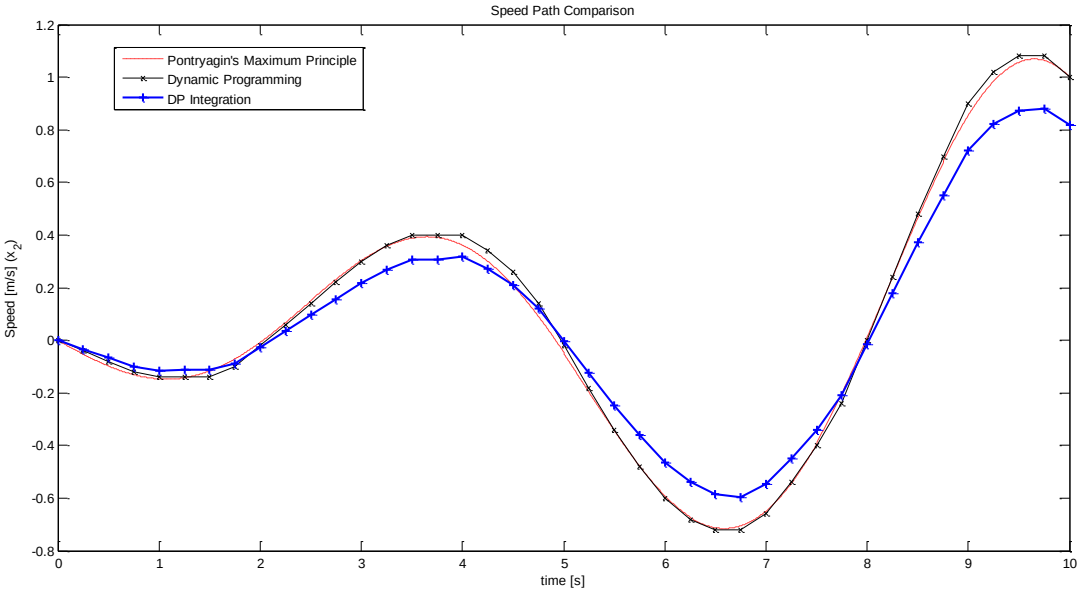

Figure 3.9 Speed path comparison between both methods, and the integration of the control obtained using Dynamic Programming

The Dynamic Programming solution shown in figure 3.7 follows closely Pontryagin Maximum Principle solution, considering the discretization of the system. Figures 3.8 and 3.9 show that the chosen trajectory using Dynamic Programming is consistent with Pontryagin's Maximum Principle solution, however the curve representing the integration of the control shown in figure 3.7 presents a steady state error.

The lack of interpolation and the "rounding" errors from selecting the nearest neighbour are responsible for the steady state error. In the results it can be seen that the optimal route is chosen but when integrating the control obtained, the actual path is different.

It is clear that an interpolation or an equivalent strategy is needed in order to eliminate the error and reach consistency between both methods. However, since linear interpolation does not work for this kind of system, it will not work for a more complicated model such as a KERS, therefore another solution has to be found.

Optimal Control of the Linear Oscillator via Nonstandard Discrete Dynamic Programming

Exploring different ideas of how to get the right control variable with a more precise strategy than interpolation, and one that could be suitable for a nonlinear system, is the problem that needs to be solved. Standard Dynamic Programming normally evaluates the system state from a specific starting point and, by applying a control u, the state at the following time step is found by integration. The proposed new strategy is instead of having a set of allowable control variables to drive the system, the approach is to find the control parameters in an assumed form of u at each time step, from the initial state to the final state by inverting the integration process. This approach is based on the mathematical characteristics of a discrete system, meaning that for a number of algebraic equations it is possible to solve for the same number of unknowns; for Dynamic Programming, given an initial and final states at an iteration time it is possible to find the control variable.

In standard Dynamic Programming, due to the discretization of the state and control variables, a numerical approximation for integration purposes is normally used. O'Neil (1995) describes in detail some methods that are frequently used, these are: the Euler method, the Taylor method and the Runge Kutta method. When implementing standard Dynamic Programming the Euler method is used in most cases, however the first order method is not the very accurate. Higher order methods are often used to achieve higher accuracy (especially when using larger time steps).

One of the main reasons for using the Euler method while implementing standard Dynamic Programming, is that the systems representation in state space already forms Euler's algorithm after discretization. In the case of the oscillator equations (3.16) and (3.17) in discrete form are given as:

$$\frac{\Delta x_{1k}}{\Delta t} = x_{2k} \tag{3.41}$$

and

$$\frac{\Delta x_{2k}}{\Delta t} = \frac{u_k - cx_{2k} - kx_{1k}}{m} \tag{3.42}$$

where the suffix k represents the time step. And using equations (3.41) and (3.42), the discrete states are given as:

$$x_{1k+1} = x_{1k} + x_{2k}\Delta t \tag{3.43}$$

and

$$x_{2k+1} = x_{2k} + \frac{u_k - cx_{2k} - kx_{1k}}{m}\Delta t \tag{3.44}$$

In order to invert the integration process, the equations must be solved for u (the control variable). For the second order oscillator only equation (3.44) is function of the control variable, therefore the desired equation is given as:

$$\frac{(x_{2k+1} - x_{2k})m}{\Delta t} + cx_{2k} + kx_{1k} = u_k \tag{3.45}$$

Before implementing this strategy for an optimal control solution, it is necessary to test that given a known path, it is possible to determine the control that forces the system to follow the given path. To do this, the solution from Pontryagin's Maximum Principle is used, therefore at each time step, both initial and final states are known. Figure 3.10 shows a set of results.

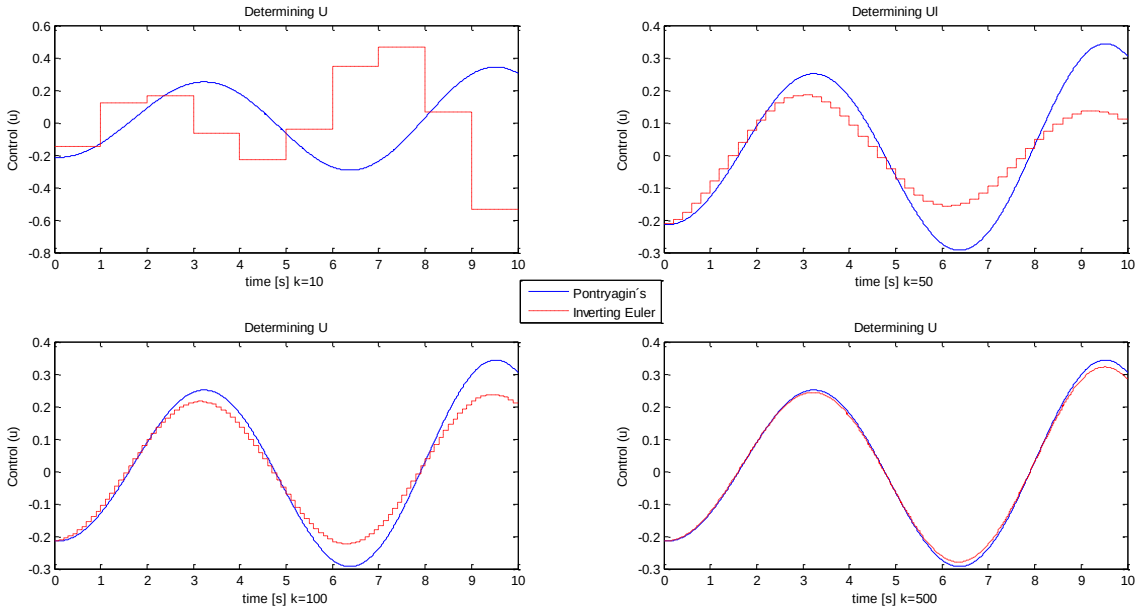

Figure 3.10 Optimal controls found by inverting the Euler integration method compared with, using Pontryagin Maximum Principle states trajectories (where the optimal trajectory is known).

The results in figure 3.10 show that an error is present, especially when using a small number of steps. For a large number of steps the control obtained follows the known solution closely.

The results show that when the trajectory is known, it is possible to find the control that drives the system between two specified points. It also shows that the strategy converges as the number of steps is increased. However it is also clear that a high number of steps is needed for the system to have an accurate response.

To attempt to achieve higher accuracy, and to reach convergence faster, the Runge Kutta numerical integration method is now considered. The Runge Kutta algorithm can be found in Appendix A, and its implementation for the oscillator is given in the following steps: for the states, and the control variable, the general equations are given as:

$$x_{1k+1} = x_{1k} + \frac{1}{6}\left[Wk_{11} + 2Wk_{21} + 2Wk_{31} + Wk_{41}\right]\Delta t \qquad (3.46)$$

which represents the discretized first state, and

$$x_{2k+1} = x_{2k} + \frac{1}{6}\left[Wk_{12} + 2Wk_{22} + 2Wk_{32} + Wk_{42}\right]\Delta t \tag{3.47}$$

which represents the discretized second state, and

$$u_{k+1} = u_k + \frac{1}{6}\left[Wk_{1u} + 2Wk_{2u} + 2Wk_{3u} + Wk_{4u}\right]\Delta t \tag{3.48}$$

which represents the discretized control variable, where the first set of equations are given as:

$$Wk_{11} = x_{2k} \tag{3.49}$$

for the discretized first state, and

$$Wk_{12} = \frac{u_k - cx_{2_k} - kx_{1_k}}{m} \tag{3.50}$$

for the discretized second state, and

$$Wk_{1u} = u_g \tag{3.51}$$

for the discretized control variable. Where the variable u_g is the gradient of the control, which in this example is given only constant values allowing the control to be constant or linear for each discrete step.

The second set of equations are given as:

$$Wk_{21} = x_{2k} + Wk_{12} * \frac{\Delta t}{2} \tag{3.52}$$

for the discretized first state, and

$$Wk_{22} = \frac{u_k + Wk_{1u} * \frac{\Delta t}{2} - c\left(x_{2k} + Wk_{12} * \frac{\Delta t}{2}\right) - k\left(x_{1k} + Wk_{11} * \frac{\Delta t}{2}\right)}{m} \tag{3.53}$$

for the discretized second state, and

$$Wk_{2u} = u_g \tag{3.54}$$

for the discretized control variable.

The third set of equations are given as:

$$Wk_{31} = x_{2k} + Wk_{22} * \frac{\Delta t}{2} \tag{3.55}$$

for the discretized first state, and

$$Wk_{32} = \frac{u_k + Wk_{2u} * \frac{\Delta t}{2} - c\left(x_{2k} + Wk_{22} * \frac{\Delta t}{2}\right) - k\left(x_{1k} + Wk_{21} * \frac{\Delta t}{2}\right)}{m} \tag{3.56}$$

for the discretized second state, and

$$Wk_{3u} = u_g \tag{3.57}$$

for the discretized control variable.

Finally, the fourth set of equations are given as:

$$Wk_{41} = x_{2k} + Wk_{32} * \Delta t \tag{3.58}$$

for the discretized first state, and

$$Wk_{42} = \frac{u_k + Wk_{3u} * \Delta t - c\left(x_{2k} + Wk_{32} * \Delta t\right) - k\left(x_{1k} + Wk_{31} * \Delta t\right)}{m} \tag{3.59}$$

for the discretized second state, and

$$Wk_{4u} = u_g \tag{3.60}$$

for the discretized control variable.

Substituting equations (3.49) to (3.60) into (3.46) to (3.48), (and grouping the terms that do not depend on u) the final expressions are given as:

$$x_{1k+1} = x_{1k} + \frac{1}{6}\left[\begin{array}{l} x_{2_k} + 2\left(\frac{u_k \Delta t}{2m} + B\right) + 2\left(\frac{u_k \Delta t}{2m}\left(1 - \frac{c\Delta t}{2m}\right) + \frac{u_g \Delta t^2}{4m} + E\right) \\ \quad + \frac{u_k \Delta t}{m}\left(1 - \frac{c\Delta t}{2m}\left(1 - \frac{c\Delta t}{2m}\right) - \frac{k\Delta t^2}{4m}\right) + \frac{u_g \Delta t^2}{2m}\left(1 - \frac{c\Delta t}{2m}\right) + G \end{array}\right]\Delta t \tag{3.61}$$

and

$$x_{2k+1} = x_{2k} + \frac{1}{6}\left[\begin{array}{l} \frac{u_k}{m} + A + 2\left(\frac{u_k}{m}\left(1 - \frac{c\Delta t}{2m}\right) + \frac{u_g \Delta t}{2m} + D\right) \\ \quad + 2\left(\frac{u_k}{m}\left(1 - \frac{c\Delta t}{2m}\left(1 - \frac{c\Delta t}{2m}\right) - \frac{k\Delta t^2}{4m}\right) + \frac{u_g \Delta t}{2m}\left(1 - \frac{c\Delta t}{2m}\right) + F\right) \\ \quad + \frac{u_k}{m}\left(1 - \frac{c\Delta t}{m}\left(1 - \frac{c\Delta t}{2m}\left(1 - \frac{c\Delta t}{2m}\right) - \frac{k\Delta t^2}{4m}\right) - \frac{k\Delta t^2}{2m}\left(1 - \frac{c\Delta t}{2m}\right)\right) \\ \quad + \frac{u_g dt}{m}\left(1 - \frac{c\Delta t}{2m}\left(1 - \frac{c\Delta t}{2m}\right) - \frac{k\Delta t^2}{4m}\right) + H \end{array}\right]\Delta t \tag{3.62}$$

and

$$u_{k+1} = u_k + u_g \Delta t \tag{3.63}$$

where the coefficients in the discretized set of equations are given by:

$$A = \frac{-cx_{2_k} - kx_{1_k}}{m} \tag{3.64}$$

and

$$B = x_{2k} + A\frac{dt}{2} \tag{3.65}$$

and

$$D = -\frac{cB}{m} - \frac{k}{m}\left(x_{1k} + x_{2k}\frac{dt}{2}\right) \tag{3.66}$$

and

$$E = x_{2k} + D\frac{dt}{2} \tag{3.67}$$

and

$$F = -\frac{Ec}{m} - \frac{kx_{1k}}{m} - \frac{Bkdt}{2m} \tag{3.68}$$

and

$$G = Fdt + x_{2k} \tag{3.69}$$

and

$$H = -\frac{Gc}{m} - \frac{kx_{1k}}{m} - \frac{Ekdt}{m} \tag{3.70}$$

In equations (3.61) and (3.62) it can be seen that since the Runge Kutta algorithm is a recursive method, both of the state equations are functions of the control variable. For this reason it is necessary to solve the pair of equations algebraically in order to find the control variable. In addition, equation (3.48) shows the possibility of using a time dependent control variable, unlike the Euler method which can only evaluate constant controls for each discrete step.

Considering the first case, for which the control variable can only take constant values $\left(u_g = 0\right)$, by doing this, it can be seen that the use of equations (3.61) and (3.62) create a problem of two equations and one unknown (i.e. an oversubscribed system), for which the solution can be obtained by using the least square method. As done earlier, the inverting method is tested using Pontryagin's Maximum Principle solution as a known given path. The results are shown in figure 3.11

The results in figure 3.11 show that the solutions obtained follow closely the given solution, even when a low number of steps is used.

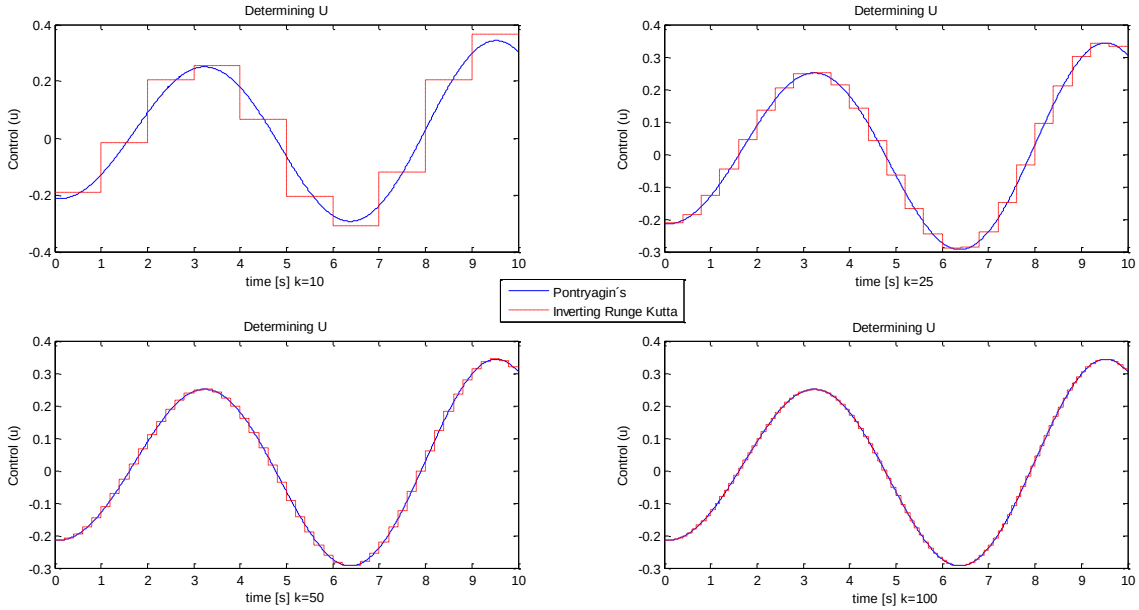

Figure 3.11 Optimal controls found by inverting Runge Kutta integration method and using Pontryagin's Maximum Principle states trajectories (where the optimal trajectory is known).

By comparing figures 3.10 and 3.11 it is clear that for determining the control variable of the linear oscillator, Runge Kutta integration represents an improvement both in accuracy and convergence time. The Euler method required at least 500 points in order to converge and eliminate the integration error whereas Runge Kutta integration requires less than 10% of the points to achieve a comparative solution, and it almost totally eliminates the integration truncate errors even for large time steps.

Modified Discrete Dynamic Programming Strategy with Constant Control

Since the previous strategy determines the control given the state trajectory and the initial and final states, a modified version of Dynamic Programming was created which, given a set of discrete admissible states, finds the control ('integrating forwards') and then by using the principle of optimality, selects the optimal path, whereas standard Dynamic Programming follows 'backwards integration' and only allows a set of admissible controls at each time step.

In an example, the admissible values to be used are shown in Table 3.4 and the system specifications and boundary conditions are drawn from Tables 3.1 and 3.2.

Admissible Values for Dynamic Programming (case 2)	
x1 (position) [m]	[-0.9:0.1:0.7]
x2 (speed) [m/s]	[-0.8:0.1:1.3]
k (number of steps)	10

Table 3.3 Admissible values used in the modified version of Dynamic Programming

However when implementing this strategy the modified discrete Dynamic Programming approach fails; after doing several tests it was found that for some combinations where the solution is not unique and especially for the ones that are unachievable by using constant control (or physically impossible to achieve), the least squared solution is compromised and the control found is no longer representative of the system behaviour even when it seems to be meeting the boundary conditions. The results are shown in figures 3.12, 3.13, and 3.14.

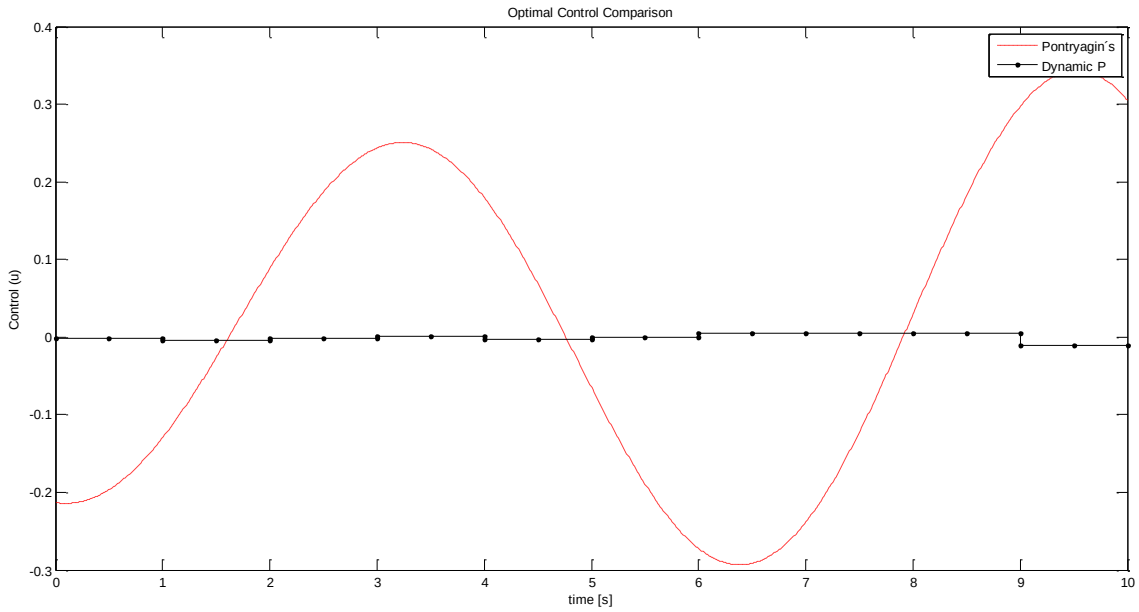

Figure 3.12 Control found using the modified dynamic programming program by determining the constant control using the Runge Kutta integration method.

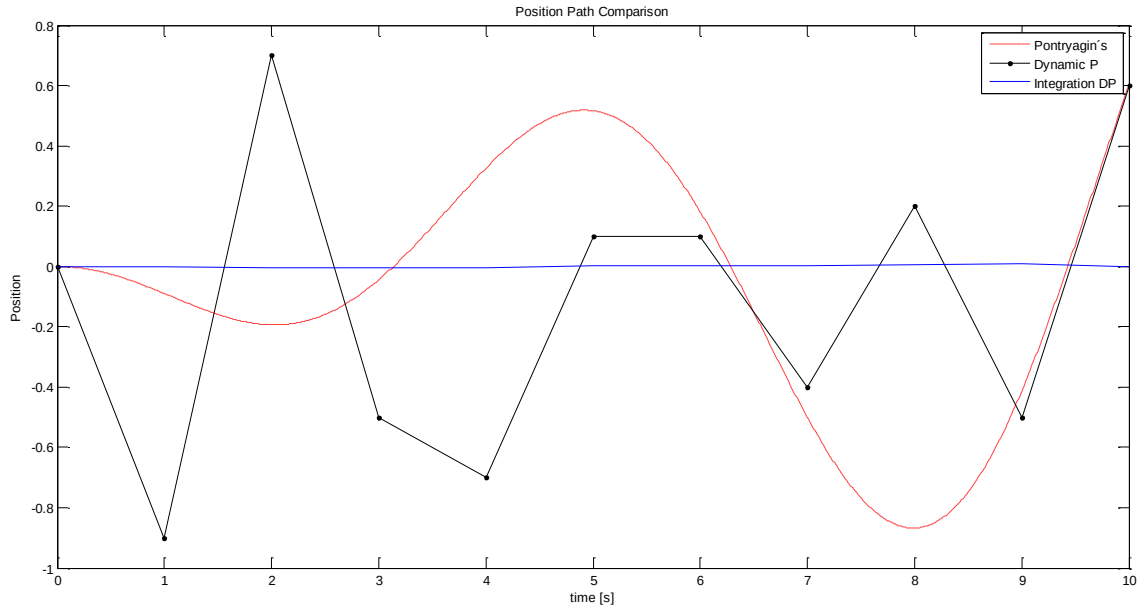

Figure 3.13 Comparison between position paths for Pontryagin's Maximum Principle and the modified dynamic programming program.

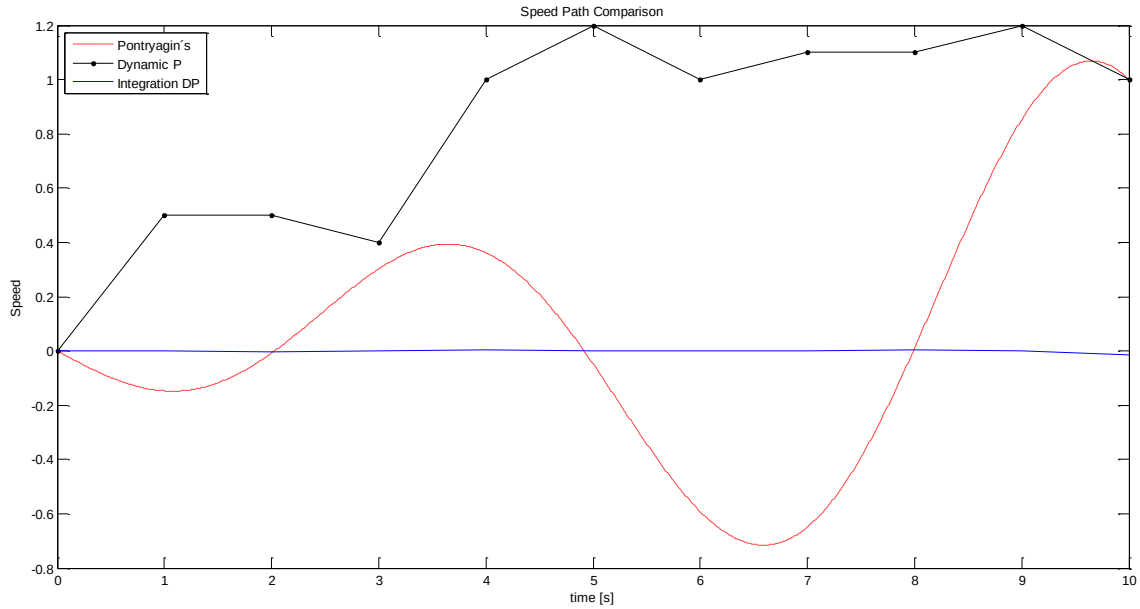

Figure 3.14 Comparison between speed paths for Pontryagin's Maximum Principle and the modified dynamic programming program.

The Dynamic Programming result in figure 3.12 shows an almost constant zero control. Figures 3.13 and 3.14 show that the trajectories obtained with Dynamic Programming meet the boundary conditions but are not close to the solution obtained using Pontryagin's Maximum Principle. They also show that the integration of the control does not follow the trajectory chosen.

In figures 3.11, 3.12, and 3.13 it is clearly shown that after integrating the results found using constant control in the modified version of Dynamic Programming, the system does not follow the trajectory for which that control was found. As mentioned before, this is because the solution is compromised and therefore it is necessary to investigate new possibilities.

Modified Discrete Dynamic Programming with Linear Control

In order to eliminate the problem thus described, the first option is to assume linear control between the discrete steps. As in the previous case, the first test is to determine the control parameters given a known solution. In this case, equations (3.61) and (3.63) represent a system of two equations and two unknown parameters, which can be solved either as a system of algebraic equations or using a least square strategy. The results for a convergence analysis are shown in figure 3.15.

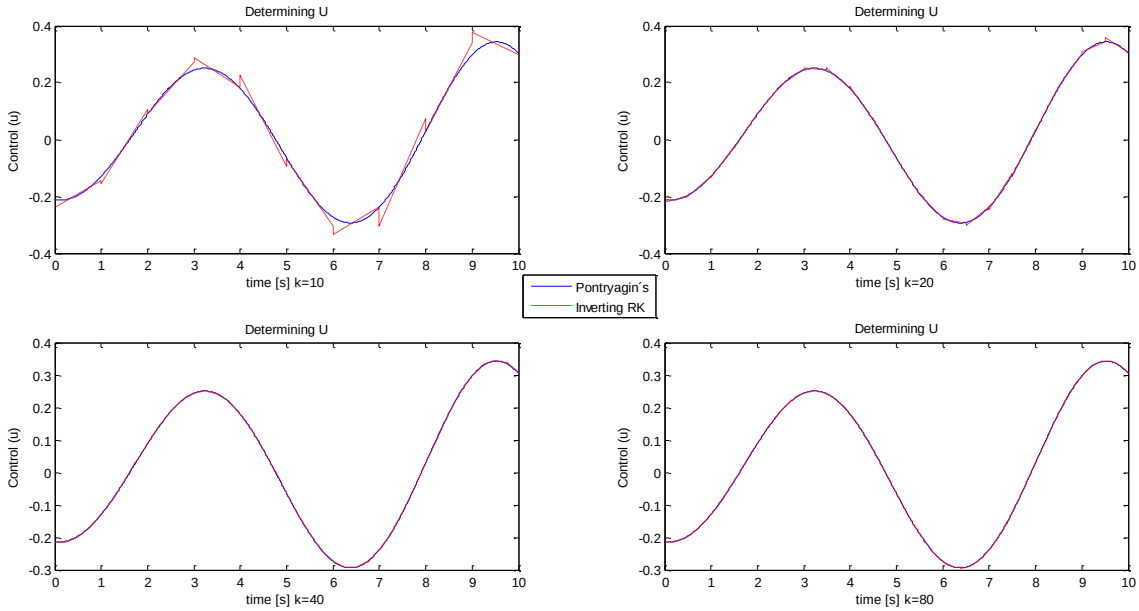

Figure 3.15 Optimal linear controls found by inverting Runge Kutta integration method and using Pontryagin states trajectories (where the optimal trajectory is known).

The results in figure 3.15 show a very accurate control is found. Using only ten time steps the solution is very good compared to the one given. And for twenty or more time steps, the solution can be considered identical.

In addition to correcting the problem which arises with being an oversubscribed system, figure 3.15 shows that the solution converges even for a reduced number of steps which is a very positive characteristic when implementing combinatorial problems. The determination of the control variable inverting Runge Kutta was then implemented in the modified dynamic programming problem, with input values drawn from Tables 3.1, 3.2 and 3.4. The results are shown in figures 3.16, 3.17, and 3.18.

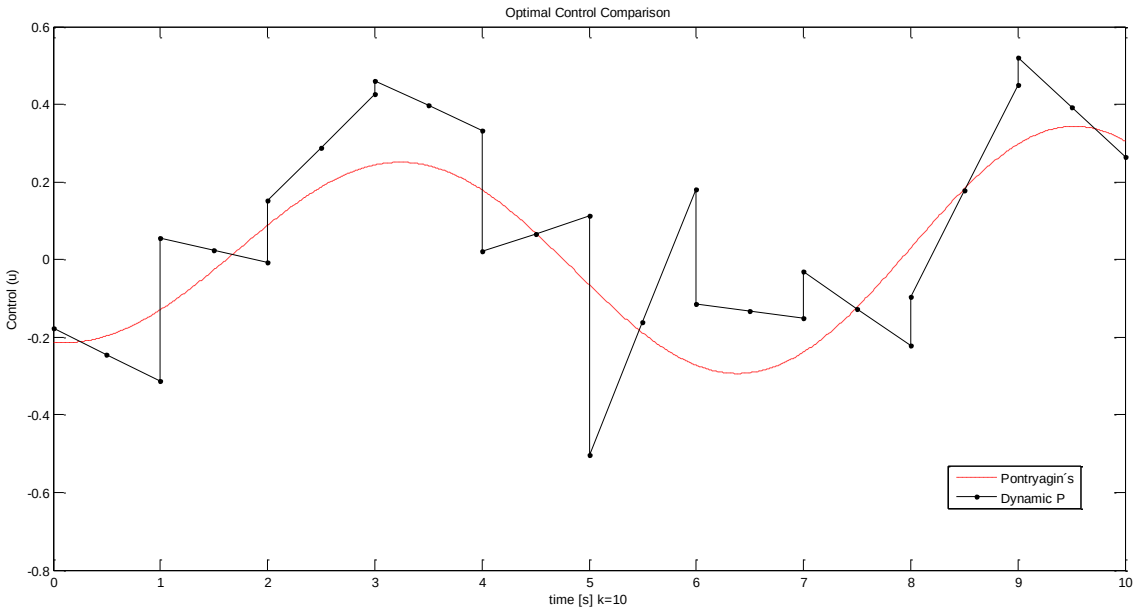

Figure 3.16 Comparison between Pontryagin Maximum Principle and the modified dynamic programming program optimal control solutions.

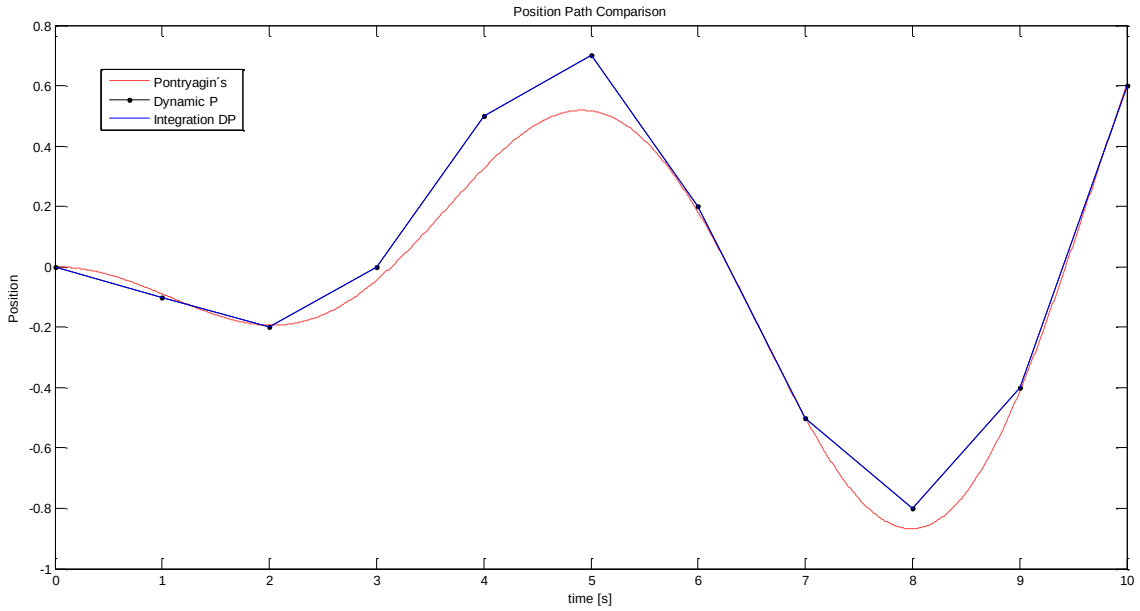

Figure 3.17 Comparison between Pontryagin Maximum Principle and the modified dynamic programming program optimal position trajectory.

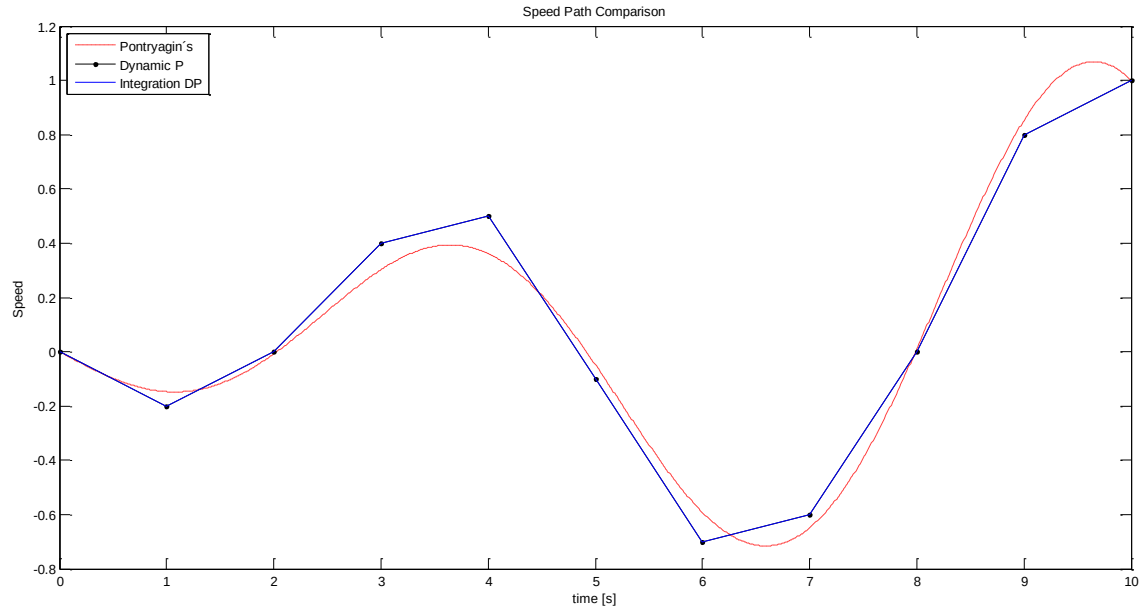

Figure 3.18 Comparison between Pontryagin Maximum Principle and the modified dynamic programming program optimal speed trajectory.

As seen in figures 3.16, 3.17, and 3.18 the solution found drives the system close to the Pontryagin Maximum Principle solution and it meets the boundary conditions. This means that the accuracy of the strategy proposed is good and it converges for a more refined selection of values for both states and time steps. The modified version of Dynamic Programming is however not efficient since

the computational effort to achieve good results is enormous and therefore further improvements have still to be made.

Improved Discrete Dynamic Programming with Linear Control

Knowing the capability for determining the control, given the initial and final states, a modification to the standard Dynamic Programming is now made in order to combine it with the previous strategy. Starting from the standard Dynamic Programming, that has a specific set of possible control variables which eliminates unnecessary combinations, the system is driven from a starting point by the selected (constant) control but instead of penalizing the error in the cost function, and carrying the state error, the nearest neighbourhood point is used as final point to determine the control that drives the system. This combination makes the most of both strategies since the main disadvantage for each case has been corrected. The input values to test the Improved Dynamic Programming version are drawn from Tables 3.1, 3.2, and 3.4, and the results are shown in figures 3.19, 3.20, 3.21.

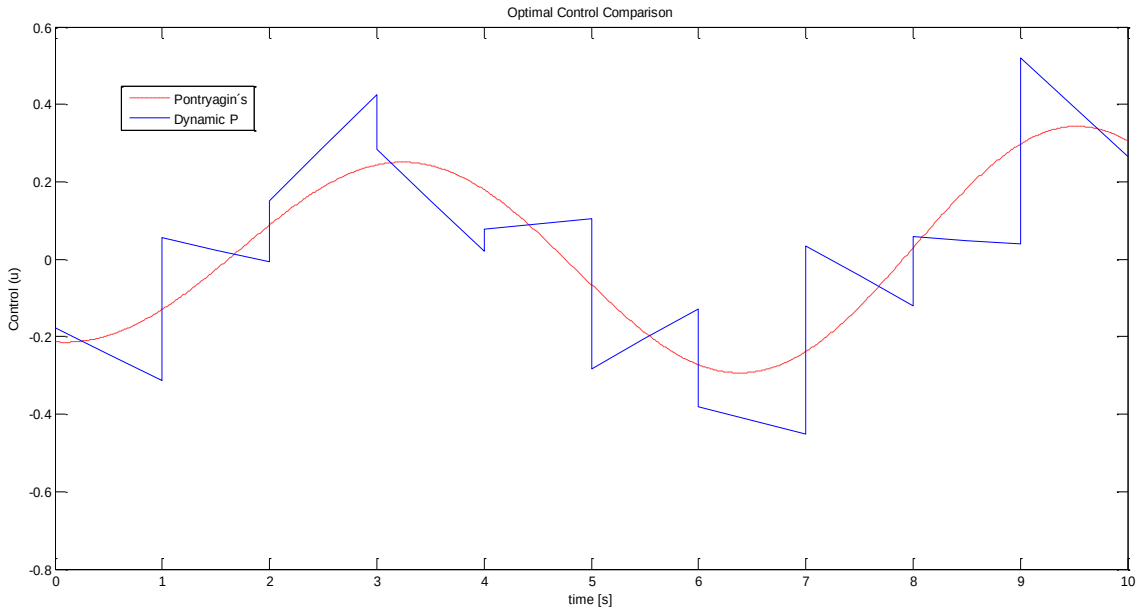

Figure 3.19 Comparison between Pontryagin Maximum Principle and the modified Dynamic Programming optimal control solutions.

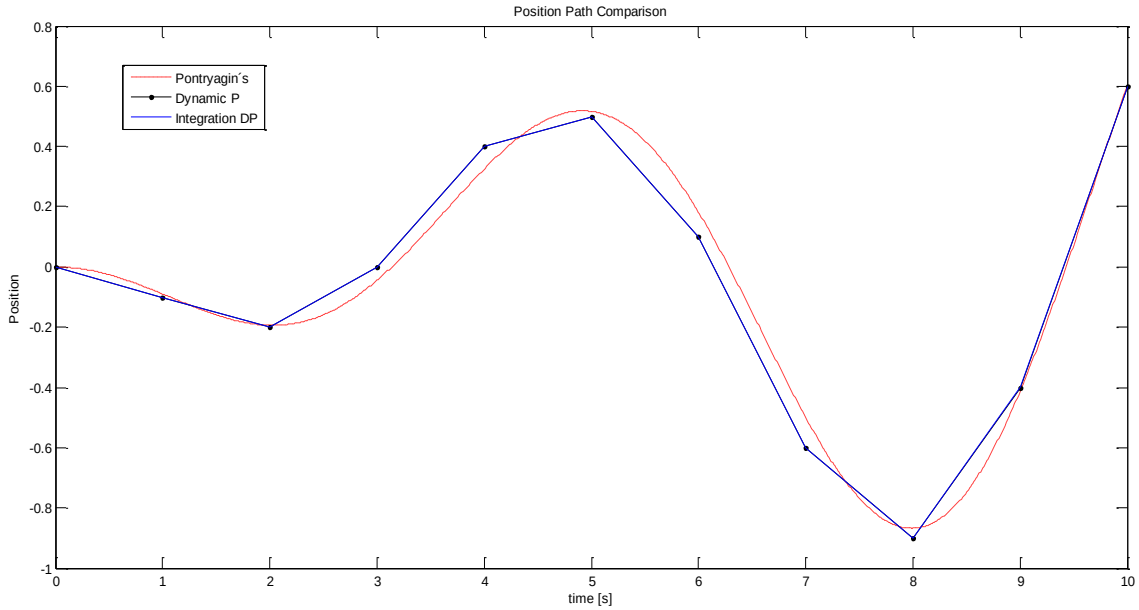

Figure 3.20 Comparison between Pontryagin Maximum Principle and the modified Dynamic Programming optimal position trajectory.

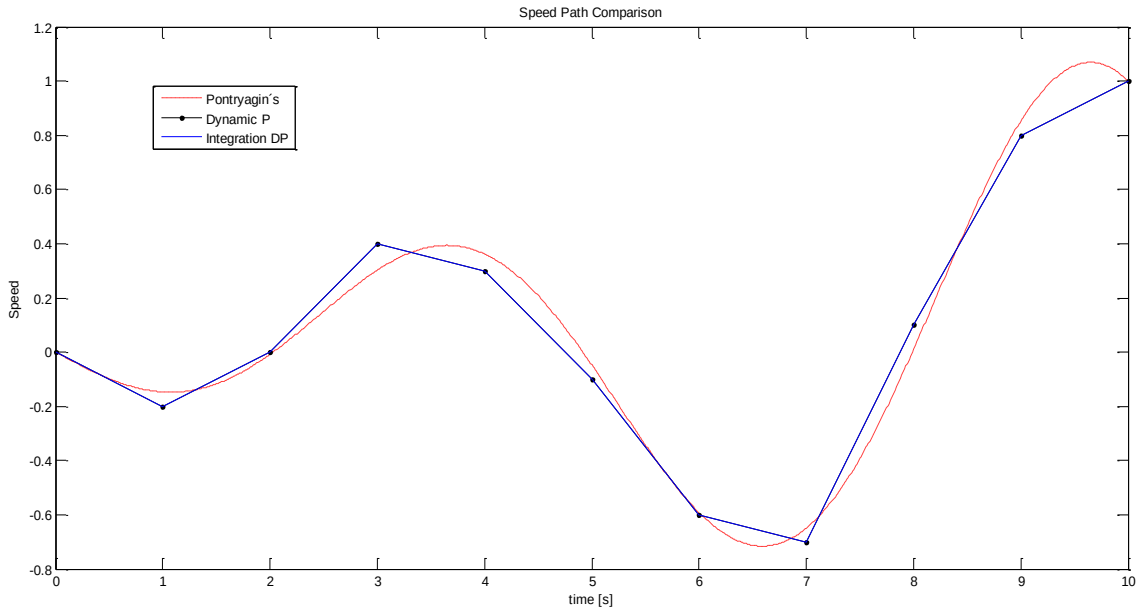

Figure 3.21 Comparison between Pontryagin Maximum Principle and the modified Dynamic Programming optimal speed trajectory.

As shown in figures 3.19, 3.20, and 3.21 the results are consistent with the improved Dynamic Programming and the modified version, the difference being in the computational time, which is reduced by 95%.

The previous results clearly show that the strategy of determining the control parameters considerably improves the accuracy as compared with the standard Dynamic Programming, and by combining both strategies an efficient and accurate solution is found.

3.5 Dynamic Programming Summarizing Table

Table 3.4 shows a compact summary of the different discrete Dynamic Programming versions explored in this Chapter. This table summarizes the most important features, their benefits and limitations

Discrete Dynamic Programming strategy	Main Characteristic	Accuracy	Computational Efficiency	Control Smoothness
Standard	• Backwards integration • From initial state uses admissible control and selects closest neighbourhood (no interpolation is used)	Poor	Very Good	Average
Modified	• Forwards Integration • Matches all admissible states and later uses the Principle of Optimality • Point-to-point control variable determination	Very Good	Poor	Sometimes Poor
Improved Modified	• Same as Standard but uses the selected closest state for control variable determination	Very Good	Good	Sometimes Poor

Table 3.4 Discrete Dynamic Programming versions explored

3.6 Conclusions of the findings in Chapter 3

Modern engineering systems and recent engineering requirements often make demands beyond the scope of the classical control approaches. Optimal control gives the flexibility of evaluating the quantitative system performance based on different indicators that are found to be useful, such as in terms of minimum time and energy. The two most common optimal control methodologies Pontryagin's Maximum Principle, and discrete Dynamic Programming have

been implemented on a linear oscillator, placing particular emphasis on Dynamic Programming because of its ability to provide both necessary and sufficient conditions for optimality and because it can be potentially applied where Pontryagin's Maximum Principle will become intractable. Moreover owing to the limitations of standard Dynamic Programming, and a commonly used interpolation strategy, a new modified version has been developed and implemented; this achieves higher accuracy by finding the control parameters at each iteration, which is done by inverting the integration process. While developing this new technique it was found that Euler method converges only for a very high number of time steps and introduces errors for reducing number of steps whereas the Runge Kutta method is highly accurate and converges quickly. The results show that, despite dealing with a discrete system, constant control is not useful since the system becomes oversubscribed has to be solved with a least square solution. Even if it seems promising when following the optimal trajectory, the solution was compromised when implemented in a modified version of Dynamic Programming. Therefore the need for a time dependent control was obvious and was then implemented; first in a modified Dynamic Programming and later in an improved version of Dynamic Programming where the results were successful. The objective of this study was to examine the most common optimal control methodologies in order to build up a computational strategy that can be implemented in more complex systems such as the KERS. This will be taken up in Chapter 4.

4. APPLICATION OF OPTIMAL CONTROL THEORY TO A FLYWHEEL-BASED KERS WITH A CVT

In this Chapter the optimal control strategies described and developed in Chapter 3 are applied, first to a simplified frictionless flywheel-based KERS with a CVT, and later to a more realistic model that includes friction. First, by using Pontryagin's Maximum Principle, the mathematical procedure is presented and the two-point boundary-value problem is solved using Matlab. Later the application of an improved Dynamic Programming is presented.

As in Chapter 3, for the application of Dynamic Programming to the KERS, the methodology for determining the control parameters is assessed and developed. Initially this uses the Runge Kutta integration method and results are presented and analysed but difficulties are found. Then use is made of the modified Euler integration method. After selecting the appropriate integration method to determine the control variable, a full Dynamic Programming solution is presented and compared with the Pontryagin's Maximum Principle solution for both the simplified and the more realistic KERS models. In the case of the frictionless model, the optimal control objective is to minimize the control variable; whereas for more realistic models the stored energy is maximized. An additional approach to a two-input system model is explored at the end of the Chapter, the additional input being associated with the friction brakes of the vehicle; this approach clearly shows the possibilities of optimal control in KERS applications. The motivation to explore the application of optimal control in KERS comes from the good deal of interest in the literature (presented in Chapter 1) on the control of KERS, and although the information about flywheel-based KERS is limited several publications have mentioned the need for developing and implementing strategies for KERS control.

4.1 Application of Optimal Control to a simplified flywheel-based KERS with a CVT

The simplified model described in Chapter 2 is now considered for optimal control application. In a similar approach to Chapter 3, to construct the optimal control problem, the first step is to establish the system model. In this case, this

procedure has been presented in Chapter 2, and from it the state space representation (equations (2.11), (2.12), and (2.14) to (2.17)) is recalled as:

$$x_1(t) = \theta_w(t) \tag{4.1}$$

and

$$\dot{x}_1(t) = \dot{\theta}_w(t) = x_2(t) \tag{4.2}$$

and

$$\dot{x}_2(t) = \frac{-J_f u(t) x_2(t) x_3^{0.5}(t)}{J_w + J_f x_3(t)} \tag{4.3}$$

and

$$\dot{x}_3(t) = 2u(t) x_3^{0.5}(t) \tag{4.4}$$

where J_w represents the moment of inertia for the wheel, J_f the moment of inertia for the flywheel, $\theta_w(t)$ is the angular displacement of the wheel and $u(t)$ is the rate of change of the gear ratio ($\dot{G}(t)$).

The next step is to define the systems constraints and to specify the task to be performed. Since for an automotive KERS application the objective is to transfer the energy from the vehicle wheels to the flywheel as a regenerative process, the procedure is to drive the wheel to a lower speed by changing the gear ratio of the CVT. As outlined in Chapter 3, the system is to be driven from an initial state to a final state in a given time.

It has to be noted that for this case the specifications do not relate to the angular position of the wheel (or flywheel). Therefore, since the variables of interest are not a function of the angular position, the system given in equations (4.1) to (4.4) can be represented in a simplified form:

$$x_1(t) = \omega_w(t) \tag{4.5}$$

and

$$f_1(t) = \dot{x}_1(t) = \dot{\omega}_w(t) = \frac{-J_f u(t) x_1(t) x_2^{0.5}(t)}{J_w + J_f x_2(t)} \tag{4.6}$$

and

$$f_2(t) = \dot{x}_2(t) = 2u(t)x_2^{0.5}(t) \tag{4.7}$$

After defining the desired task, the last step is to define a performance index, which in this case involves only the control variable. Its mathematical representation is given as:

$$J(t) = \int_0^{t_f} \alpha u(t)^2 dt \tag{4.8}$$

By specifying numerical values to the initial and final speeds, the construction of the optimal control problem is complete. With this, the full application of optimal control theory can be considered.

Application of Pontryagin's Maximum Principle

Turning to the use of Pontryagin's Maximum Principle, the formulation of the optimal control solution proceeds in a similar way to Chapter 3. The Hamiltonian for the simplified KERS model (equations (4.5) to (4.7)) is given as:

$$H(t) = \alpha u^2(t)\psi_0(t) + \frac{-J_f u(t)x_1(t)x_2^{0.5}(t)}{J_w + J_f x_2(t)}\psi_1(t) + 2u(t)x_2^{0.5}(t)\psi_2(t) \tag{4.9}$$

And, from equation (4.9), the co-state functions are given as:

$$\dot{\psi}_0(t) = 0 \tag{4.10}$$

and

$$\dot{\psi}_1(t) = \frac{J_f u(t)x_2^{0.5}(t)}{J_w + J_f x_2(t)}\psi_1(t) \tag{4.11}$$

and

$$\dot{\psi}_2(t) = \frac{0.5J_w J_f u(t)x_1(t)x_2^{-0.5}(t) - 0.5J_f^2 u(t)x_1(t)x_2^{0.5}(t)}{\left(J_w + J_f x_2(t)\right)^2}\psi_1(t) - u(t)x_2^{-0.5}(t)\psi_2(t) \tag{4.12}$$

And, the condition for optimality is given as:

$$\frac{\partial H(t)}{\partial u(t)} = 2\alpha u(t)\psi_0(t) - \frac{J_f x_1(t) x_2^{0.5}(t)}{J_w + J_f x_2(t)}\psi_1(t) + 2x_2^{0.5}(t)\psi_2(t) \tag{4.13}$$

From equation (4.13), the optimal control is given as:

$$u(t) = \frac{J_f x_1(t) x_2^{0.5}(t)}{2\alpha J_w + 2\alpha J_f x_2(t)}\frac{\psi_1(t)}{\psi_0(t)} - \frac{x_2^{0.5}(t)}{\alpha}\frac{\psi_2(t)}{\psi_0(t)} \tag{4.14}$$

When the state and co-state variables correspond to optimal paths. The formulation of the Optimal Control problem using Pontryagin's Maximum Principle for the simplified KERS is now complete. From equations (4.6), (4.7), (4.11), (4.12) and equation (4.14) can be seen that the analytical solution of this system of equations poses an enormous challenge. Therefore the problem is solved as a two-point boundary-value (TPBV) problem numerically.

Optimal Control solution for simplified KERS model via Pontryagin's Maximum Principle and two-point boundary-value problem solution

In order to solve the two-point boundary-value problem numerically using Matlab function (bvp4c), the following is required: i) a system representation as a set of differential equations (in state space form), in this case given in equations (4.6), (4.7), (4.11), (4.12) and equation (4.14); ii) a set of boundary conditions (involving at least as many conditions as equations), these are given in Table 4.2; and iii) an initial guess at the solution. Like many numerical strategies, a good initial guess (or starting point) is crucial for successful determination of the solution. In the case of complex systems this might be difficult to achieve and the solver may not reach stability. In fact most of the time, either an unstable solution or no-solution is found when physical or mathematical constrains are reached. In the case of the KERS, the gear ratio cannot take a value of zero (or infinite) which represents a spinning wheel connected to a static flywheel.

As an example KERS to demonstrate a Pontryagin solution, a test model is selected and the system model parameters are given in Table 4.1 and the boundary conditions are given in Table 4.2.

Simplified KERS	
Parameters	
α	40
Jw [kg*m^2*rad^-2]	2
Jf [kg*m^2*rad^-2]	1

Table 4.1 The parameters for a simplified KERS

Boundary Conditions		
	to	tf
time [s]	0	10
x1 (t) [rad/s]	60	10
x2 (t) [ratio^2]	1	106

Table 4.2 Example boundary condition

In the case of the simplified KERS model, the implementation of the two-point boundary-value problem needs further consideration than that presented in Chapter 3 for the linear oscillator. It can be observed that both problems (i.e. KERS and the oscillator problem) are in the same category as Case 1 in Kirk (1998), i.e. the initial and final states are specified with final time fixed. However solving the two-point boundary-value problem, is different owing to the simplification of the model since when both states are specified as initial and final conditions, one condition becomes redundant. This means that for the simplified model, given an initial speed and gear ratio, for any given final speed there is only one gear ratio. Therefore, implementation of Pontryagin's Maximum Principle for the simple KERS model requires an additional condition for the co-states (this is also true for Case 3 in Kirk (1998). In this case the initial states are specified and the final states on the surface $m(x(t))=0$ with final time fixed. Consequently an additional boundary condition equation must be included. The additional boundary condition is given as:

$$\frac{\partial h}{\partial \bar{x}}\left(\bar{x}\left(t_f\right)\right)-\bar{\psi}\left(t_f\right)=\sum_{i=1}^{k} d_i\left[\frac{\partial m_i}{\partial \bar{x}}\left(\bar{x}\left(t_f\right)\right)\right] \tag{4.15}$$

where d_i represents a set of unknown variables to be determined, along with the constants of integration, and h (defined in Chapter 3) represents the cost

function associated with the final state error. In equation (4.8), it can be seen that for this example $h = 0$, and since $x_1(t_f)$ is specified, the boundary condition for the co-state is as follows:

$$\psi_1(t_f) = 0 \tag{4.16}$$

The additional boundary condition for the co-state eliminates the redundancy problem that arises with the simplified model. It is important to note that this is possible only for x_1 (and its co-state) because it is function of both states, however x_2 does not depend on x_1.

The implementation is done using the values from Tables 4.1, 4.2 and equation (4.16). The initial guess values are given in Table 4.3, and the results are shown below.

Initial Guess	
x0	112
x1	35
x2	50
p0	-1
p1	0
p2	14

Table 4.3 The initial guess used with the TBVP solver (Matlab function bvp4c)

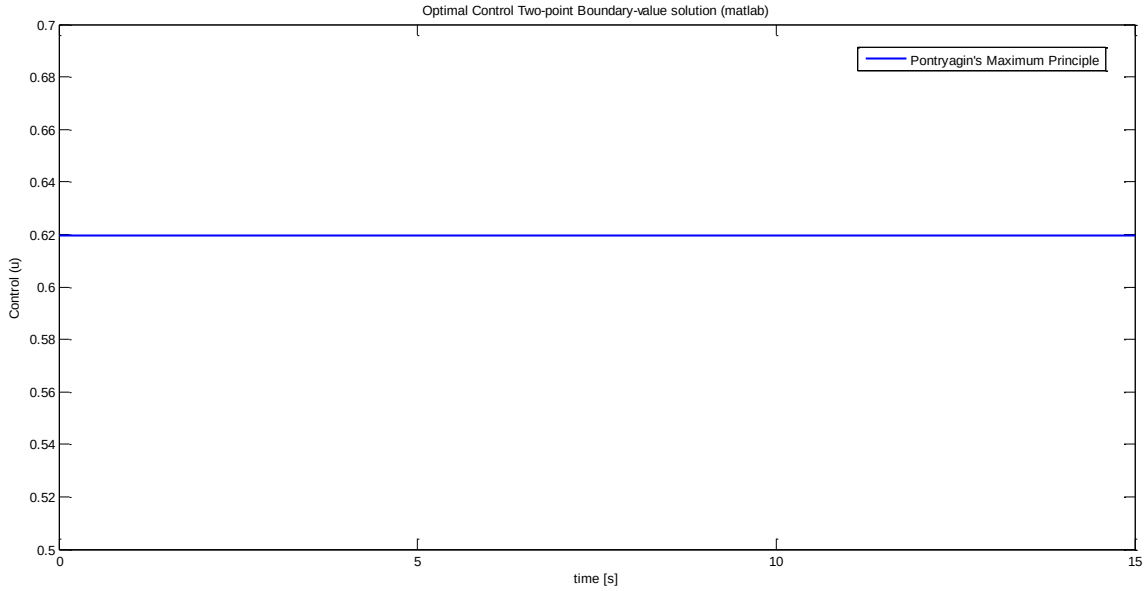

Figure 4.1 Optimal control u for the simplified KERS model using Pontryagin Maximum Principle two-point boundary-value numerical solution

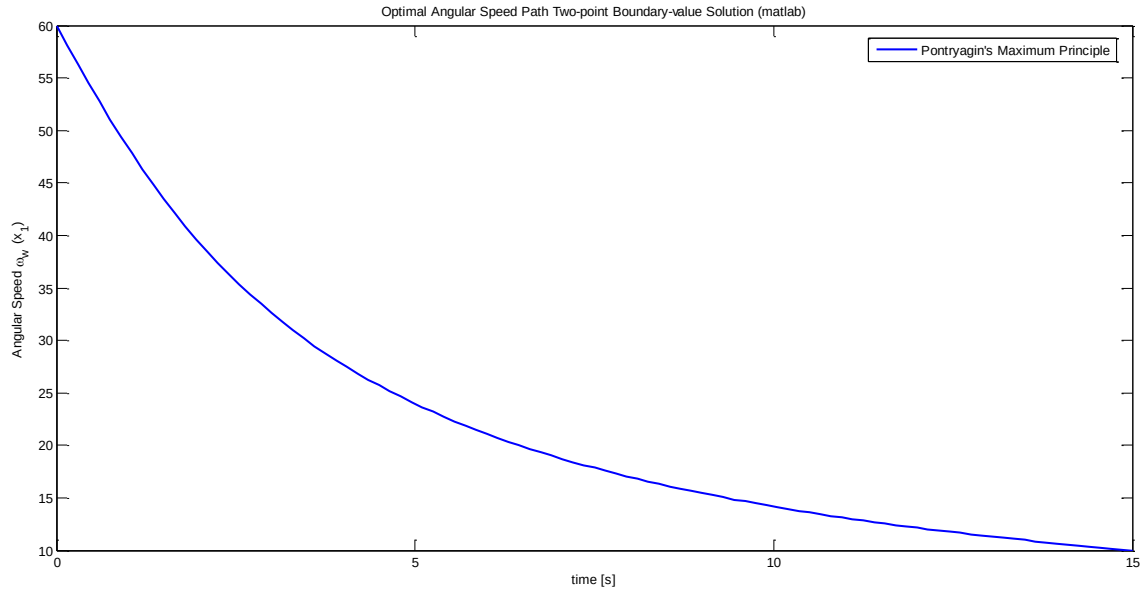

Figure 4.2 Optimal trajectory for x_1 (speed) for the simplified KERS model using Pontryagin Maximum Principle two-point boundary-value numerical solution

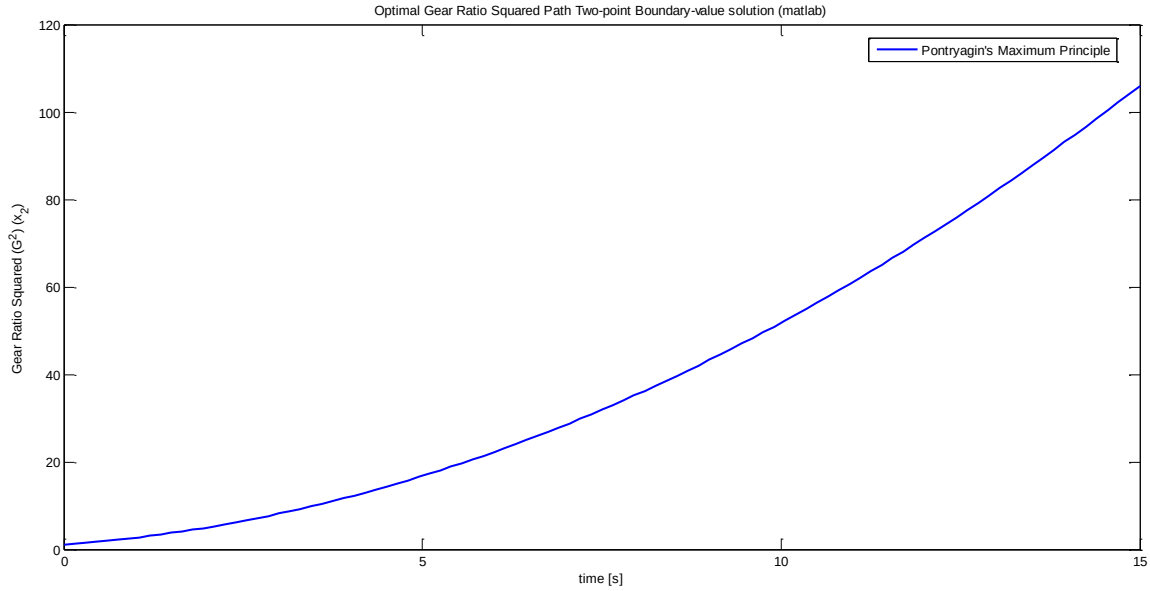

Figure 4.3 Optimal trajectory for x_2 (Gear ratio squared) for the simplified KERS model using Pontryagin Maximum Principle two-point boundary-value numerical solution

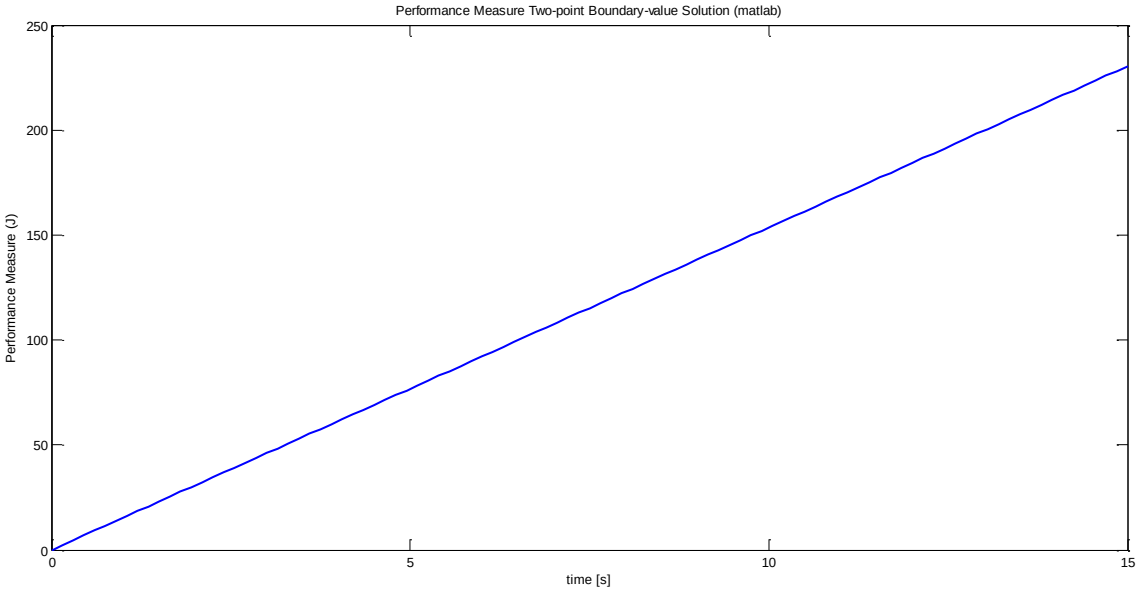

Figure 4.4 Cost evaluation of the optimal control for the simplified KERS model using Pontryagin Maximum Principle two-point boundary-value numerical solution

Figure 4.1 shows the optimal control obtained solving the TPBV. Figure 4.2 and 4.3 show the optimal path for the two state variables x_1 and x_2 respectively. Figures 4.2 and 4.3 show that the boundary conditions are met and the control found has a constant value (figure 4.1). In fact, figure 4.1 shows that a trivial

solution is obtained, indicating that the best way to move from an initial gear ratio to a final gear ratio is to keep the rate of change constant (when optimizing for the rate of change). As a first stage these results are compared and verified using the developed discrete time Dynamic Programming method.

Application of Dynamic Programming - Determination of Control Parameters

In order to verify the results obtained from Pontryagin's Maximum Principle and to compare with a different methodology, the implementation of Dynamic Programming is now considered. The improved discrete Dynamic Programming strategy developed in Chapter 3 (which showed high accuracy) is now to be implemented for the simple KERS.

The developed strategy is based on the determination of the control parameters given the initial and final states by inverting the integration process using the Runge Kutta method. Implementation for the simple KERS is achieved as follows: The states and control variable general equations are drawn from Chapter 3, equations (3.36) to (3.38), and the first set of equations for the boundary states is given as:

$$Wk_{11} = \frac{-J_f u_k x_{1k} x_{2k}^{0.5}}{J_w + J_f x_{2k}} \tag{4.17}$$

and

$$Wk_{12} = 2u_k x_{2k}^{0.5} \tag{4.18}$$

And, the second set of equations is given as:

$$Wk_{21} = \frac{-J_f\left(u_k + Wk_{1u}\frac{\Delta t}{2}\right)\left(x_{1k} + Wk_{11}\frac{\Delta t}{2}\right)\left(x_{2k} + Wk_{12}\frac{\Delta t}{2}\right)^{0.5}}{J_w + J_f\left(x_{2k} + Wk_{12}\frac{\Delta t}{2}\right)} \tag{4.19}$$

and

$$Wk_{22} = 2\left(u_k + Wk_{1u}\frac{\Delta t}{2}\right)\left(x_{2k} + Wk_{12}\frac{\Delta t}{2}\right)^{0.5} \tag{4.20}$$

The third set of equations is given as:

$$Wk_{31} = \frac{-J_f\left(u_k + Wk_{2u}\frac{\Delta t}{2}\right)\left(x_{1k} + Wk_{21}\frac{\Delta t}{2}\right)\left(x_{2k} + Wk_{22}\frac{\Delta t}{2}\right)^{0.5}}{J_w + J_f\left(x_{2k} + Wk_{22}\frac{\Delta t}{2}\right)} \tag{4.21}$$

and

$$Wk_{32} = 2\left(u_k + Wk_{2u}\frac{\Delta t}{2}\right)\left(x_{2k} + Wk_{22}\frac{\Delta t}{2}\right)^{0.5} \tag{4.22}$$

Finally, the fourth set of equations is given as:

$$Wk_{41} = \frac{-J_f\left(u_k + Wk_{3u}\Delta t\right)\left(x_{1k} + Wk_{31}\Delta t\right)\left(x_{2k} + Wk_{32}\Delta t\right)^{0.5}}{J_w + J_f\left(x_{2k} + Wk_{32}\Delta t\right)} \tag{4.23}$$

and

$$Wk_{42} = 2\left(u_k + Wk_{3u}\Delta t\right)\left(x_{2k} + Wk_{32}\Delta t\right)^{0.5} \tag{4.24}$$

The final expressions are obtained using equations (4.17), (4.24), (3.36), (3.27), and equation (3.53) for the control variable. The final expressions for the states are given as:

$$x_{1k+1} = x_{1k} + \frac{1}{6}\left[Wk_{11} + Wk_{21}^{*} + Wk_{31}^{*} + Wk_{41}^{*}\right]\Delta t \tag{4.25}$$

where Wk_{11} is given in equation (4.17), and

$$Wk_{21}^{*} = \frac{-2J_f U\left(x_{2k} + \frac{-J_f u_k x_{1k} x_{2k}^{0.5}\Delta t}{2\left(J_w + J_f x_{2k}\right)}\right)\left(x_{3k} + u_k x_{2k}^{0.5}\Delta t\right)^{0.5}}{J_w + J_f\left(x_{3k} + 2u_k x_{2k}^{0.5}\frac{\Delta t}{2}\right)} \tag{4.26}$$

and

$$Wk_{31}^{*}=\frac{-2J_fU\left(\begin{array}{l}x_{2k}+\\ -J_fU\left(\begin{array}{l}x_{2k}+\\ \dfrac{-J_fu_kx_{1k}x_{2k}^{0.5}\Delta t}{2\left(J_w+J_fx_{2k}\right)}\end{array}\right)\left(\begin{array}{l}x_{3k}+\\ u_kx_{2k}^{0.5}\Delta t\end{array}\right)^{0.5}\Delta t\\ \hline 2\left(J_w+J_f\left(x_{3k}+u_kx_{2k}^{0.5}\Delta t\right)\right)\end{array}\right)\left(\begin{array}{l}x_{3k}+\\ U\left(\begin{array}{l}x_{3k}+\\ u_kx_{2k}^{0.5}\Delta t\end{array}\right)^{0.5}\Delta t\end{array}\right)^{0.5}}{J_w+J_f\left(x_{3k}+U\left(x_{3k}+u_kx_{2k}^{0.5}\Delta t\right)^{0.5}\Delta t\right)}} \tag{4.27}$$

and Wk_{41}^{*} is grouped in three terms

$$Wk_{41}^{*}=Term1^{*}Term2^{*}Term3 \tag{4.28}$$

where the first term is

$$Term1=-J_f\left(u_k+u_g\Delta t\right) \tag{4.29}$$

and, the second term is

$$Term2=x_{2k}+\frac{-J_fU\left(\begin{array}{l}x_{2k}+\\ -J_fU\left(\begin{array}{l}x_{2k}+\\ \dfrac{-J_fu_kx_{1k}x_{2k}^{0.5}\Delta t}{2\left(J_w+J_fx_{2k}\right)}\end{array}\right)\left(\begin{array}{l}x_{3k}+\\ u_kx_{2k}^{0.5}\Delta t\end{array}\right)^{0.5}\Delta t\\ \hline 2\left(J_w+J_f\left(x_{3k}+u_kx_{2k}^{0.5}\Delta t\right)\right)\end{array}\right)\left(\begin{array}{l}x_{3k}+\\ U\left(\begin{array}{l}x_{3k}+\\ u_kx_{2k}^{0.5}\Delta t\end{array}\right)^{0.5}\Delta t\end{array}\right)^{0.5}}{J_w+J_f\left(x_{3k}+U\left(x_{3k}+u_kx_{2k}^{0.5}\Delta t\right)^{0.5}\Delta t\right)}dt \tag{4.30}$$

finally, the last term is

$$Term3=\frac{\left(x_{3k}+2U\left(x_{3k}+U\left(x_{3k}+u_kx_{2k}^{0.5}\Delta t\right)^{0.5}\Delta t\right)^{0.5}\Delta t\right)^{0.5}}{J_w+J_f\left(\begin{array}{l}x_{3k}+2U\left(\begin{array}{l}x_{2k}+2U\left(\begin{array}{l}x_{2k}+2u_kx_{2k}^{0.5}\dfrac{\Delta t}{2}\end{array}\right)^{0.5}\dfrac{\Delta t}{2}\end{array}\right)^{0.5}\Delta t\end{array}\right)} \tag{4.31}$$

Whereas for the second state the final expression is given as:

$$x_{2k+1} = x_{2k} + \frac{1}{6}\left[\begin{array}{l} 2u_k x_{2k}^{0.5} + \\ 4U\left(x_{2k} + u_k x_{2k}^{0.5}\Delta t\right)^{0.5} + \\ 4U\left(x_{2k} + 2U\left(x_{2k} + 2u_k x_{2k}^{0.5}\dfrac{\Delta t}{2}\right)^{0.5}\dfrac{\Delta t}{2}\right)^{0.5} + \\ 2\left(u_k + u_g\Delta t\right)\left(x_{2k} + 2A\left(x_{2k} + 2A\left(x_{2k} + 2u_k x_{2k}^{0.5}\dfrac{\Delta t}{2}\right)^{0.5}\dfrac{\Delta t}{2}\right)^{0.5}\Delta t\right)^{0.5} \end{array}\right]\Delta t \tag{4.32}$$

where

$$U = u_k + u_g\frac{\Delta t}{2} \tag{4.33}$$

Both discrete state equations (4.25) to (4.32) are dependent on the control variable; therefore, as for the linear oscillator, to invert the integration process it is necessary to solve both equations in order to find the control parameters.

It is appropriate to test this procedure before implementing the full improved Dynamic Programming approach. Following the steps used in Chapter 3, the solution found from Pontryagin's Maximum Principle (which are slightly manipulated in order to confirm that changes in the control variable are possible), is used as a known given path. Linear control is assumed. The system of equations is solved using the least squares method and the results are shown in figure 4.5.

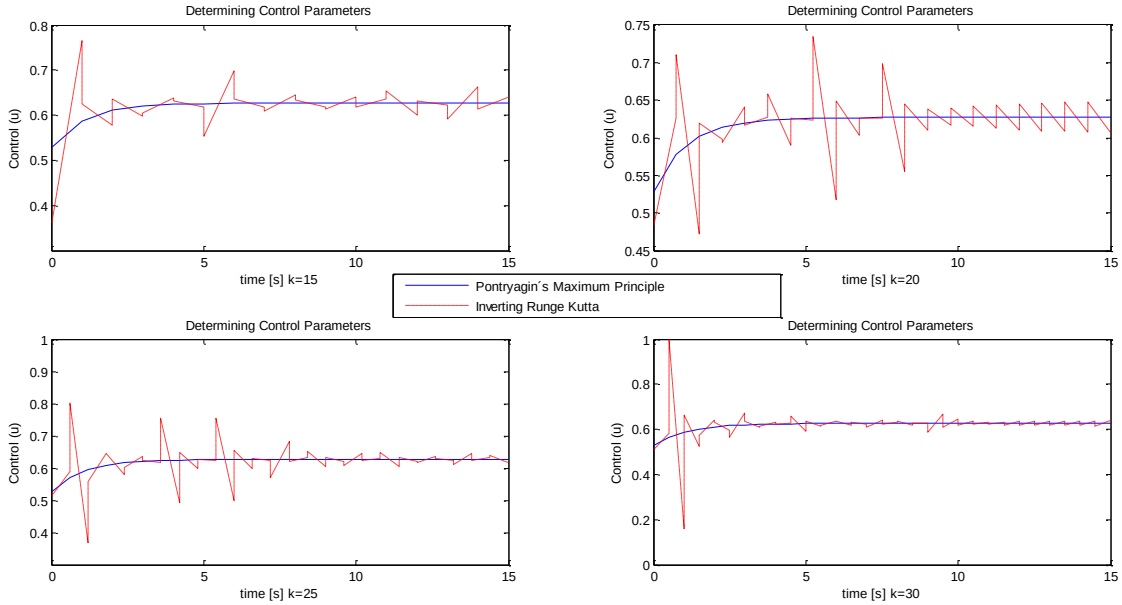

Figure 4.5 Linear controls found by inverting Runge Kutta integration method and using Pontryagin Maximum Principle (manipulated solution) states trajectories.

It can be seen from figure 4.5 that the solution becomes unstable for larger number of steps, even when the results with a discrete time step of 1s initially seemed promising. This shows that an even deeper analysis of the results is necessary in order to develop a strategy that can be implemented with the improved Dynamic Programming approach.

In order to have a closer look to the possible values that the control parameters can take, the least squares solution can be used to reconstruct the problem. The least squares solution is found by minimizing q in equation (4.34), and since the discrete state equations (f_1, f_2) are functions of the control parameters, equation (4.34) can be evaluated for different u_k and u_g values, meaning that the lowest value from equation (4.34) represents the minimum control used. The least squares equation is shown below and the evaluation of the function is shown in figures 4.6 and 4.7.

$$f_1^{\,2} + f_2^{\,2} = q \tag{4.34}$$

where q is the objective function to be minimized.

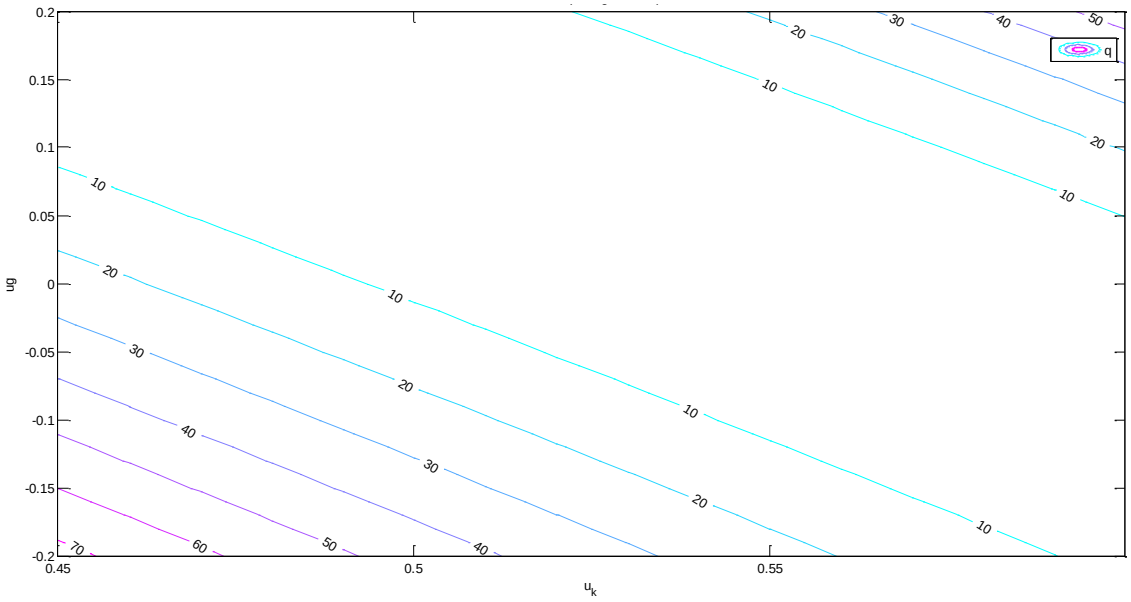

Figure 4.6 Evaluation of q for various combinations of u_k, ug

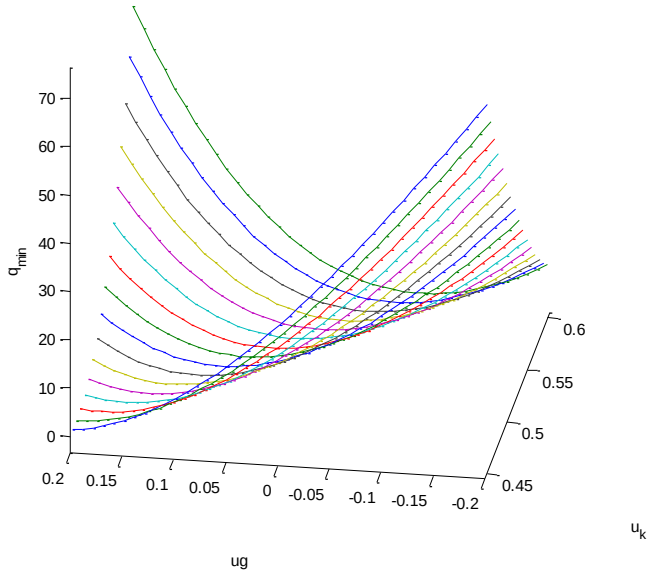

Figure 4.7 Evaluation of q for various combinations of u_k, u_g

Figure 4.6 shows the values of the function q where it can be seen that for various values of u_k, u_g the evaluated function takes the same value. Figure 4.7 shows a three-dimensional graph with the evaluation of q in terms of u_k, u_g.

Figures 4.6 and 4.7 show that when using the least squares method various combinations of control parameters minimise the function q, it can be clearly seen that this is not a single point but a valley which represents the minimum of q. This suggests that an additional condition is needed in order to find a solution.

Since the manipulated Pontryagin's Maximum Principle solution is available, in order to explore the possibility of an improvement, an additional condition must be considered. To create this additional condition the cost function f_0 can be included in the least squares minimization procedure. The results are shown in figure 4.8.

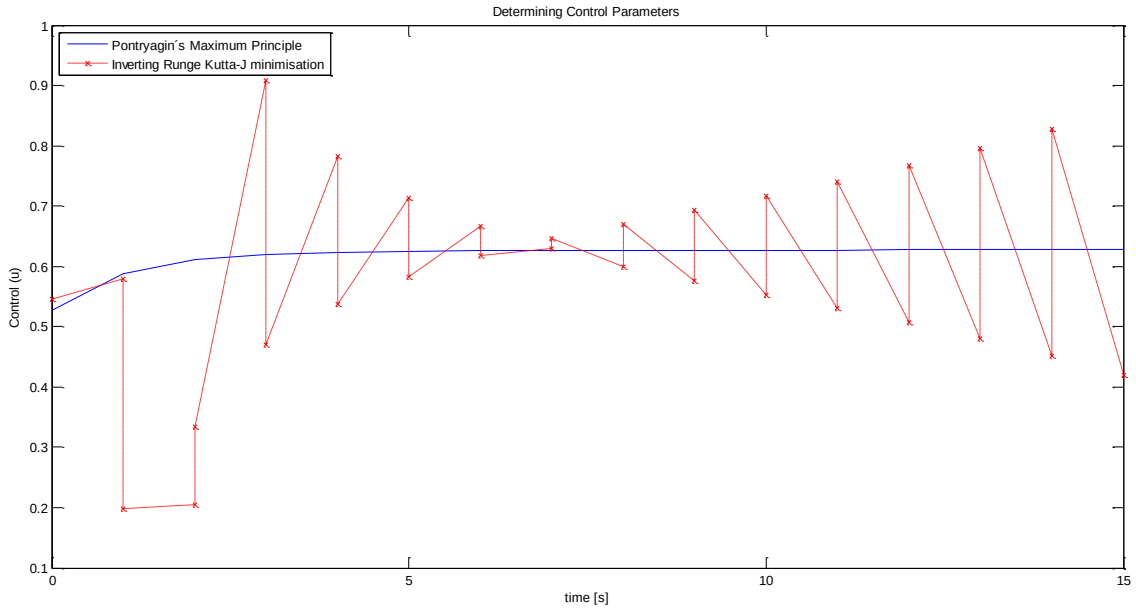

Figure 4.8 Linear control found by inverting Runge Kutta integration method, using Pontryagin Maximum Principle (manipulated solution) states trajectories and minimising cost function.

In figure 4.8 it can be seen that the solution is unstable oscillating around the expected value. The first step is close to the modified Pontryagin solution, but from the second step an error is introduced and carried forward.

These results show that including the cost function as part of the minimisation criteria for the least squares method, does not improve the response. Therefore,

a different approach is needed. In order to provide more information to the system the introduction of the cost function is considered now in explicit form for minimising the control parameters. The explicit form of J assuming linear control is given by:

$$J = \int_0^t \left(u_k + u_g t\right)^2 dt \tag{4.35}$$

The minimisation of J with respect to the control parameters, involves solving:

$$\frac{\partial J}{\partial u_k} = u_g t^2 + 2u_k t \tag{4.36}$$

and

$$\frac{\partial J}{\partial u_g} = \frac{2}{3} u_g t^3 + u_k t^2 \tag{4.37}$$

Equations (4.36) and (4.37) are now considered for the least squares solution in combination with the discrete state equations (f_1, f_2) with the objective of eliminating the instability. The results are shown in figure 4.9.

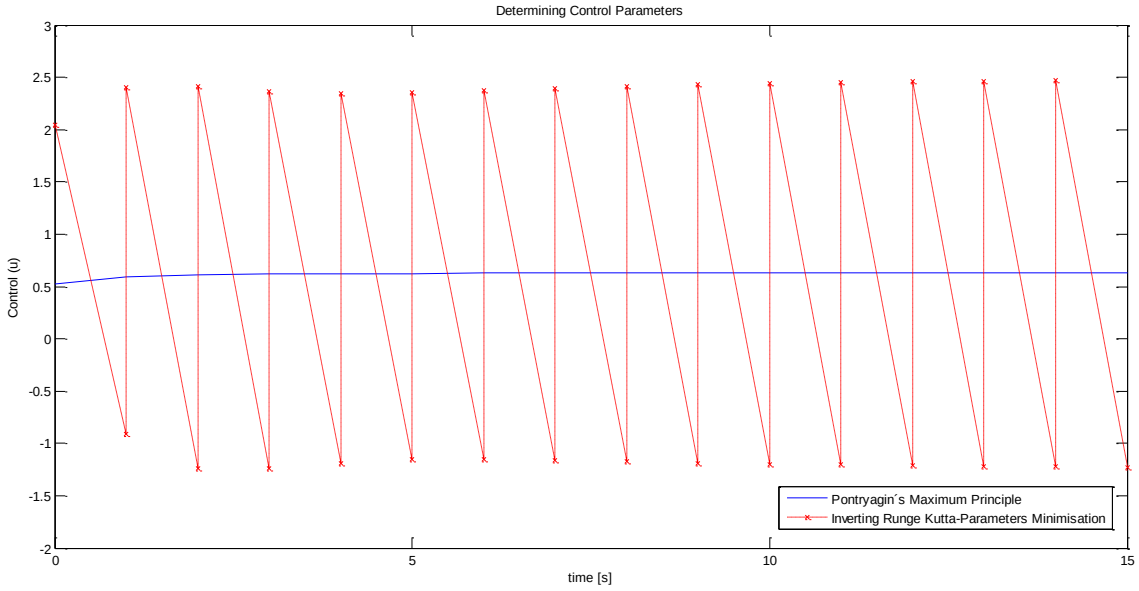

Figure 4.9 Linear control found by inverting the Runge Kutta integration method, using Pontryagin Maximum Principle (manipulated solution) states trajectories and minimising the control parameters.

Figure 4.9 shows that the system is unstable and the parameters found for the control make it oscillate around the expected value.

Since the results shown before are not within a desirable range compared with the expected solution (and indeed the solution behaviour becomes unstable even after giving the additional information), it is appropriate to even further explore if the approach to this problem is adequately posed to start with. This is done by assuming that the final cost at each time step is known. The results of doing this are shown in figure 4.10.

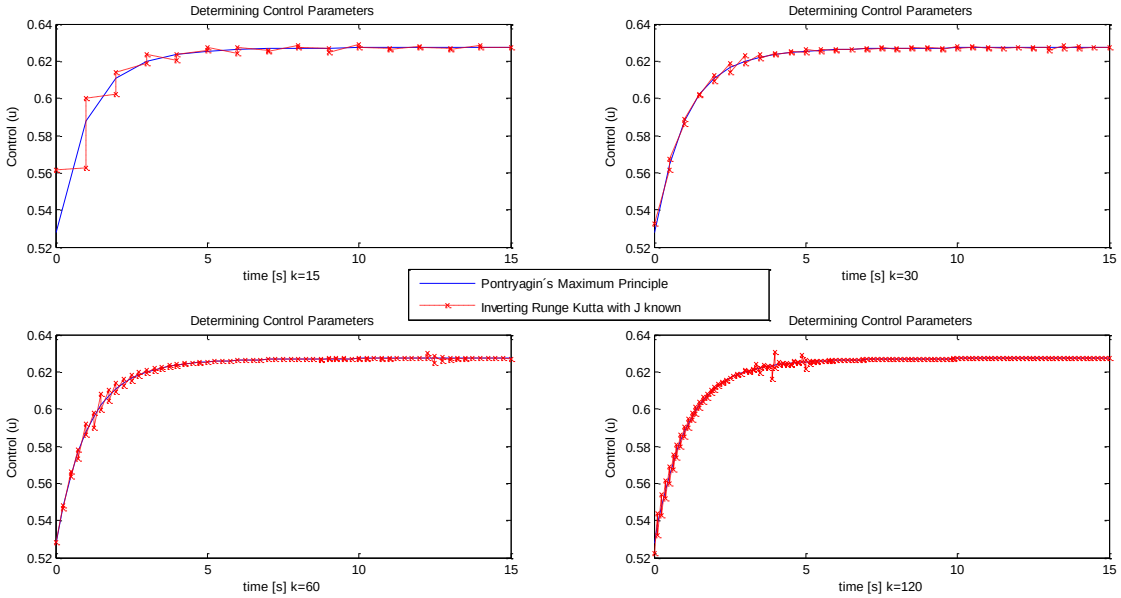

Figure 4.10 Linear controls found by inverting Runge Kutta integration method and using Pontryagin Maximum Principle (manipulated solution) states trajectories and cost function (J known).

Figure 4.10 contains the results assuming the final cost for each step is known. It can be seen that the method responds very well when the final cost is included, however given that the cost is actually not known when implementing Dynamic Programming, a different approach should be explored.

The results shown from figures 4.5 to 4.9 suggest that the approach to the problem is indeed not adequate. From the results in figures 4.6 and 4.7 it is clear that there are various solutions that minimise the function used in the least squares method, however, when enough information to reconstruct the

manipulated from Pontryagin's Maximum Principle is provided, the solution is followed accurately as shown in figure 4.10.

With the objective of exploring further the strategy used, where the cost function is included into the least square approach, it is now decided to try using a quadratic control; the reason for this is to avoid (as it was shown in Chapter 3) the use of an oversubscribed system. The equation for the control is given as:

$$u_{k+1} = u_k + u_g \Delta t + u_{g2} \Delta t^2 \tag{4.38}$$

where u_g and u_{g2} are the coefficients associated with the linear and quadratic factors respectively, and u_k is the initial value for the control variable.

The results of determining the control parameters assuming quadratic control are shown in figure 4.11.

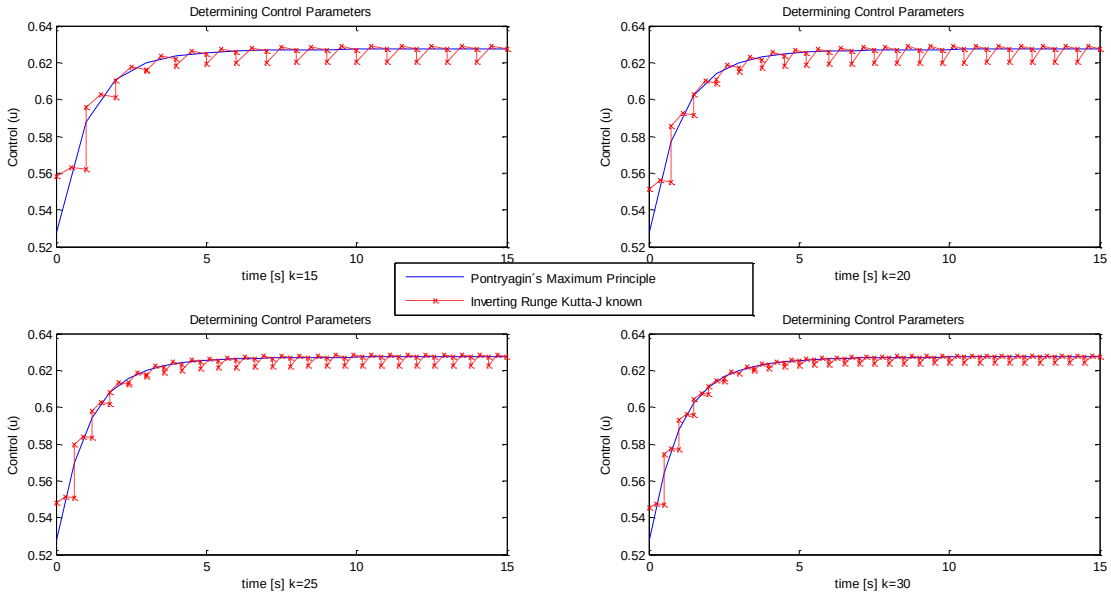

Figure 4.11 Quadratic controls found by inverting Runge Kutta integration method and using Pontryagin Maximum Principle (manipulated solution) states trajectories and cost function (J known).

The results in figure 4.11 show that given the cost function as part of the least squares approach it is possible to find a piece-wise quadratic control that fits the Pontryagin's Maximum Principle modified solution.

The results shown so far suggest that using the Runge Kutta integration method although very suitable for the linear oscillator control in Chapter 3 is actually not suitable for the KERS application within the inverse integration strategy. The nonlinear characteristics of the system model, and the recursive characteristics of the Runge Kutta method make the implementation complicated. This combination increases the number of roots (as can be seen in equations (4.25) and (4.26)) when trying to solve for the control variable. Therefore, even with the help of advanced computational tools, to obtain the desired solution it might not be possible or is too difficult to do this.

In fact, the recursive character of the higher order numerical approximation method (Runge Kutta) is found to be a key factor in the increase in complexity in the system of equations. Therefore, it is appropriate to consider a lower order numerical approximation method. As it was earlier shown (Chapter 3), the Euler integration method was not suitable for the desired oscillator application. However consideration of the use of the Modified Euler method, is appropriate to address the problems encountered. This first order method evaluates the system at a middle integration point; therefore, the desired character of time dependent control is suitable, and the recursive iteration is avoided. The time integration of a second order dynamic system using Modified Euler method is given (from Appendix A) as:

$$x_{1k+1} = x_{1k} + f_{1k}\left(t_k + \frac{\Delta t}{2}, x_{1k} + \frac{f_{1k}\Delta t}{2}\right)\Delta t \tag{4.39}$$

and

$$x_{2k+1} = x_{2k} + f_{2k}\left(t_k + \frac{\Delta t}{2}, x_{2k} + \frac{f_{2k}\Delta t}{2}\right)\Delta t \tag{4.40}$$

where the function f_{ik} denotes the finite difference for each state respectively. For the model represented by equations (4.6) and (4.7), assuming linear control, the integration equations become:

$$x_{1k+1} = x_{1k} + \left(\frac{-J_f\left(u_k + u_g \frac{\Delta t}{2}\right)\left(x_{1k} + \frac{-J_f u_k x_{1k} x_{2k}^{0.5}}{J_w + J_f x_{2k}} \frac{\Delta t}{2}\right)\left(x_{2k} + 2u_k x_{2k}^{0.5} \frac{\Delta t}{2}\right)^{0.5}}{J_w + J_f\left(x_{2k} + 2u_k x_{2k}^{0.5} \frac{\Delta t}{2}\right)} \right) \Delta t \tag{4.41}$$

and

$$x_{2k+1} = x_{2k} + \left(2\left(u_k + u_g \frac{\Delta t}{2}\right)\left(x_{2k} + 2u_k x_{2k}^{0.5} \frac{\Delta t}{2}\right)^{0.5} \right) \Delta t \tag{4.42}$$

Following the same procedures as adopted earlier, the system of equations is implemented in order to find the control parameters given by the Pontryagin's Maximum Principle manipulated solution. This case considers only equations (4.41) and (4.42) in the least squares method. The results are presented in figure 4.12.

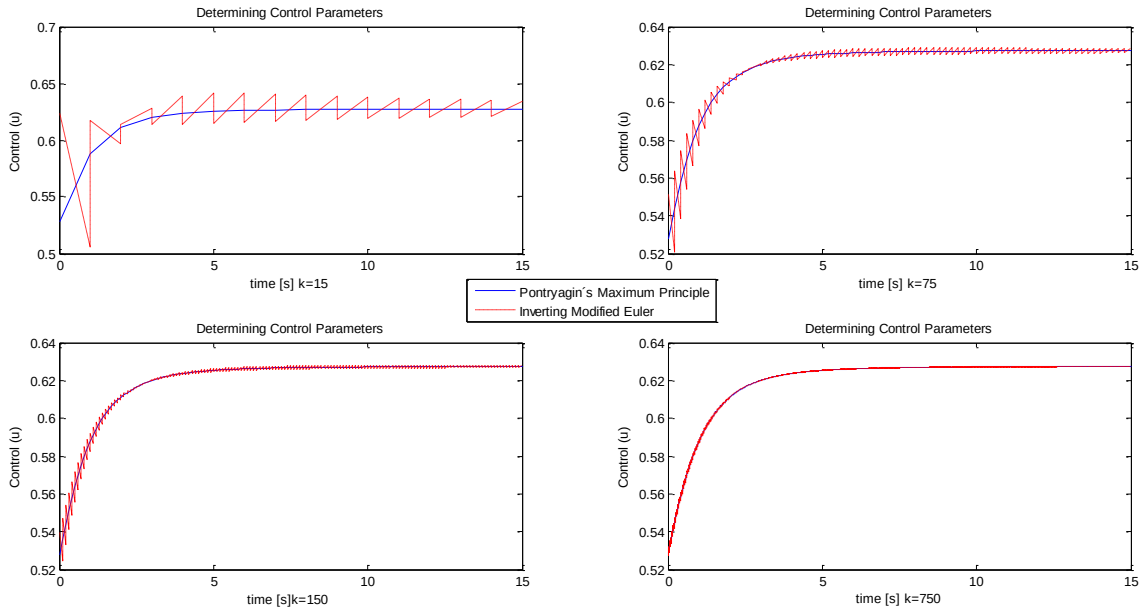

Figure 4.12 Linear controls found by inverting Modified Euler integration method and using Pontryagin Maximum Principle (manipulated solution) states trajectories.

The results in figure 4.12 show that for smaller time steps, the Modified Euler method converges and is able to follow the given trajectory from Pontryagin's

Maximum Principle manipulated solution. The solution obtained is oscillatory but the oscillations become very small for larger number of steps.

The convergence of the Modified Euler method suggests that it is adequate to be implemented in the improved version of Dynamic Programming, even for small number of time steps, the solution shows an accurate response which is highly desirable for Dynamic Programming.

Improved Discrete Dynamic Programming with Linear Piece-wise Control

Given that using the Modified Euler method to determine the control parameters is an accurate strategy which converges when increasing the number of steps, the full implementation for Optimal Control purposes using the improved discrete Dynamic Programming strategy developed earlier (Chapter 3) is now considered.

The values shown in Tables 4.1 and 4.2 are the systems parameters and boundary conditions respectively, and the values shown in Table 4.4 are the input parameters for Dynamic Programming, which are presented in vector form.

Admissible Values for Dynamic Programming (case 1)	
x1 (speed) [rad/s]	[10:0.25:60]
x2 (Gear ratio squared)	[1:0.75:106]
u (control)	[0.1:0.05:1]
k (number of steps)	15

Table 4.4 The admissible values used in Dynamic Programming

The results and their comparison with Pontryagin's Maximum Principle are shown in figures 4.13 to 4.16. In order to validate that the control found corresponds to the states chosen in the Dynamic Programming process, an integration of the control is made and also displayed with the results.

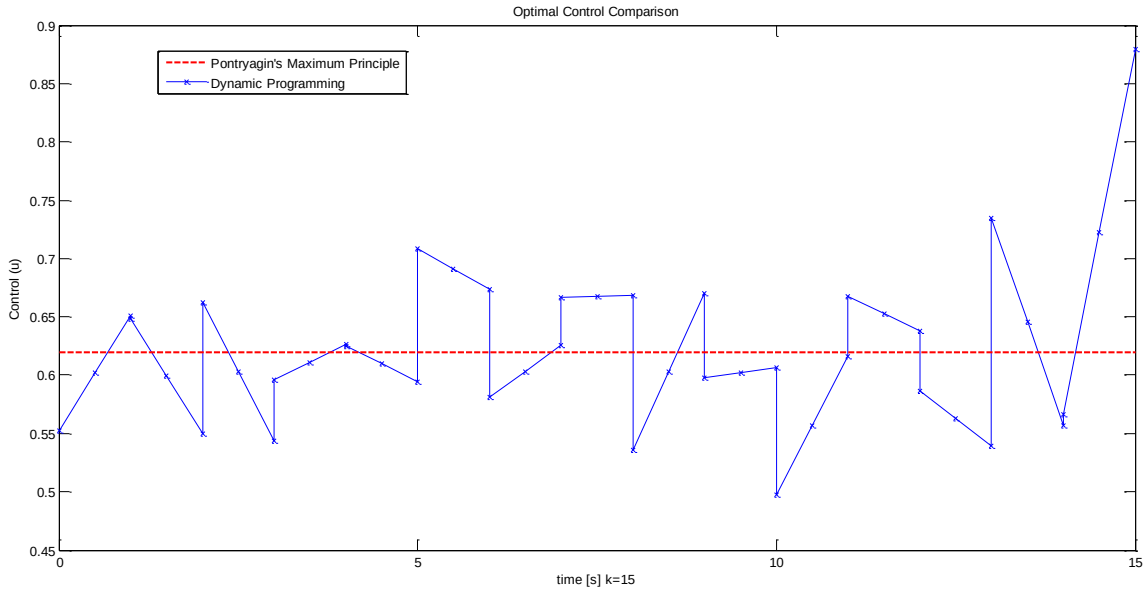

Figure 4.13 Optimal control comparison between Dynamic Programming and Pontryagin Maximum Principle for the simplified KERS.

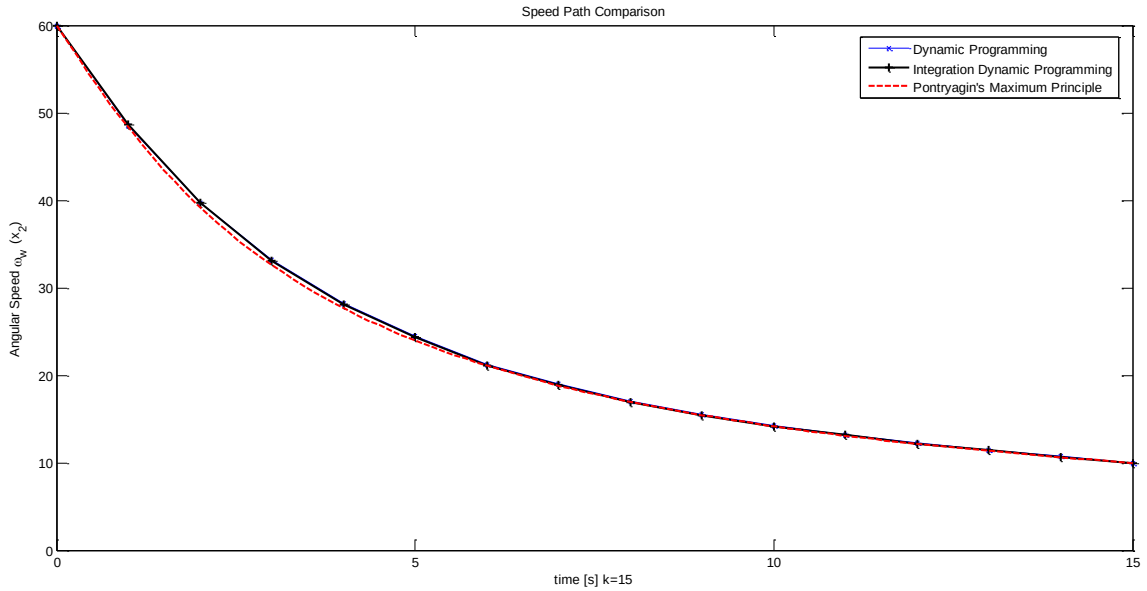

Figure 4.14 Optimal speed trajectories comparison between Dynamic Programming and Pontryagin Maximum Principle for the simplified KERS.

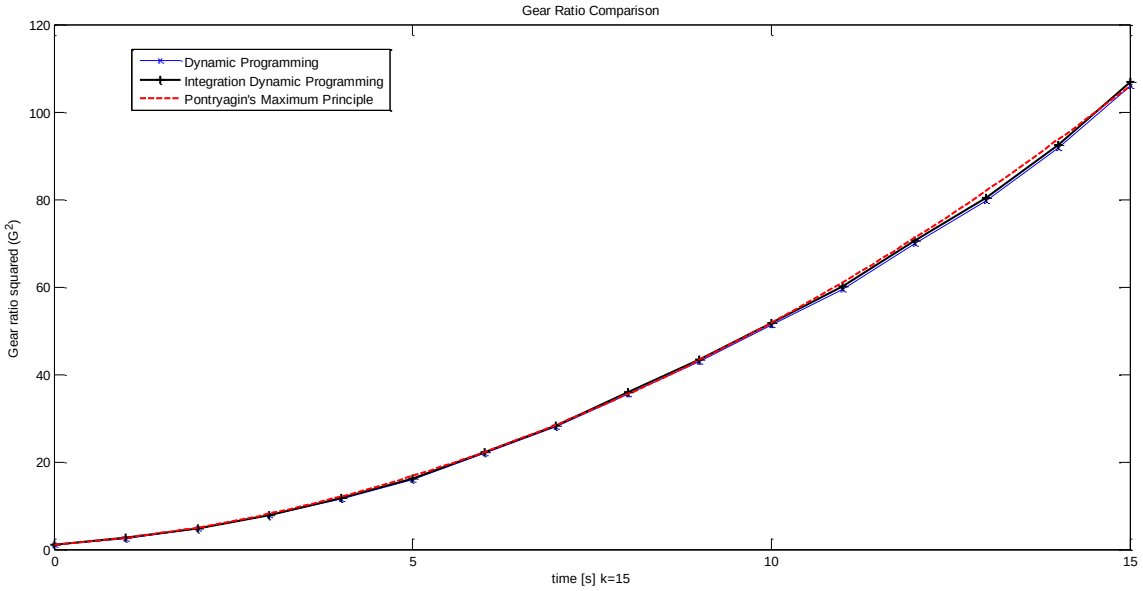

Figure 4.15 Optimal Gear ratio squared trajectories comparison between Dynamic Programming and Pontryagin Maximum Principle for the simplified KERS.

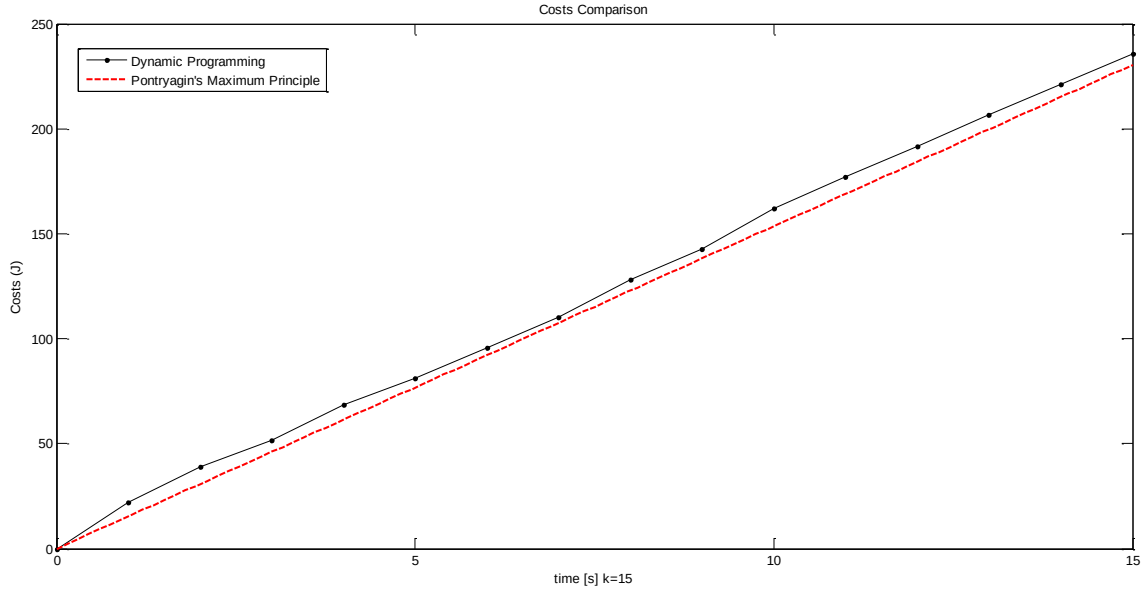

Figure 4.16 Performance (J) comparison between Dynamic Programming and Pontryagin Maximum Principle for the simplified KERS.

In figure 4.13 the optimal control comparison is shown, the solution obtained with Dynamic Programming does not match the one from Pontryagin's Maximum Principle, however it stays close to it. Figures 4.14 and 4.15 show that the trajectories followed by the system, by applying the control shown in figure 4.13 follow the solution from Pontryagin's Maximum Principle very

closely. Figure 4.16 shows the performance index comparison between both the methods, where the final values are found to be very close.

Although the control found in figure 4.13 using Dynamic Programming does not seem very accurate, in figures 4.15 and 4.16 can be seen that the trajectory followed is actually good. Considering that the time steps are 1s and that the admissible discrete states are limited, the solution found is very accurate. It can be seen that the first step (where Dynamic Programming solution is found backwards) is the one with least accuracy, and after that, the system compensates.

The improved version of Dynamic Programming using the Modified Euler method gives very promising results; further tests are now performed with a more realistic model.

4.2 Application of Optimal Control to more realistic flywheel-based KERS with a CVT – Including friction from ball bearings

The simplified model used earlier (equations (4.5) to (4.7)) is not a very good model for optimal control applications, since it has been shown that a “trivial” solution is found. Therefore, the more realistic model (outlined in Chapter 2) is now considered. The friction features in the model are functions of the angular speeds (flywheel and traction wheel), so a non-trivial solution is now expected.

As was realised with the simplified model, since there is no interest in the angular displacement of the system, a state reduction is applied. The system given by equations (2.11), (2.12), (2.14), (2.15), (2.17) and (2.31), can be expressed in state space form as follows:

$$x_1(t) = \omega_w(t) \tag{4.43}$$

and

$$f_1(t) = \dot{x}_1(t) = \dot{\omega}_w(t) = \frac{-J_f u(t) x_1(t) x_2{}^{0.5}(t) - K_{vf} x_1(t) x_2(t) - K_{vw} x_1(t)}{J_w + J_f x_2(t) + m_v r_w{}^2} \tag{4.44}$$

and

$$f_2(t) = \dot{x}_2(t) = 2u(t)x_2^{0.5}(t) \tag{4.45}$$

Where K_v represents the friction coefficient due to viscosity for the bearings associated with the flywheel and the traction wheel, m_v is the mass of the vehicle and r_w is the traction wheel radius.

In order to approach an energy optimization solution, it is important to specify that the objective is to maximize the energy stored in the flywheel. Since the kinetic energy of the flywheel is given as:

$$E = \frac{J_f \omega_f^{\,2}}{2} \tag{4.46}$$

where J_f is the moment of inertia and ω_f is the angular velocity of the flywheel. For the control problem described, the angular velocity of the flywheel is a function of the wheels speed (equation (2.7); this relationship is given as:

$$\omega_f = G\omega_w \tag{4.47}$$

where G is the gear ratio. In this case, the final speed of the wheels is specified $\omega_w(t_f) = known$, for this reason the energy at the flywheel is dependent on the final gear ratio $G(t_f)$. And, since the second state $x_2(t)$ is function of the gear ratio (equation (4.7)), the energy optimization problem can be achieved by minimizing the difference respect to a desired final value of $x_2(t)$.

The final value of the gear ratio is unspecified (since $x_2(t_f)$ is unknown) so that the final gear ratio difference optimization can be included in the performance index (with this maximizing the energy at the flywheel). The cost function is given as:

$$J(t) = \beta\left(x_2(t_f) - x_{2d}\right)^2 + \int_0^{t_f} \alpha u(t)^2 dt \tag{4.48}$$

where, x_{2d} is the desired final gear ratio squared, and β is a weighting factor.

The optimal control objective is to drive the system from a given initial state $\bar{x}(t_0)$, to a final specified rotational speed for the vehicle traction wheels $x_1(t_f)$, by minimizing the cost function given by (4.48)

Application of Pontryagin Maximum Principle

Turning to the use of Pontryagin's Maximum Principle, the formulation of the problem is as follows: The Hamiltonian for the KERS model and friction, based on the ball bearing models, is given as:

$$\begin{aligned} H(t) = {} & \left(4\beta u(t) x_2^{0.5}(t)\left(x_2(t) - x_{2d}\right) + \alpha u^2(t)\right)\psi_0(t) \\ & + \frac{-J_f u(t) x_1(t) x_2^{0.5}(t) - K_{vf} x_1(t) x_2(t) - K_{vw} x_1(t)}{J_w + J_f x_2(t) + m_v r_w^{\,2}}\psi_1(t) + 2u(t) x_2^{0.5}(t)\psi_2(t) \end{aligned} \tag{4.49}$$

And, from equation (4.49) the co-state functions are given as:

$$\dot{\psi}_0(t) = 0 \tag{4.50}$$

and

$$\dot{\psi}_1(t) = \frac{J_f u(t) x_2^{0.5}(t) + K_{vf} x_2(t) + K_{vw}}{J_w + J_f x_2(t) + m_v r_w^{\,2}}\psi_1(t) \tag{4.51}$$

and

$$\dot{\psi}_2(t) = -\left(6\beta u(t) x_2{}^{0.5}(t) - 2\beta x_{2d} u(t) x_2{}^{-0.5}(t)\right)\psi_0(t) + \frac{\left(0.5 J_f u(t) x_1(t) x_2^{-0.5}(t) + K_{vf} x_1(t)\right)\left(J_w + m_v r_w{}^2\right) - 0.5 J_f^2 u(t) x_1(t) x_2^{0.5}(t) - K_{vw} J_f x_1(t)}{\left(J_w + J_f x_2(t) + m_v r_w{}^2\right)^2}\psi_1(t) - u(t) x_2^{-0.5}(t)\psi_2(t) \tag{4.52}$$

The condition for optimality is now given as:

$$\frac{\partial H(t)}{\partial u(t)} = \left(4\beta x_2{}^{1.5}(t) - 4\beta x_{2d} x_2{}^{0.5} + 2\alpha u(t)\right)\psi_0(t) - \frac{J_f x_1(t) x_2^{0.5}(t)}{J_w + J_f x_2(t) + m_v r_w{}^2}\psi_1(t) + 2x_2^{0.5}(t)\psi_2(t) \tag{4.53}$$

And, from equation (4.53) the optimal control is given as:

$$u(t) = \frac{J_f x_1(t) x_2^{0.5}(t)}{J_w + J_f x_2(t) + m_v r_w{}^2}\frac{\psi_1(t)}{2\alpha\psi_0(t)} - \frac{x_2^{0.5}(t)}{\alpha}\frac{\psi_2(t)}{\psi_0(t)} - \frac{2\beta x_2{}^{0.5}(t)\left(x_2(t) - x_{2d}\right)}{\alpha} \tag{4.54}$$

Assuming the final condition for the gear ratio is not specified, additional information is required to complete the boundary conditions. Using equation (4.15) the additional relationship is given as:

$$\psi_2(t_f) = \beta\left(x_2(t_f) - x_{2d}\right) \tag{4.55}$$

The formulation of the Optimal Control problem using Pontryagin's Maximum Principle for the friction KERS is now complete. This problem has to be solved using a two-point boundary-value numerical method.

Optimal Control for simplified KERS via Pontryagin's Maximum Principle and two-point boundary-value problem numerical solution

An example for the friction model is now considered, the systems parameters are given in Table 4.5 and the boundary conditions are specified in Table 4.6. As mentioned earlier, the final gear ratio is not specified (NS).

Friction KERS	
Parameters	
α	20
β	1
Jw [kg*m^2*rad^-2]	2
Jf [kg*m^2*rad^-2]	1
mv [kg]	600
rw [m]	0.33
Kvf	1
Kvw	5

Table 4.5 Friction KERS model parameters

Boundary Conditions		
	to	tf
time [s]	0	2
x1 (t) [rad/s]	60	40
x2 (t) (gear ratio squared)	0.16	NS

Table 4.6 Boundary conditions used in the optimal control of the KERS with friction

Since $x_2(t_f)$ is not specified, a target is set to a numerical value of 30. This is done since although the final gear ratio is not restricted, the final difference from the desired point should be minimized. The results are shown in figures 4.17, 4.18 and 4.19.

Figure 4.18 shows that the boundary conditions for the angular speed are met. The optimal control obtained (figure 4.17) shows that it has a high initial value and is decreasing with time, and figure 4.19 shows the gear ratio squared resulting from such a change. However the final value for the gear ratio squared does not reach the desired value.

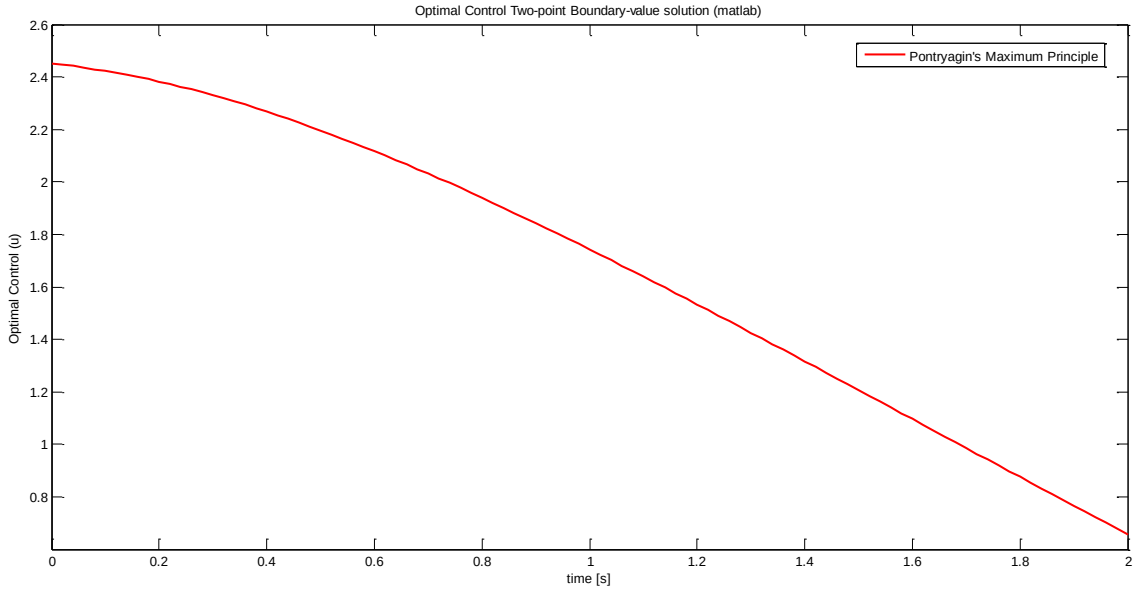

Figure 4.17 Optimal control u for the friction KERS using Pontryagin Maximum Principle and a two-point boundary-value numerical solution

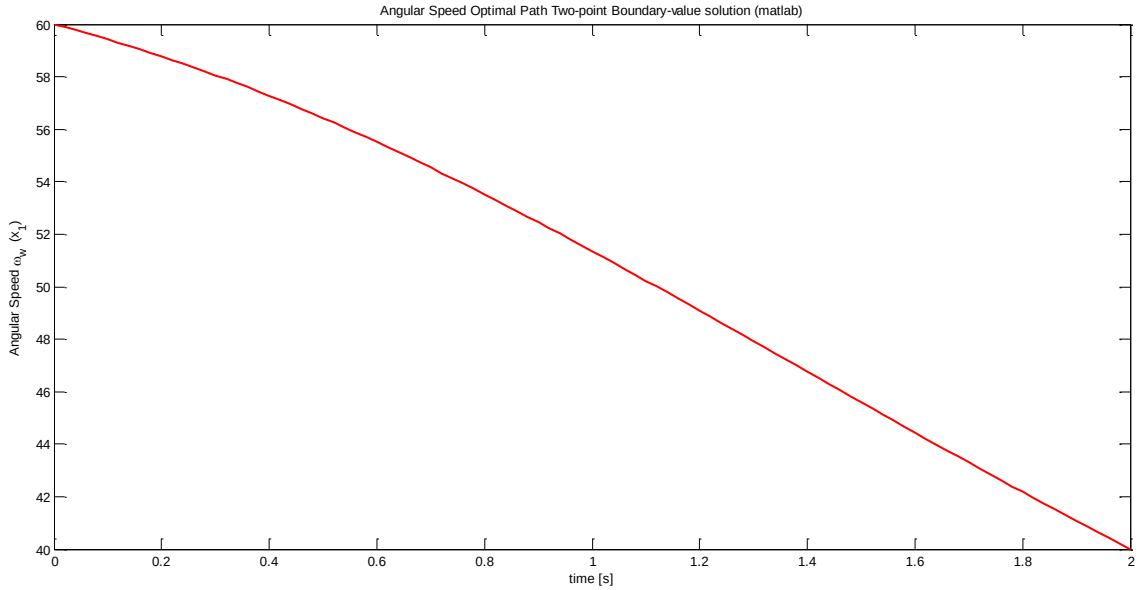

Figure 4.18 Optimal trajectory x_1 (speed) for the friction KERS using Pontryagin Maximum Principle and a two-point boundary-value numerical solution

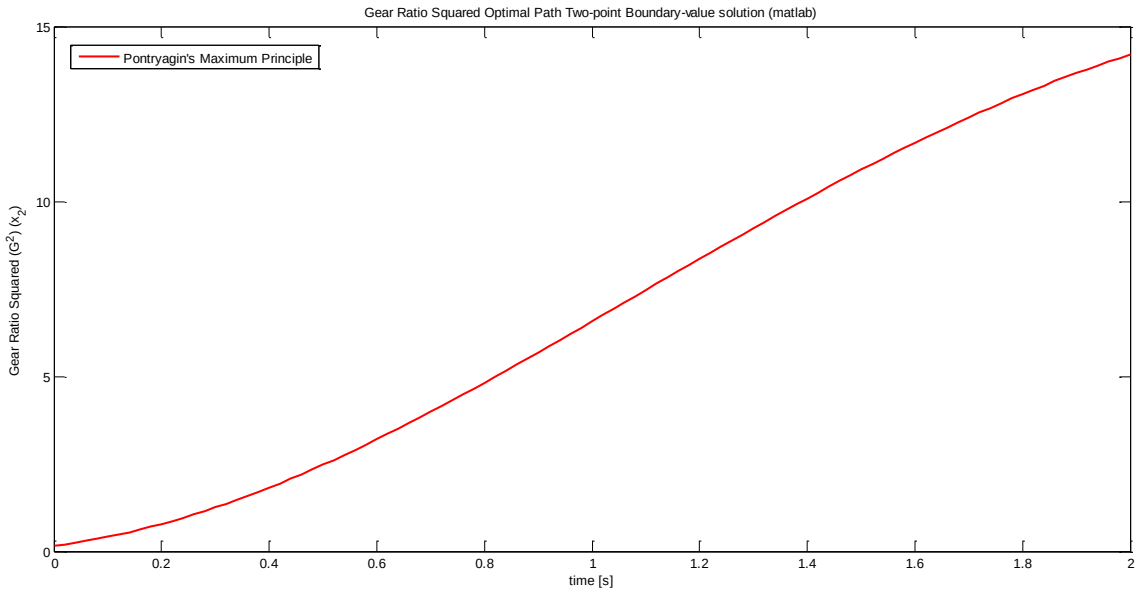

Figure 4.19 Optimal trajectory in x_2 (Gear ratio squared) for the friction KERS using Pontryagin Maximum Principle and a two-point boundary-value numerical solution

In this example, there are two different ways to reduce the energy of the system: i) through friction losses at the flywheel, and ii) through friction losses in the traction wheel bearing. Additionally, energy is transferred to the flywheel from the traction wheel; being these three the only sources to reduce the wheel speed. Given the system parameters, the optimal control solution shows that in order to meet the final conditions, the friction losses have to be modified; therefore, the trivial solution found in the simplified model does not arise anymore. The energy stored is maximized with the requirement to meet the final condition, given the friction losses this means that the desired final value for the gear-ratio-squared may not be reached, and, as it is seen in Figure 4.19, it is indeed not reached. After various tests, where the desired final value was modified, the same results are obtained; therefore the conclusion is that there is only one optimal solution for different desired conditions. This makes sense since given a final time there is a minimum amount of energy that will be lost due to friction and for this reason the reminder to meet the final conditions has to be transferred to the flywheel. An additional and more representative approach, where the energy to be stored can be controlled, is shown after a comparison and validation of the Pontryagin's Principle results with the improved modified Dynamic Programming Strategy is done.

Application of Dynamic Programming - Improved Discrete Dynamic Programming with Linear Piece-wise Control

The full implementation of the improved discrete Dynamic Programming strategy developed is now considered. Given the results for the simplified model, the best option is to use the Modified Euler method to determine the control parameters. Therefore, using equations (4.38) and (4.39) and the model represented by equations (4.43) and (4.44), the Modified Euler equations are given as:

$$x_{1k+1} = x_{1k} + \frac{\left(\begin{array}{l} -J_f\left(u_k + u_g\frac{\Delta t}{2}\right)\left(x_{1k} + \frac{-J_f u_k x_{1k} x_{2k}^{0.5} - K_{vf} x_{1k} x_{2k} - K_{vw} x_{1k}}{J_w + J_f x_{2k} + m_v r_w{}^2}\frac{\Delta t}{2}\right)\left(x_{2k} + 2u_k x_{2k}^{0.5}\frac{\Delta t}{2}\right)^{0.5} \\ -K_{vf}\left(x_{1k} + \frac{-J_f u_k x_{1k} x_{2k}^{0.5} - K_{vf} x_{1k} x_{2k} - K_{vw} x_{1k}}{J_w + J_f x_{2k} + m_v r_w{}^2}\frac{\Delta t}{2}\right)\left(x_{2k} + 2u_k x_{2k}^{0.5}\frac{\Delta t}{2}\right) \\ -K_{vw}\left(x_{1k} + \frac{-J_f u_k x_{1k} x_{2k}^{0.5} - K_{vf} x_{1k} x_{2k} - K_{vw} x_{1k}}{J_w + J_f x_{2k} + m_v r_w{}^2}\frac{\Delta t}{2}\right) \end{array} \right)\Delta t}{J_w + J_f\left(x_{2k} + 2u_k x_{2k}^{0.5}\frac{\Delta t}{2}\right) + m_v r_w{}^2} \tag{4.56}$$

and

$$x_{2k+1} = x_{2k} + \left(2\left(u_k + u_g\frac{\Delta t}{2}\right)\left(x_{2k} + 2u_k x_{2k}^{0.5}\frac{\Delta t}{2}\right)^{0.5}\right)\Delta t \tag{4.57}$$

And the control variable is obtained from equation (4.33).

The values shown in Table 4.5 are the physical characteristics of the system, and the ones in Table 4.6 indicate the boundary conditions to be met. The values presented in Table 4.7 are the admissible values for Dynamic Programming.

Admissible Values for Dynamic Programming (case 2)	
x1 (speed) [rad/s]	[40:0.25:60]
x2 (Gear ratio squared)	[0.16:0.1404:14.2]
u (control)	[0.5:0.1:2.5]
k (number of steps)	10

Table 4.7 Admissible values used in Dynamic Programming

The results and comparison with Pontryagin's Maximum Principle solution are shown in figures 4.20 to 4.23. The integration of the solution obtained with Dynamic Programming, as a form of validation, are included in the results.

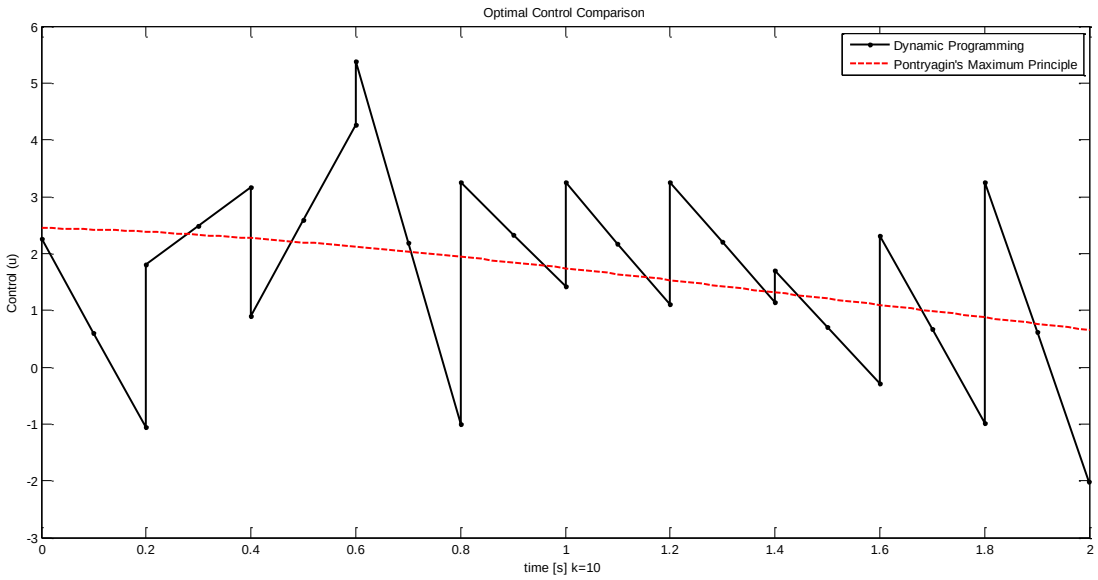

Figure 4.20 Optimal control comparison between Dynamic Programming and Pontryagin Maximum Principle for the friction KERS.

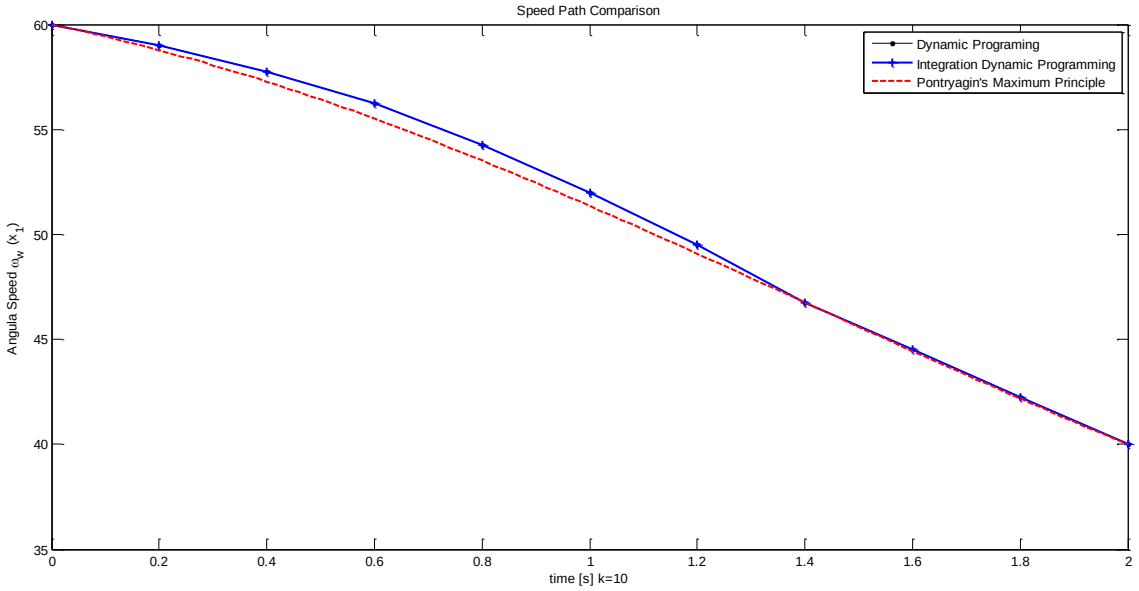

Figure 4.21 Optimal speed trajectories comparison between Dynamic Programming and Pontryagin Maximum Principle for the friction KERS.

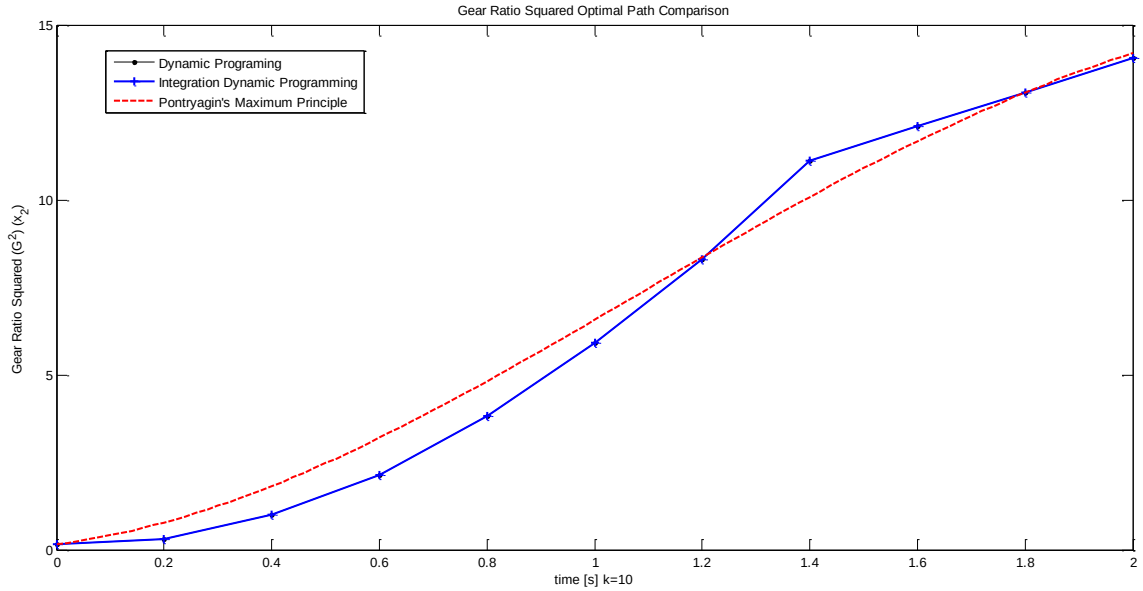

Figure 4.22 Optimal Gear ratio squared trajectories comparison between Dynamic Programming and Pontryagin Maximum Principle for the friction KERS.

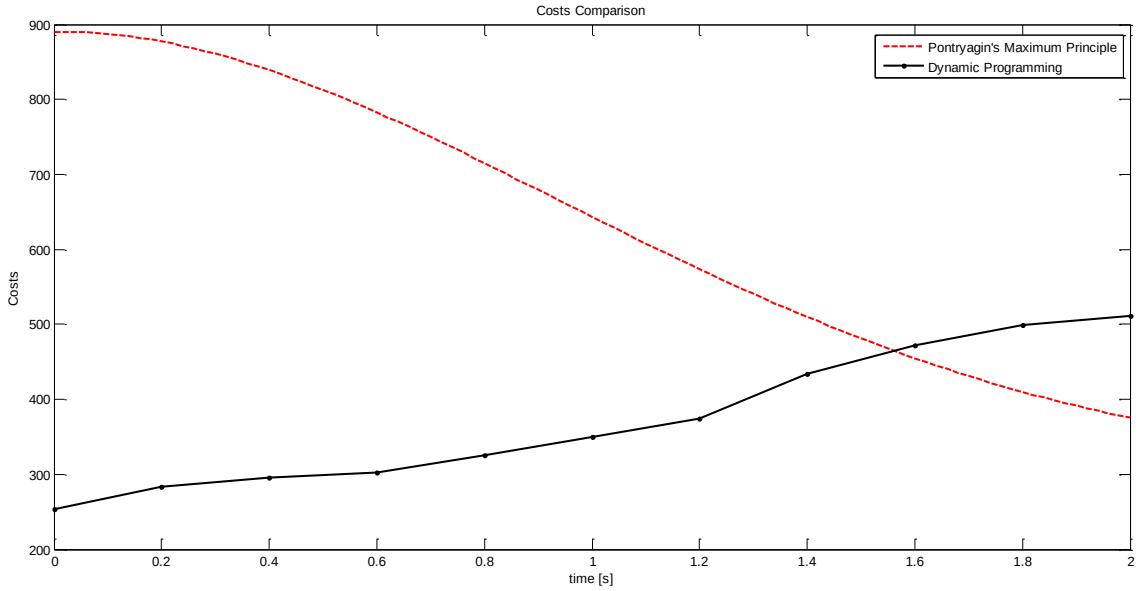

Figure 4.23 Performance (J) comparison between Dynamic Programming and Pontryagin Maximum Principle for the friction KERS.

The results in figure 4.20 show an oscillatory response from Dynamic Programming around the Pontryagin Maximum Principle Solution. However, it can be seen from figure 4.21 that the boundary conditions are met, and from figure 4.22, that the gear ratio squared follows closely the solution obtained from Pontryagin Maximum Principle. The cost shown in figure 4.23 is considerably higher for Dynamic Programming.

The results in figures 4.21 and 4.22 show consistency between the two methods employed. However, with the discretization of the system, the control obtained with Dynamic Programming is not consistent with Pontryagin's Principle and incurs a higher cost.

4.3 Application of Optimal Control to the simplified flywheel-based KERS with a CVT – Including brake friction (A two inputs system)

Regenerative braking generally works together with the friction brakes since it is not always possible to store all the kinetic energy in a vehicle given a specifically short period of time. An example that combines both braking and KERS is now considered.

Using the simplified model given by equations (4.5), (4.6) and (4.7), equation (4.6) can be modified to include a friction torque from the brakes. This relationship is given as:

$$f_1(t) = \dot{x}_1(t) = \dot{\omega}_w(t) = \frac{-J_f u(t) x_1(t) x_2{}^{0.5}(t) - u_2(t)}{J_w + J_f x_2(t)} \tag{4.58}$$

where $u_2(t)$ is an additional input which represents the friction brakes. The extra input has no direct impact on equation (4.7), so it can be used as previously stated.

When minimizing the difference of the gear ratio squared with a desired final value, an optimization of the amount of energy stored is taking place. Therefore, the performance index is then given as:

$$J(t) = \beta\left(x_2(t_f) - x_{2d}\right)^2 + \int_0^{t_f} \left(\alpha u^2(t) + \varsigma u_2{}^2(t)\right) dt \tag{4.59}$$

The control solution is now evaluated using Pontryagin's Maximum Principle and by solving the two-point boundary-value problem numerically (using Matlab).

Application of Pontryagin's Maximum Principle - An Optimal Energy Approach

Turning to the use of Pontryagin's Maximum Principle, the formulation of the problem is as follows: The Hamiltonian for the simplified KERS model including friction brakes is given as:

$$\begin{aligned} H(t) = {} & \left(4\beta u(t) x_2{}^{0.5}(t)\left(x_2(t) - x_{2d}\right) + \alpha u^2(t) + \varsigma u_2{}^2(t)\right)\psi_0(t) \\ & + \frac{-J_f u(t) x_1(t) x_2^{0.5}(t) - u_2(t)}{J_w + J_f x_2(t)} \psi_1(t) + 2u(t) x_2^{0.5}(t) \psi_2(t) \end{aligned} \tag{4.60}$$

And, from equation (4.60) the co-state functions are given as:

$$\dot{\psi}_0(t) = 0 \tag{4.61}$$

and

$$\dot{\psi}_1(t) = \frac{J_f u(t) x_2^{0.5}(t)}{J_w + J_f x_2(t) + m_v r_w^{\ 2}} \psi_1(t) \tag{4.62}$$

and

$$\begin{aligned} \dot{\psi}_2(t) = &-\left(6\beta u(t) x_2^{\ 0.5}(t) - 2\beta x_{2d} u(t) x_2^{\ -0.5}(t)\right)\psi_0(t) \\ &+ \frac{0.5 J_w J_f u(t) x_1(t) x_2^{-0.5}(t) - 0.5 J_f^2 u(t) x_1(t) x_2^{0.5}(t) - J_f u_2(t)}{\left(J_w + J_f x_2(t)\right)^2} \psi_1(t) \\ &- u(t) x_2^{-0.5}(t) \psi_2(t) \end{aligned} \tag{4.63}$$

And, the condition for optimality is given as:

$$\begin{aligned} \frac{\partial H(t)}{\partial u(t)} = &\left(4\beta x_2^{\ 1.5}(t) - 4\beta x_{2d} x_2^{\ 0.5} + 2\alpha u(t)\right)\psi_0(t) \\ &- \frac{J_f x_1(t) x_2^{0.5}(t)}{J_w + J_f x_2(t) + m_v r_w^{\ 2}} \psi_1(t) + 2 x_2^{0.5}(t) \psi_2(t) \end{aligned} \tag{4.64}$$

and

$$\frac{\partial H(t)}{\partial u_2(t)} = 2\varsigma u_2(t) \psi_0(t) - \frac{\psi_1(t)}{J_w + J_f x_2(t)} \tag{4.65}$$

From equation (4.64) and (4.65), the control variables are given as:

$$\begin{aligned} u(t) = &\frac{J_f x_1(t) x_2^{0.5}(t)}{J_w + J_f x_2(t) + m_v r_w^{\ 2}} \frac{\psi_1(t)}{2\alpha\psi_0(t)} - \frac{x_2^{0.5}(t)}{\alpha} \frac{\psi_2(t)}{\psi_0(t)} \\ &- \frac{2\beta x_2^{\ 0.5}(t)\left(x_2(t) - x_{2d}\right)}{\alpha} \end{aligned} \tag{4.66}$$

and finally:

$$u_2(t) = \frac{\psi_1(t)}{J_w + J_f x_2(t)} \frac{1}{2\varsigma\psi_0(t)} \tag{4.67}$$

As it has been seen earlier, the final condition for $x_2(t_f)$ is not specified and additional information is required to complete the boundary conditions. Using equation (4.15), the additional relationship is given as:

$$\psi_2(t_f) = \beta(x_2(t_f) - x_{2d}) \tag{4.68}$$

The formulation of the problem using Pontryagin's Maximum Principle is now complete. Three examples are presented where the energy optimization is clearly shown; the parameter values for the system are shown in Table 4.8 and the boundary conditions are given in Table 4.9

Friction KERS	
Parameteres	
α	20
β	10
ς	0.1
Jw [kg*m^2*rad^-2]	2
Jf [kg*m^2*rad^-2]	1

Table 4.8 – System Parameters

Boundary Conditions		
	to	tf
time [s]	0	15
x1 (t) [rad/s]	60	0
x2 (t) (gear ratio squared)	0.16	NS

Table 4.9 – Boundary conditions

The vehicle is to be taken from an initial speed to rest in a given period of time minimizing the performance index given by equation (4.59). The desired final values for the gear ratio squared are shown in Table 4.10.

Desired final Gear ratio square	
Case 1	0.16
Case 2	10
Case 3	20

Table 4.10 Different desired values for x_2

The comparisons between the three different cases are shown in figures 4.24 to 4.26:

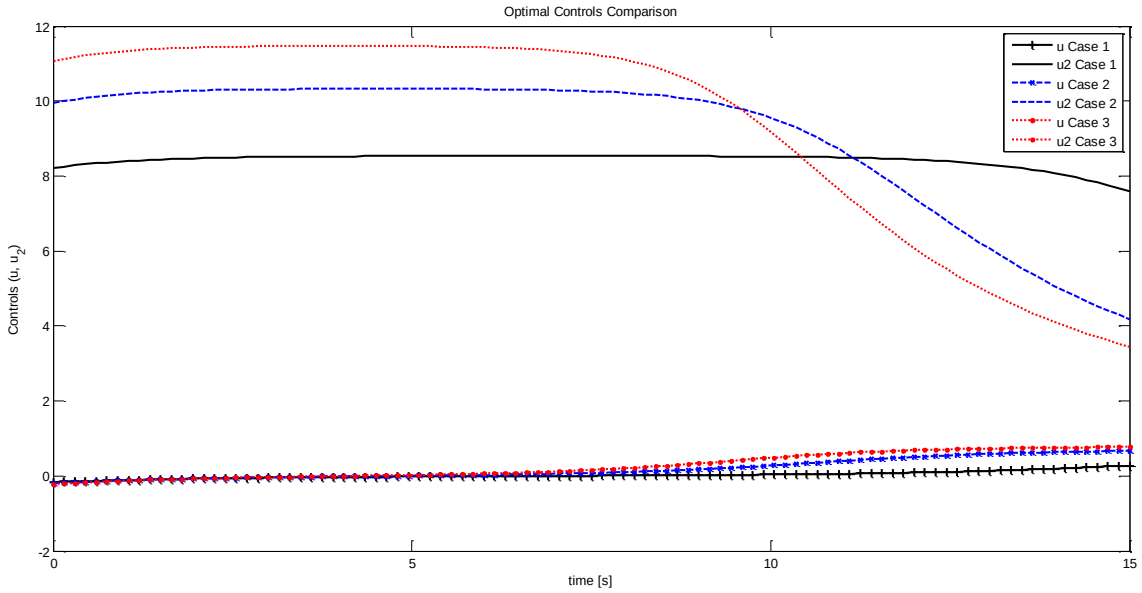

Figure 4.24 Optimal controls comparison for three different desired final values for x_2.

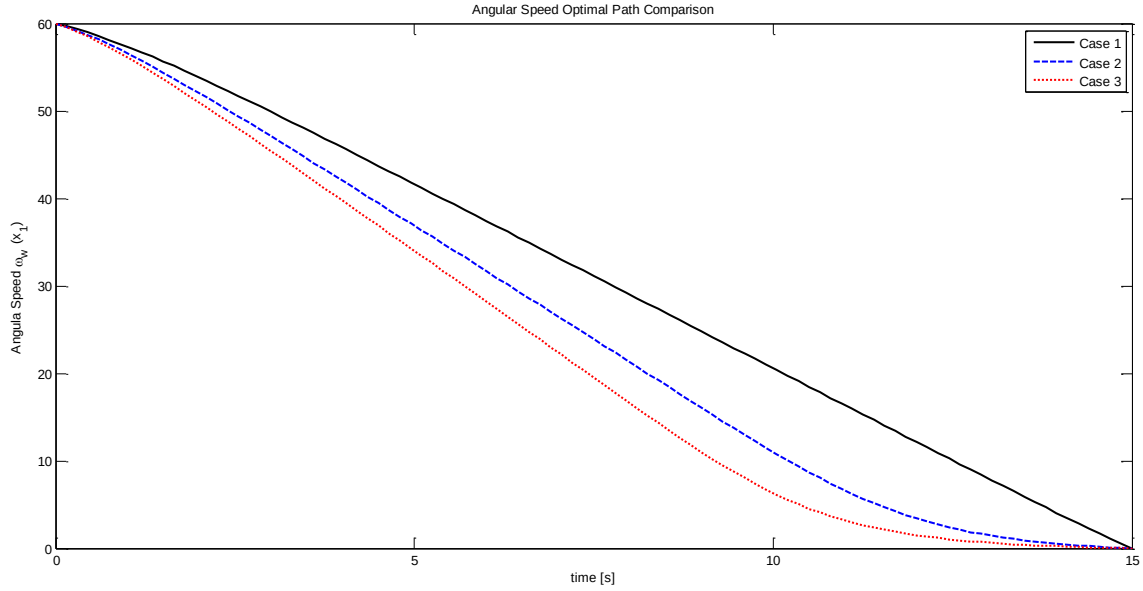

Figure 4.25 Angular speed optimal path comparison for three different desired final values for x_2.

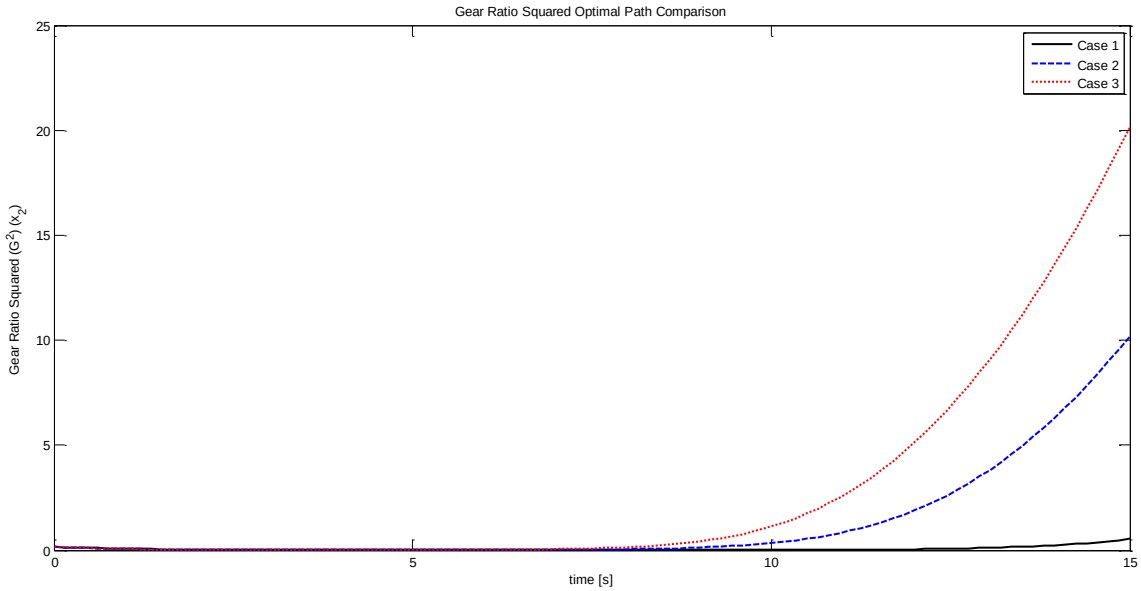

Figure 4.26 Gear ratio squared optimal path comparison for three different desired final values for x_2.

The results in figure 4.25 show that the boundary conditions are met, and in figure 4.26 it can be seen that the solution is close to the desired final value for x_2. In figure 4.24 it can be seen that the brakes and the KERS combine in order to achieve the system specifications.

It can be seen that the results show that the energy stored is optimized and the use of the brakes is reduced when the KERS activates. It has been shown, using a simplified representation, what the impact of the KERS is, and how it combines with the friction brakes. A more realistic representation of the system is achieved by combining the more realistic models developed in Chapter 2 with the inclusion of the friction brakes. This result is however not yet verified with Dynamic Programming since the additional control variable requires further modification, as well as a somewhat different approach from the strategy developed this far.

4.4 Optimal Control Summarizing Table and Practical Implementation

Summarizing Table

Table 4.11 shows a compact summary of the different optimal control objectives, emphasising their applications and stating their cost functions.

Control Objective	Description	Cost Function
General form	• Minimization of the state and control variables is possible as well as time • An desired final state value can be included	$J = \int_{t_o}^{t_f} f_0(\bar{x}(t), \bar{u}(t), t)dt + h(\bar{x}(t_f), t_f)$
Time Optimization	• Minimization of the time to meet the boundary conditions and specifications	$J = \int_{t_o}^{t_f} f_0(t)dt$
"Fuel" Optimization	• Minimization of the control variable • Final time can be fixed or free	$J = \int_{t_o}^{t_f} f_0(\bar{u}(t))dt$

Table 4.11Summary of the optimal control performance criteria measurements

It is important to mention that for the KERS applications included in this chapter the "Fuel" Optimization strategy (equation (4.8)) is used to obtain a control with smooth changes of the gear ratio, and the General form (equation (4.48) is used to maximize the energy stored in the flywheel.

Practical Implementation

The discussion of this chapter has focused on two optimal control methodologies, namely discrete Dynamic Programming, and the Pontryagin's Maximum Principle; which along with other modern control methodologies that are used in new technologies, require a heavy computational effort to find the

desired control strategies. This makes the online implementation virtually impossible and for this reason, in a good deal of cases the control solutions would be found offline leading to a practical implementation via Lookup Tables or Path Tracking strategies.

4.5 Conclusions of the findings in Chapter 4

The conventional Optimal Control strategies of Pontryagin's Maximum Principle, and Dynamic Programming have been successfully implemented for a flywheel-based KERS with a CVT application. The implementation of Dynamic Programming based on the strategy developed in Chapter 3 (inverting Runge Kutta integration method) needed further investigation and development, since results were unsatisfactory for this application, the conclusion that the recursive characteristics used in the Runge Kutta algorithm, in addition to the nonlinear characteristics of the system, increased the possible values that the control variable could take was drawn, and indeed shown. The Modified Euler method was implemented and tested giving successful results and the improved discrete Dynamic Programming strategy was updated to use this method to do the inversion in the integration procedure. The results obtained for a simplified model and a friction based model, were contrasted between the optimal control strategies, showing consistency in the trajectories but discrepancies in the control variable. But also incurring differences in the performance index. At the end of this Chapter a two-input simplified flywheel-based KERS with a CVT was developed to show clearly an energy optimization case, the two inputs represents the combination of using the KERS with friction brakes. The advantages and disadvantages of the two conventional Optimal Control strategies can be seen in the chapter. Pontryagin's Maximum Principle can easily lead to laborious algebraic operations, and for different models or different optimization requirements, the strategy has to be implemented from the start. On the other hand, Dynamic Programming needs very little modification. The results obtained from the improved discrete Dynamic Programming were accurate, despite the fact that the control obtained does not always follow very closely to the Pontryagin's Maximum Principle solution, the behaviour of the state trajectory is satisfactory. The developed strategy is not only accurate but converges for a much reduced number of time steps. With

further development of this Dynamic Programming method it should prove appropriate for use on more realistic optimal energy management of KERS provided there are not too many states in the overall model.

5. ASSESMENT OF OPTIMAL CONTROL STRATEGIES FOR A FLYWHEEL-BASED KERS WITH A CVT

Full implementation of conventional Optimal Control strategies for flywheel-based KERS with a CVT has been achieved in Chapter 4. In this Chapter, a comparison with a more conventional classical control strategy is made. The system models are implemented in Matlab-Simulink and verified using results from Chapter 4. Two controllers (namely a proportional (P), and a proportional plus 'integral' (PI)) are tuned to meet the boundary conditions. The results are evaluated and compared with the optimal control solutions obtained in Chapter 4.

5.1 Flywheel-based KERS with a CVT - modelling in Matlab-Simulink

The simplified flywheel-based KERS with a CVT model described in Chapter 2 (and reduced in Chapter 4 to equations (4.5), (4.6) and (4.7)) is considered for implementation in Matlab-Simulink for classical control purposes. The system state space form is given as follows:

$$x_1(t) = \omega_w(t) \tag{5.1}$$

and

$$f_1(t) = \dot{x}_1(t) = \dot{\omega}_w(t) = \frac{-J_f u(t) x_1(t) x_2^{0.5}(t)}{J_w + J_f x_2(t)} \tag{5.2}$$

and

$$f_2(t) = \dot{x}_2(t) = 2u(t) x_2^{0.5}(t) \tag{5.3}$$

The system of equations (5.1), (5.2) and (5.3) are close in definition to what is called a bilinear system. Elliot (2009) states "the word bilinear means that the velocity contains a ux term but is otherwise linear in x and u. In this case, the system is linear in u but non-linear in x. Therefore, the approach to simulate the control of this problem is achieved using the Matlab tool Simulink.

Figure 5.1 shows the block diagram representation of the system described before.

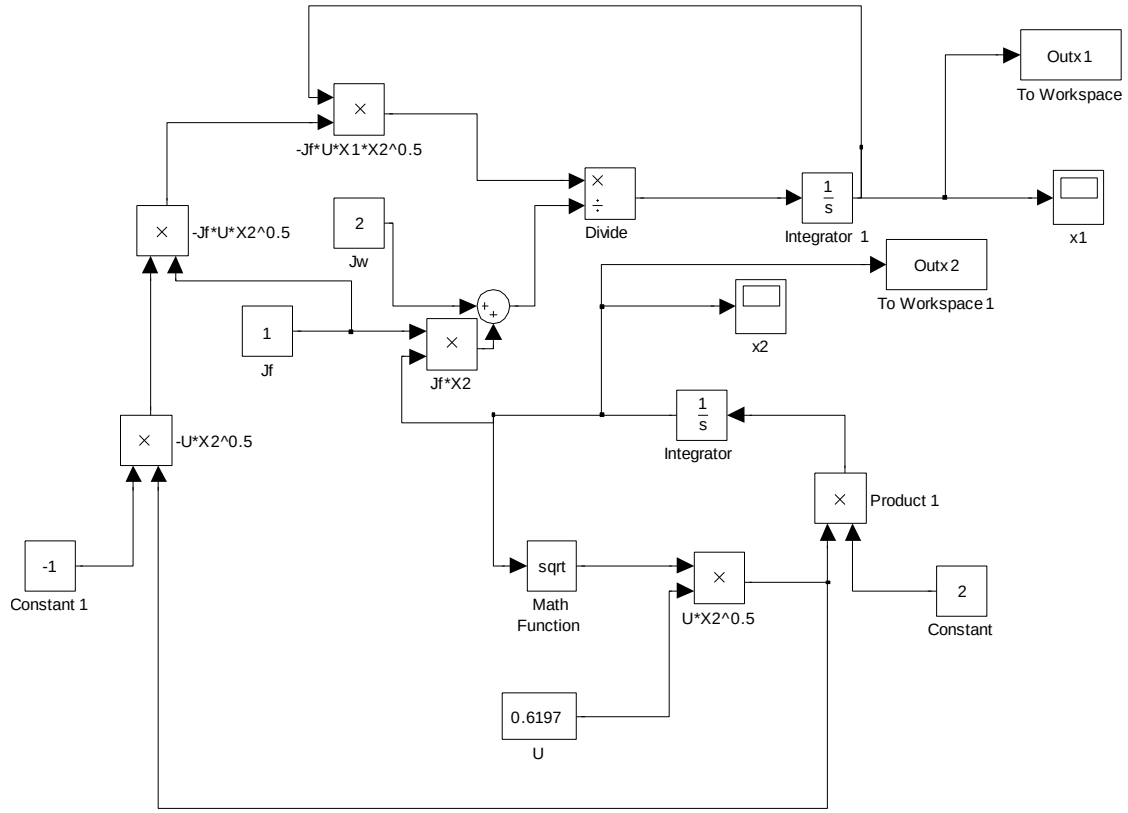

Figure 5.1 Block Simulink diagram for the simplified flywheel-based KERS with a CVT

The model shown in Figure 5.1 is verified using the results from figure 4.1, where a constant control $u(t)$ of magnitude 0.6197 was found to drive the system from the initial conditions $\bar{x}(t_0) = [60 \quad 1]^T$ to the final conditions $\bar{x}(t_f) = [10 \quad 106]^T$. The results are shown in figures 5.2 and 5.3, and the systems specifications are drawn from Table 4.1.

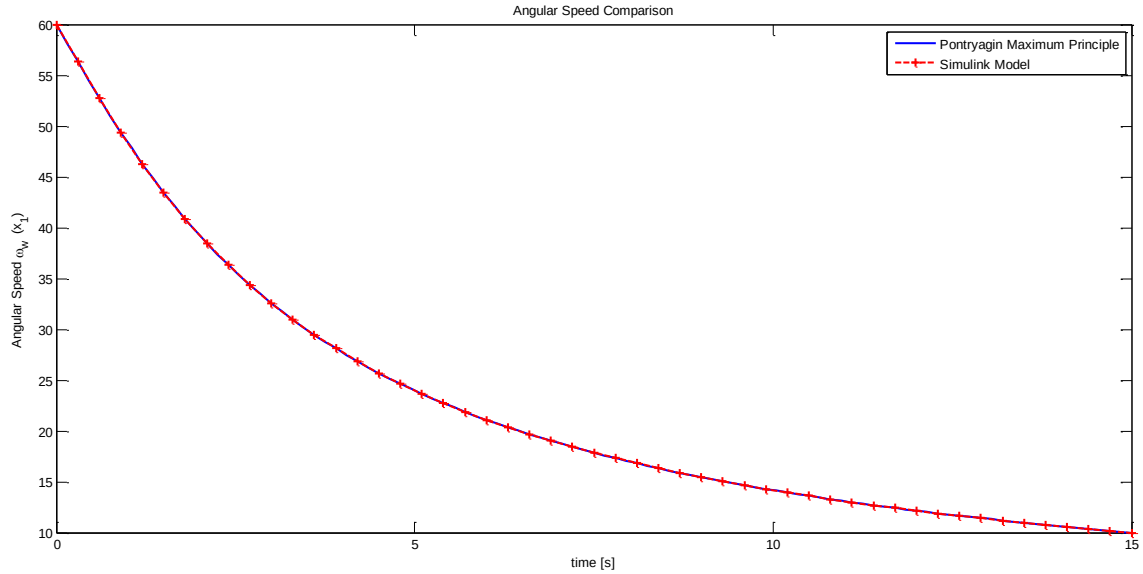

Figure 5.2 Angular speed trajectory to validate the systems representation using Simulink

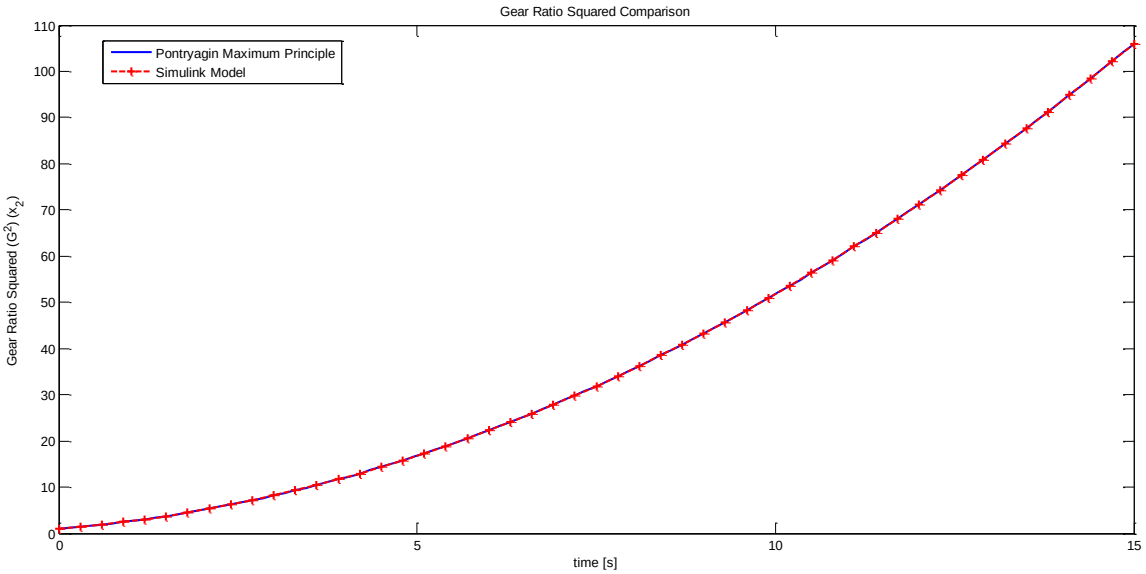

Figure 5.3 Gear ratio squared trajectory to validate the systems representation using Simulink

Figures 5.2 and 5.3 show that the boundary conditions specified are met and with this the model can be verified.

With the verification of the system complete, the necessary modifications can be made in order to implement a classical control controller.

5.2 Classical Control implementation for simplified and friction flywheel-based KERS with a CVT

Here both the Simplified KERS model of Chapter 2 and the KERS model with friction are examined.

A Comparison using the simple model in the Fuel-optimal-control problem

The system implemented in Simulink is slightly modified for control purposes, this is shown in figure 5.4. The block that contains the input variable $u(t)$ is replaced by a "summing point" and a PID controller block. The systems requirement is to drive a vehicle from given initial speed to a specified final speed. Therefore, the signal representing $x_1(t)$ is connected to the summing point, thereby closing the loop (feedback control). The other input to the summing point is given by a set up reference (in this case the final condition for $x_1(t_f)$). The output of the summing point is connected to the PID controller and its output is used as the input variable.

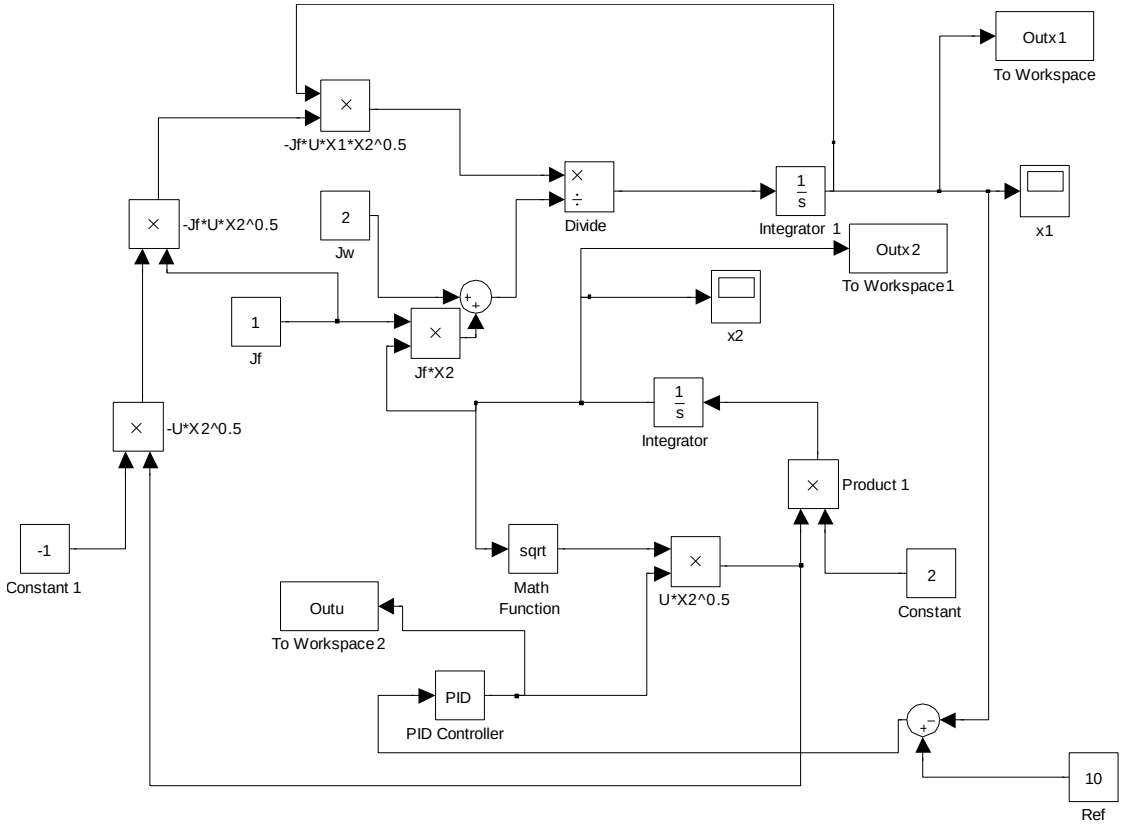

Figure 5.4 Block Simulink diagram for the simplified flywheel-based KERS with a CVT using a PID controller.

The PID controller is set up with the values given in Table 5.1. It can be seen that only the proportional constant is used for this example making it a P controller.

PID gain values Case 1	
Kp	-0.1
Kd	0
Ki	0

Table 5.1 The proportional (P) controller gain

The results are shown in figures 5.5, 5.6, and 5.7, including a comparison with the optimal control solution obtained in Chapter 4.

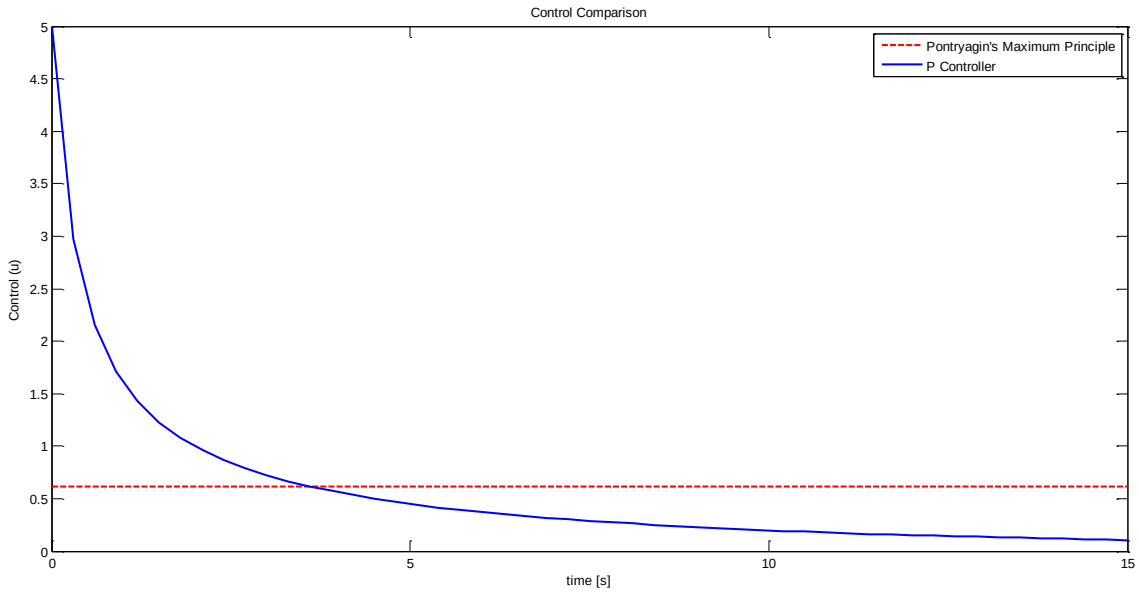

Figure 5.5 Control comparison – P Controller and Pontryagin Maximum Principle solutions (J=483)

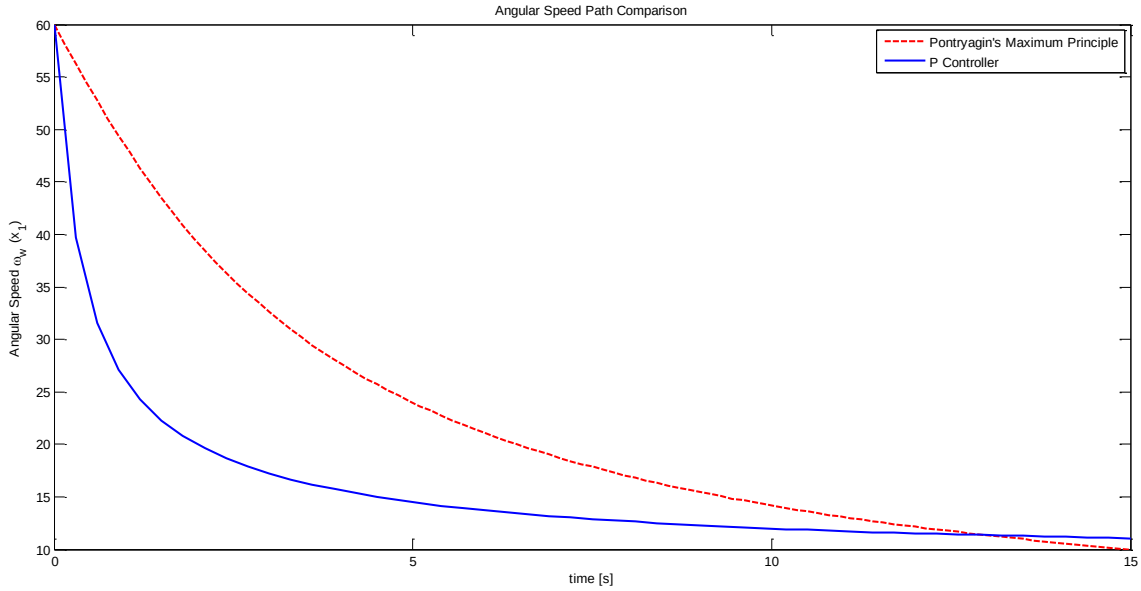

Figure 5.6 Angular speed comparison – P Controller and Pontryagin Maximum Principle solutions

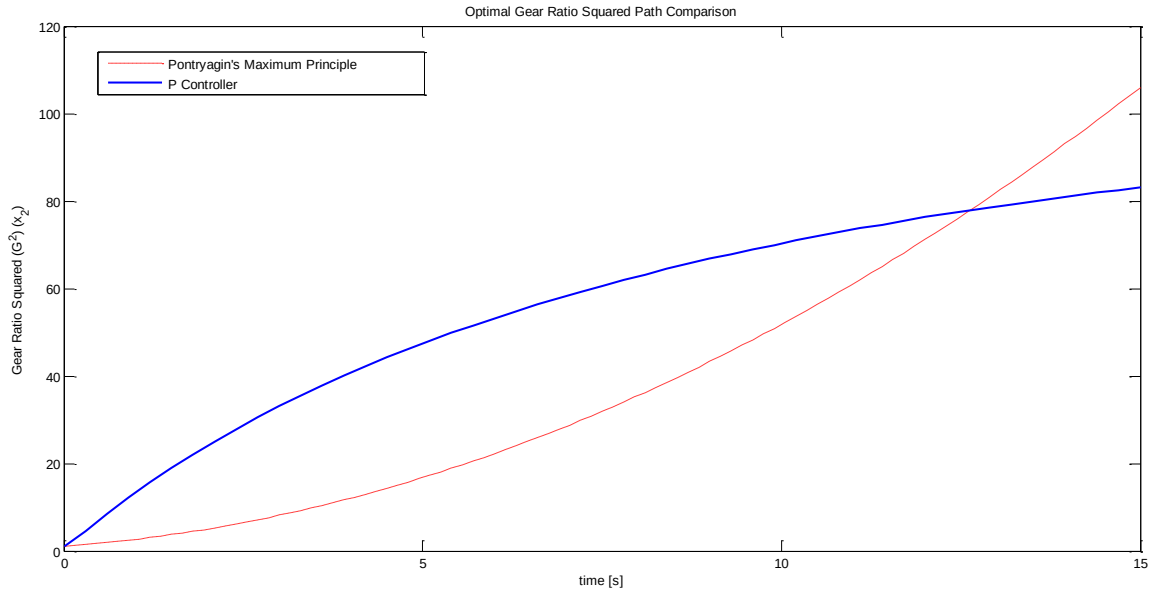

Figure 5.7 Gear ratio squared comparison – P Controller and Pontryagin Maximum Principle solutions

The results in figure 5.6 show a very rapid response from the controller, getting very close to the target in about five seconds but rapidly becoming steady and incurring a steady state error. Therefore, the results for the gear ratio in figure 5.7 also have steady state error. The control found in figure 5.5 shows a very high input at the beginning that decreases with time. The evaluation of the

performance index produced a much larger value than using optimal control via Pontryagin's Maximum Principle.

It is clear from the results (and is indeed well-known) that a proportional controller cannot drive the system to the specific target.

A proportional-integral (PI) controller is now considered for implementation. The integral feature introduced in the controller reduces, or eliminates, the steady state error. The controller gains are shown in Table 5.2, where the proportional gain is reduced in order to smooth the shape of the control to reduce the control cost control. The constant of integration was tuned to meet the boundary conditions.

PID gain values Case 2	
Kp	-0.05
Kd	0
Ki	0.003

Table 5.2 The proportional-integral (PI) controller gains

The results are shown in figures 5.8, 5.9, and 5.10

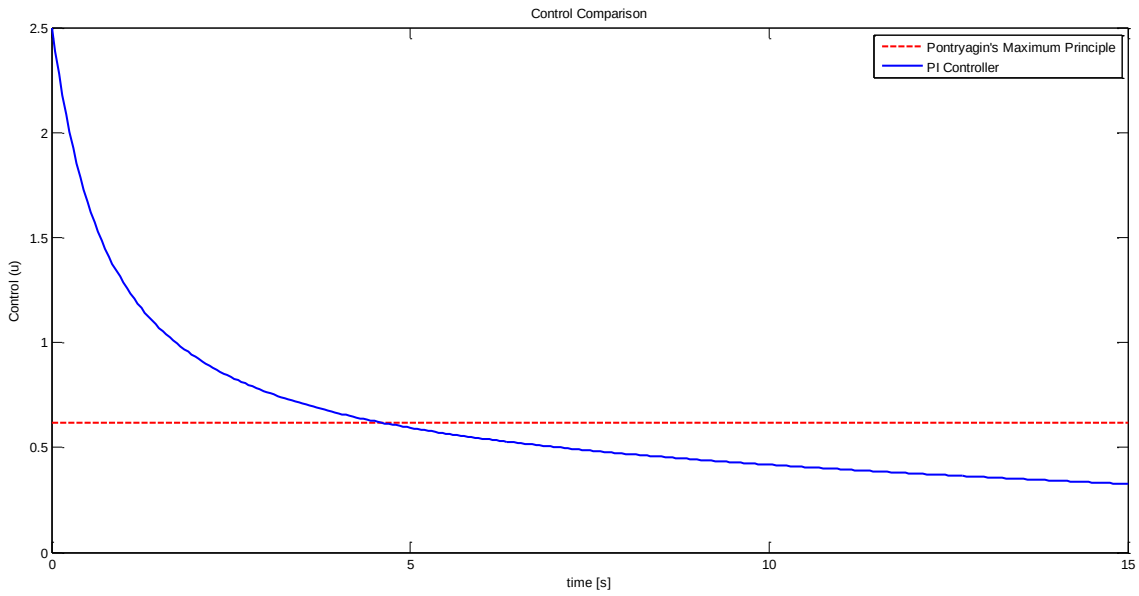

Figure 5.8 Control comparison – PI Controller and Pontryagin Maximum Principle solutions (J=317)

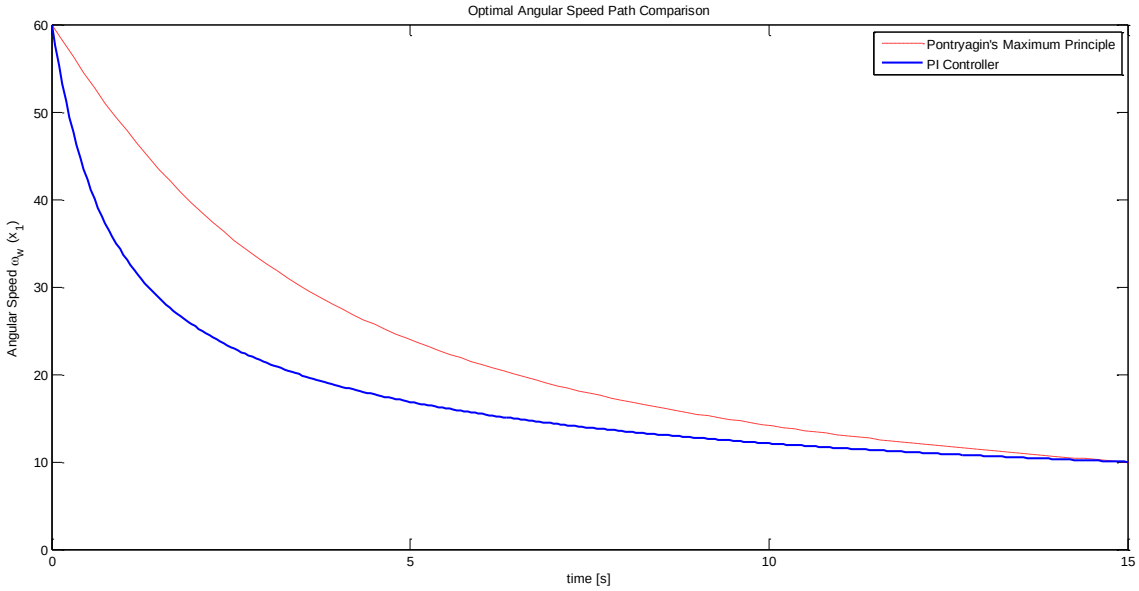

Figure 5.9 Angular speed comparison – PI Controller and Pontryagin Maximum Principle solutions

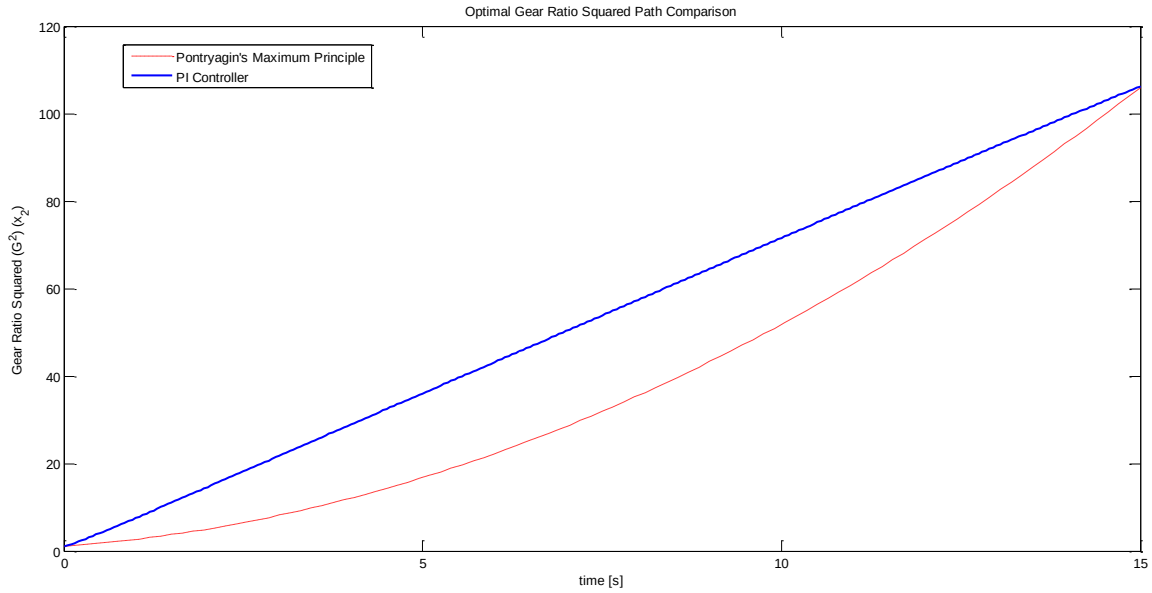

Figure 5.10 Gear ratio squared comparison – PI Controller and Pontryagin Maximum Principle solutions

It can be seen from figures 5.9 and 5.10 that the boundary conditions are met. The control shown in figure 5.8 starts from a value near to 2.5 and decreases with time, the performance index is 317, which is still far above from the value of 230 obtained from Pontryagin's Maximum Principle.

The implementation of a PI controller using Matlab-Simulink shows no improvement respect to the 'fuel-optimal' control solution obtained with the conventional strategies. The PID controller is therefore not considered for implementation since the results suggest that the best solution would be to explore the tuning combinations of the proportional and integral constants; the derivative part of a controller contributes to a faster response but also suggests much higher costs.

A Comparison using the friction model in the energy-optimal-control problem

As part of the assessment of the optimal control strategies implemented in Chapter 4, the energy-optimal control is now considered. The system shown in figure 5.4 is modified to include the friction of the rotating elements (traction wheel and flywheel). The block diagram of the friction flywheel-based KERS with a CVT is shown in figure 5.11

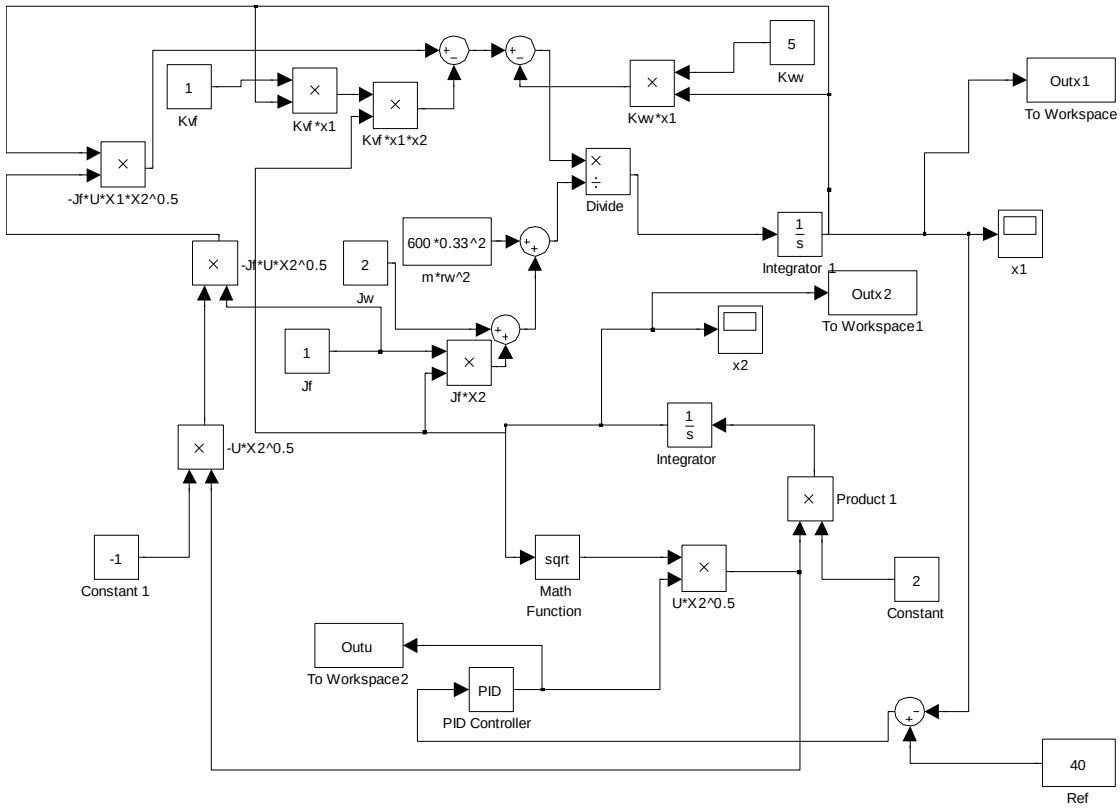

Figure 5.11 Block diagram of the friction flywheel-based KERS with a CVT using Simulink

A proportional controller is again tuned to meet the boundary conditions in this energy-optimal control problem. The parameters for the controller are shown in Table 5.3.

PID gain values Case 3	
Kp	-0.151
Kd	0
Ki	0

Table 5.3 The proportional (P) controller gain

The results are shown in figures 5.12, 5.13, and 5.14.

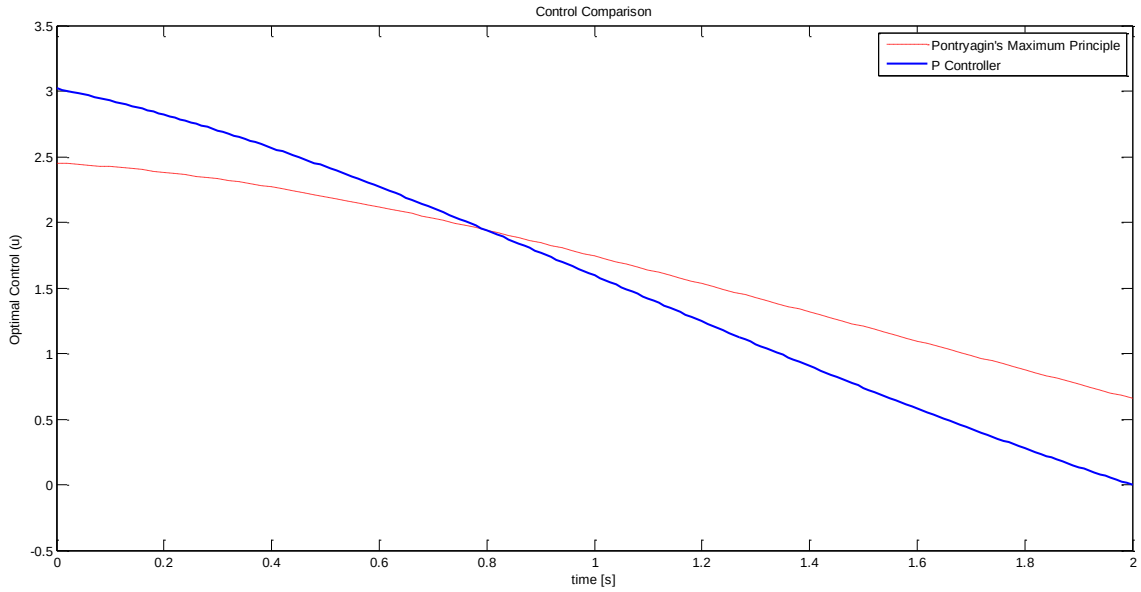

Figure 5.12 Control comparison – P Controller and Pontryagin Maximum Principle solutions

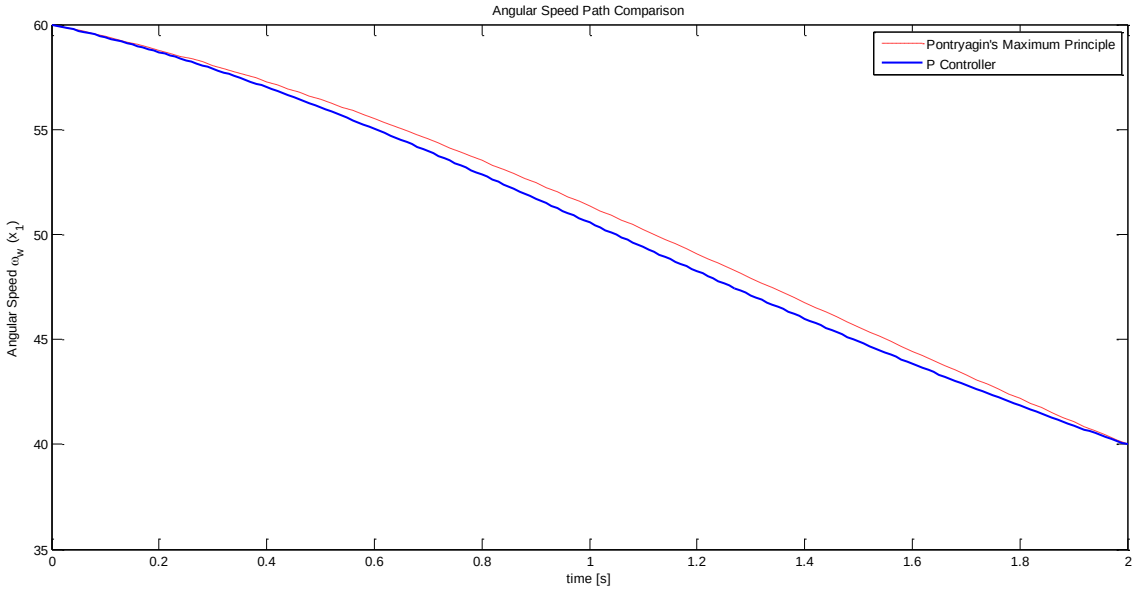

Figure 5.13 Angular speed comparison – P Controller and Pontryagin Maximum Principle solutions

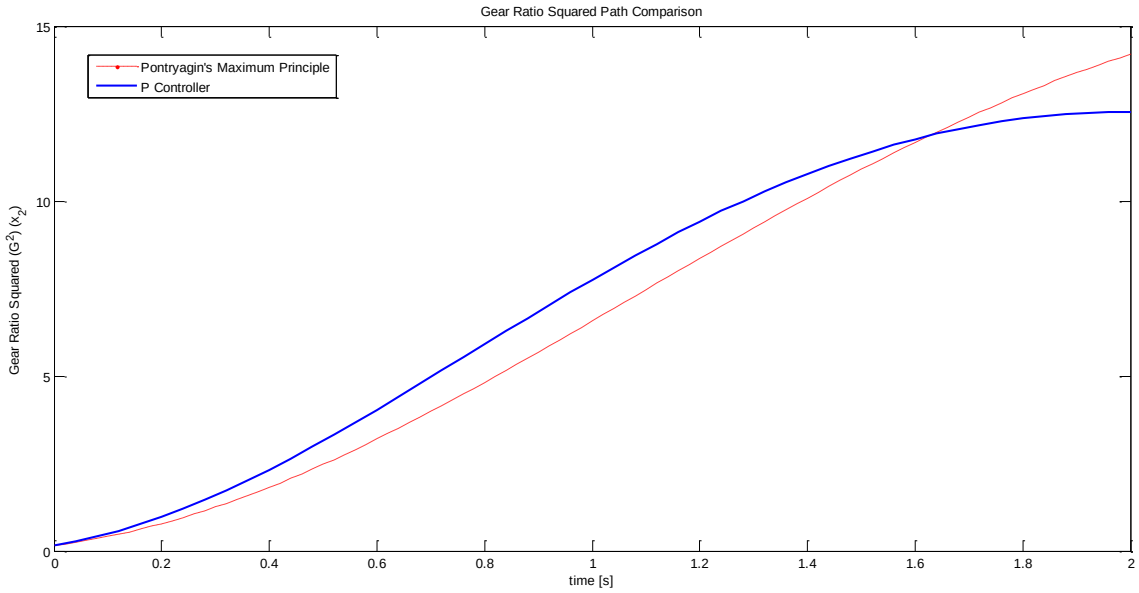

Figure 5.14 Gear ratio squared comparison – P Controller and Pontryagin Maximum Principle solutions

Figure 5.12 shows that the proportional controller generates a solution relatively close to the optimal control. In figures 5.13 and 5.14 it can be seen that the boundary conditions are met, and for the speed comparison the system is driven very closely to the optimal path. However, for the energy storage optimization the final gear ratio is lower than the optimal.

For the friction model, the proportional controller shows a good response, relatively close to the optimal. But the energy optimization is not achieved, which suggests that modification to the gain and the introduction of an integral gain would help to achieve this.

A PI controller is now considered, although a higher gain in the proportional controller seems to represent a higher control variable and therefore an increment in the energy stored, it is not necessarily the case and the possibility is shown. The controller parameters are shown in Table 5.4:

PID gain values Case 4	
Kp	-0.2
Kd	0
Ki	0.04

Table 5.4 The proportional-integral (PI) controller gains

The results and comparison with the optimal solution are shown below in figures 5.15, 5.16, and 5.17.

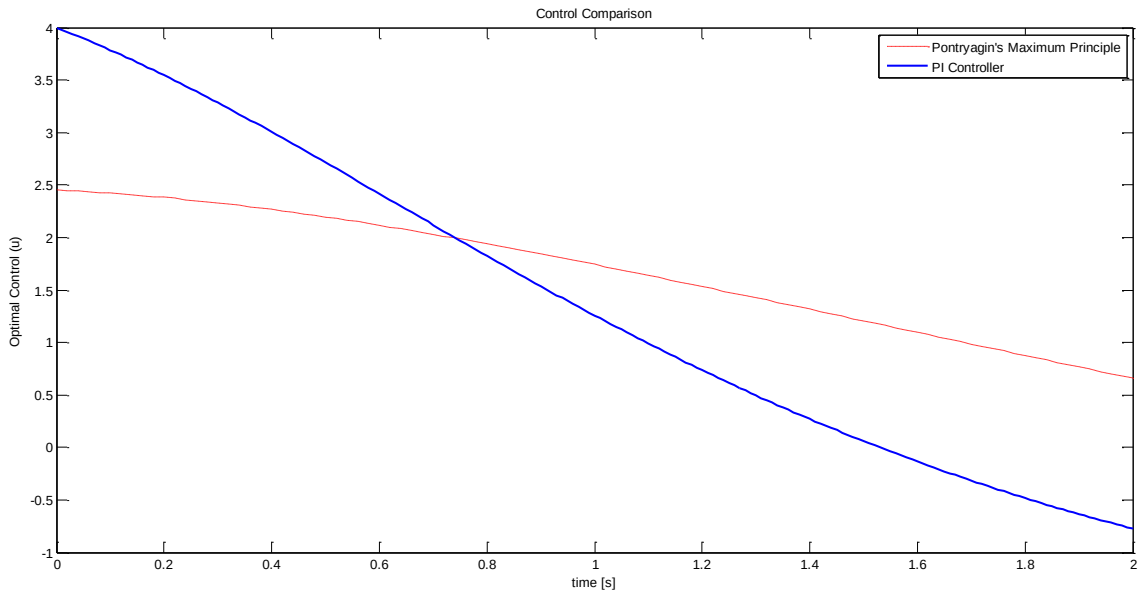

Figure 5.15 Control comparison – PI Controller and Pontryagin Maximum Principle solutions

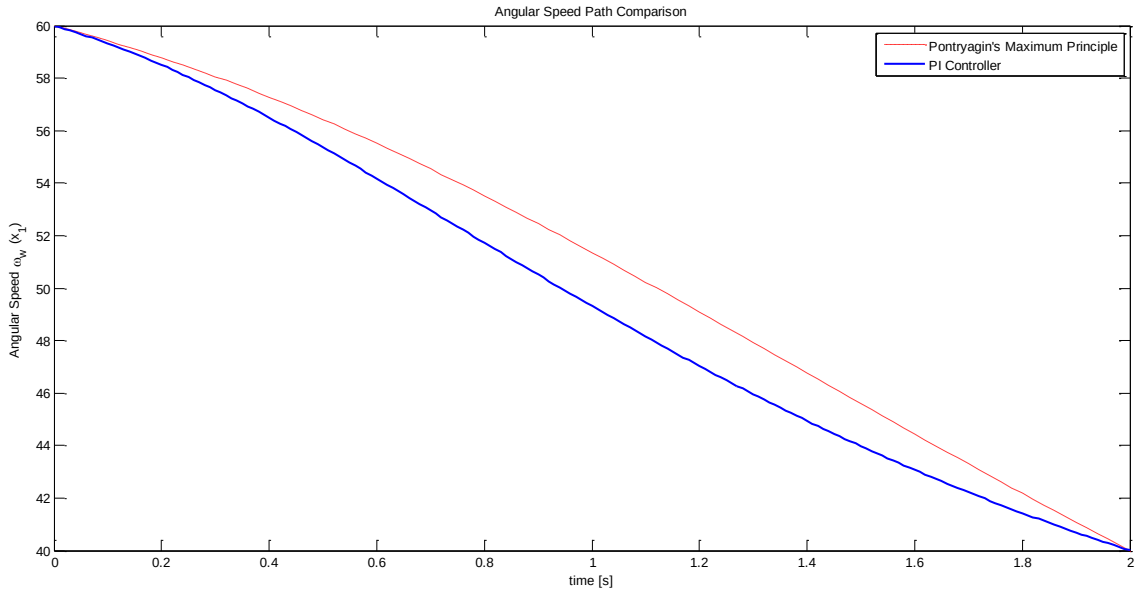

Figure 5.16 Angular speed comparison – PI Controller and Pontryagin Maximum Principle solutions

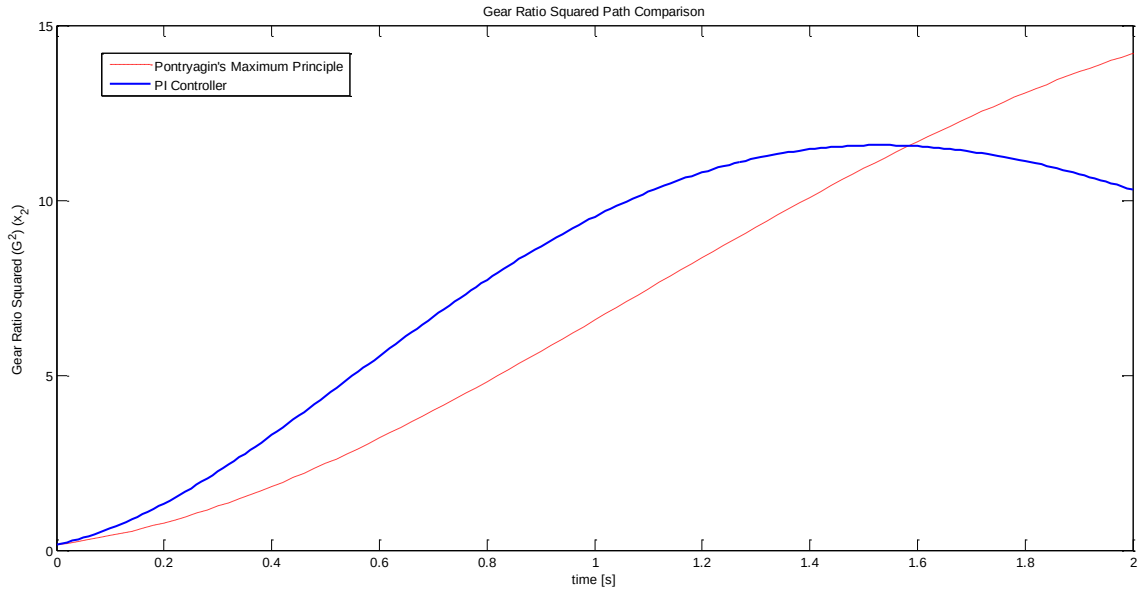

Figure 5.17 Gear ratio squared comparison – PI Controller and Pontryagin Maximum Principle solutions

The control found is shown in figure 5.15, it can be seen that it goes from a relatively high value to a negative value at the final time. Figures 5.16 and 5.17 show that the boundary conditions are met, however in figure 5.17 it can be seen that the energy optimization was not achieved.

The negative value for the control means that the KERS is using the energy stored to compensate for friction losses, actually causing the energy stored to drop. Therefore to consider friction losses and to maximize the energy stored, is not at all a straight forward task in the parameter selection for classical control strategies implementation thereby giving justification to the use of optimal control.

5.3 Conclusions of the findings in Chapter 5

The comparison between optimal control solutions and classical control implementation has been successfully achieved. The optimal control solutions show better performance. However, the tuned classical PI control strategy responds well considering that no explicit optimization was specified, especially for the simplified system. It can be noted that the more specifications there are applied to the system, the more limited a classical control strategy will be.

The relatively easy implementation of classical control theory shows the reason for its wide use; however in this Chapter some of the classical control theory limitations have been highlighted. This was clearly shown when using a performance criterion that involves the minimization of the control variable or the maximization of the energy stored in the KERS; also, the ‘two-inputs’ system shown in Chapter 4 could not be implemented by following simple classical control procedures. On the other hand, the use of optimal control strategies shows a good response given the systems behaviour requirements and performance criterion, this is clearly seen since both the smoothness of the control and the energy maximization have been achieved. Although optimal control strategy implementation is more complicated than for the classical approach, it shows some benefits, especially in terms of the desired performance, which means that optimal control theory can be considered for modern systems which have more complex demands.

6. CONCLUSIONS AND FUTURE WORK

6.1 Conclusions

The overall aim of this thesis was to develop and determine an appropriate optimal control methodology for a flywheel-based Kinetic Energy Recovery System (KERS). This was to be achieved through the identification and selection of a simple but representative system model, and the implementation of conventional optimal control strategies. An assessment of an optimal KERS control implementation involving a comparison with a non-optimal strategy has been made, in order to show the benefits and potentials of optimal control.

A number of flywheel-based KERS driven via a CVT model have been evaluated, via kinematic and dynamic models analysis, and appropriately modified to suit the control requirements. This representation includes an ideal model, two friction based models (for ball bearings, and magnetic bearings), and an extended model including a hydraulic actuator acting on a pulley-based CVT (which is included to contrast simplified models with a more realistic version). The models shown are representative when compared with the models currently used in relevant literature.

In order to build up a suitable optimal control strategy, the theory was briefly compared with classical control theory, and then detailed implementation was made (for a second order oscillator) using the two most common optimal control methodologies, namely Pontryagin's Maximum Principle, and Dynamic Programming. This implementation showed the challenges that each methodology presented, with particular emphasis placed on the potential of Dynamic Programming because of its ability to provide both necessary and sufficient conditions for optimality, whereas Pontryagin's Maximum Principle only provides necessary conditions. Owing to the limitations of standard Dynamic Programming, a new improved discrete time version was developed and implemented. This modified version finds a solution to the problems arising with the interpolation strategies that are normally present in standard Dynamic Programming. The developed strategy achieves higher accuracy by finding the control parameters at each iteration, which is achieved by inverting the

integration process. An Euler numerical integration method is first implemented, where piece-wise constant control is assumed. However, it was found that the Euler method converges only for a very high number of time steps, introducing errors for a reduced number of time steps. For this reason, the Runge Kutta numerical integration method was implemented, where constant control between sample points was initially assumed but where piece-wise linear control between sample points was ultimately considered. It was demonstrated that working with constant control is insufficient because, by introducing a recursive method (such as the Runge Kutta scheme) the system becomes oversubscribed and has to be solved via a least square method. But even when use of constant control seemed promising in following a known trajectory, the solution was compromised when implemented in full Dynamic Programming. Implementation of full Dynamic Programming (in a so called improved modified version) was successfully achieved by inverting the Runge Kutta numerical integration method and assuming piece-wise linear control. This strategy showed high accuracy, and very good computational efficiency but only for application to a linear oscillator optimal control problem.

When the developed modified Dynamic Programming methodology, and Pontryagin's Maximum Principle, were applied to the control of a flywheel-based KERS with a CVT, the modified version of Dynamic Programming was found to need further development. The conclusion drawn from several different numerical experiments was that the recursive feature of the Runge Kutta numerical integration method, increases the number of possible solutions needed to find the control. For this reason, the Modified Euler method was considered instead and tested, giving very successful results.

Full implementation of the improved modified version of Dynamic Programming was undertaken, and the results were compared to these obtained using the Pontryagin's Maximum Principle and showed general consistency. Some, Dynamic Programming derived control variables show unwanted oscillatory behaviour which are dependent on the discretization, but show high accuracy with a reduced number of time steps can still be achieved.

A simplified model for the KERS was used to explore control variable optimization, and a more realistic KERS model (including friction) was used to explore energy optimization. It is found that Pontryagin's Maximum Principle requires quite laborious algebraic manipulations, but also that for use with different models, or for application with different optimal control objectives, a complete reformulation of the problem is needed. By contrast, Dynamic Programming needs very little modification even for quite different models, except when the system order or the number of inputs are increased. The improved modified version of Dynamic Programming generally proves to be accurate and shows convergence for reduced number of time steps.

In order to assess the application of optimal control theory to a flywheel-based KERS with a CVT, a brief comparison with classical control was undertaken. Both simplified and more realistic models are implemented in Matlab-Simulink, and appropriate controllers are tested. The controllers used are proportional (P) control, and proportional plus 'integral' (PI) control; both are implemented and tuned manually to meet specific system requirements. The comparison with the results obtained from the optimal control implementation showed the benefits of using optimal control strategies. For the simplified model (used for control variable optimization), the P controller produces steady state errors, whereas the PI gives an accurate sub-optimal response. In the example, the PI controller can be tuned without major difficulty to have a sub-optimal response. In the case of a more realistic model (used for energy optimization), it is possible to tune the P controller to meet the boundary conditions giving a reasonable non-optimal solution. This is also true for the PI controller. However, if the controller parameters are to be tuned for energy optimization, a major challenge arises since various considerations must be taken into account. The results confirm that the optimal control significantly improves the system response compared to application of classical control strategies.

The development and implementation of an Optimal Control strategy for a flywheel-based KERS with a CVT was therefore successful. However, the results from the study have identified further research.

6.2 Future Work

As mentioned, improvement in the developed modified Dynamic Programming is needed, especially to make the control variable smoother. A study of potential, of imposing continuity conditions was initiated but has not yet been applied to full Dynamic Programming. Also, a reduced “first time step iteration” is worthy of further investigation. This would increase the accuracy of the response considerably reducing the oscillations. For non-linear systems, a non-uniform discrete mesh (i.e. a non-linear mesh) could improve computational efficiency. The possibility of having different mesh size gets benefit in using the developed modified Dynamic Programming, and this potentially could lead to a time variant mesh size; which can be developed into an adaptive Dynamic Programming algorithm.

A more realistic optimal control approach is also needed. This includes working with more realistic system requirements (such as braking time and vehicle speeds), and the addition of more realistic friction brake models for the system.

Finally the comparison of fully developed optimal control strategies with more realistic classical nonlinear control approaches would be more appropriate to assess the system performance and the benefits of using optimal control for a KERS.

REFERENCES

Assadian, F., Margolis, D. (2001): “Modeling and Control of a V-belt CVT” IFAC Advances in Automotive Control. Karlsruhe, Germany, 2001.

Bellman, R. E. (1962): “Applied Dynamic Programming”. Princeton, N.J., Princeton University Press.

Bellman, R., Kalaba, R. (1965): “Dynamic Programming and Modern Control Theory”. Academic Press. New York and London.

Bolund, B., Bernhoff, H., Leijon, M. (2007): “Flywheel energy and power storage systems”. Renewable and Sustainable Energy Reviews. Vol. 11, pp. 235-258.

Boretti, A. (2010): “Comparison of fuel economies of high efficiency diesel and hydogren engines powering a compact car with a flywheel based kinetic energy recovery systems”. International Journal of hydrogen energy. Vol. 35, pp. 8417-8424.

Boukettaya, G., Krichen, L., Ouali, A. (2010): “A comparative study of three different sensorless vector control strategies for a Flywheel Energy Storage System” Energy. Vol. 35, pp. 132-139.

Brogan, W. (1991): “Modern Control Theory”. 3rd ed., Prentice Hall.

Cao, J., Cao, B., Chen, W., Xu, P. (2007): “Neural Network self-adaptive PID Control for Driving and Regenerative Braking of Electric Vehicle”. Proceedings of the IEEE, International Conference on Automation and Logistics. August 18-21, 2007. Jinan, China.

Cao, B., Bai, Z., Zhang, W. (2005): “Research on Control for Regenerative Braking of Electric Vehicle”. IEEE.

Carbone, G., Mangialardi, L., Bonsen, B., Tursi, C., Veenhuizen, P. A. (2007): “CVT dynamics: Theory and experiments”. Mechanism and Machine Theory. Vol. 42, pp. 409-428.

Cheng, C., Ye, J. (2011): “GA-based neural network for energy recovery system of the electric motorcycle”. Expert Systems with Applications. Vol. 38, pp. 3034-3039.

Cikanek, S. R., Bailey, K. E. (2002): “Regenerative Braking System for a Hybrid Electric Vehicle”. Proceedings of the American Control Conference. Vol. 4, pp. 3129-3134. May 8-10, 2002.

Cross, D., Brockband, C. (2008): “Mechanical Hybrid system comprising a flywheel and CVT for Motorsport and mainstream Automotive applications”. SAE International http://www.torotrak.com/pdfs/tech_papers/2009/SAE_WC_2009_09PFL-0922_KERS.pdf [retrieved 25.9.12]

Dumé, B. (2010): “Graphene supercapacitor break storage record”. http://physicsworld.com/cws/article/news/2010/nov/26/graphene-supercapacitor-breaks-storage-record (retrieved on 27/02/2013)

Elliot, D. (2009): “Bilinear Control Systems – Matrices in Action”. Applied Mathematical Sciences. Vol. 169. Springer.

Fabien, B. (2009): “Energy storage via high-energy density composite flywheels”. University of Washington. Seattle, WA. http://depts.washington.edu/amtas/events/amtas_09spring/Fabien.pdf (retrieved on 26/02/13)

Frei, S., Guzzella, L., Onder, C., Nizzola, C. (2006): “Improved dynamic performance of turbocharged SI engine power trains using clutch actuation”. Control Engineering Practice. Vol. 14, pp. 363-373.

Gauthier, J., Micheau, P. (2012): “Adaptive control of a continuously variable transmission subject to wear”. Control Engineering Practice. Vol. 20, pp. 569-574.

Ghedamsi, K., Aouzellag, D., Berkouk, E. M. (2008): “Control of wind generator associated to a flywheel energy storage system”. Renewable Energy. Vol. 33, pp. 2145-2156.

Guzzella, L., Schmid, A. M. (1995): “Feedback Linearization of Spark-Ignition Engines with Continuously Variable Transmissions”. IEEE Transactions on Control Systems Technology. Vol. 3, No. 1. March 1995.

Haj-Fraj, A., Pfeiffer, F. (2001): “Optimal Control of Gear shift operations in automatic transmissions”. Journal of the Franklin Institute. Vol. 338, pp. 371-390.

Harnoy, A. (2003): “Bearing Design in Machinery-Engineering Tribology and Lubrication”. Marcel Dekker, Inc. New York, N. Y.

Hoon, Y., Sungho, H., Hyunsoo, K. (2006): “Regenerative braking algorithm for a hybrid electric vehicle with CVT ratio control”. Proceedings of the institution of mechanical engineers Part D Journal of Automobile Engineering. Vol. 220, No. 11, pp. 1589-1600.

Hua, L., Jian, Z., Da, X., Xiaojun, M. (2009): “Design for Hybrid Electric Drive system of armored vehicle with two energy storage devices” International Conference on sustainable power generation and supply. Vols. 1-4, pp. 1747-1750.

Inoue, K., Ogata, K., Kato, T. (2010): “Efficient Power Regeneration and Drive of an Induction Motor by Means of Optimal Torque Derived by the Variational Method”. Electrical Engineering in Japan. Vol. 173, No. 1, pp. 41-50.

Karden, E., Ploumen, S., Fricke, B., Miller, T., Snyder, T. (2007): “Energy storage devices for future hybrid electric vehicles”. Journal of Power Sources, Vol. 168, pp. 2-11.

Kiencke, U., Nielsen, L. (2005): “Automotive Control Systems For Engine, Driveline and Vehicle”. 2nd ed., Springer, Berlin; Heidelberg; New York. Germany.

Kirk, D. E. (1998): “Optimal Control Theory: An Introduction”. Prentice Hall.

Kong, L., Parker, R. G. (2008): “Steady mechanics of layered, multi-band belt drives used in continuously variable transmissions (CVT)”. Mechanism and Machine Theory. Vol. 43, pp. 171-185.

Lechner, G., Naunheimer, H. (1999): “Automotive Transmissions-Fundamentals, Selection, Design and Application”. Springer, Berlin; London.

Liu, H., Jiang, J. (2007): “Flywheel energy storage-An upswing technology for energy sustainability”. Energy and Buildings. Vol. 39, pp. 559-604.

Liu, J., Zhou, Y., Cai, Y., Su, J. (2007): “The Application of Generalized Predictive Control in CVT Speed Ratio Control”. Proceedings of the IEEE. August 18-21. Jinan, China.

Lu, X., Khonsari, M. M. (2005): “On the lift-off speed in journal bearings”. Tribology Letters. Vol. 20, pp. 299-305. Dec. 2005.

Lukic, S. M., Bansal, R. C., Emadi, A. (2008): “Energy Storage Systems for Automotive Applications”. IEEE Transactions on Industrial Electronics, Vol. 55, No 6, pp. 2258-2267, June 2008.

Mangialardi, L., Mantriota, G. (1999): “Power flows and efficiency in infinitely variable transmissions”. Mechanism and Machine Theory. Vol. 34, pp. 973-994.

Mensler, M., Joe, S., Kawabe, T. (2006): “Identification of a toroidal continuously variable transmission using continuous-time system identification methods”. Control Engineering Practice. Vol. 14, pp. 45-58.

Mukhitdinov, A. A., Ruzimov, S. K., Eshkabilov, S. L. (2006): “Optimal Control Strategies for CVT of the HEV during regenerative process. IEEE Conference on Electric and Hybrid vehicles. ICEHV ’06. Pp. 1-12.

Müller, C., Schröder, D. (2001): “Modeling and Control of Continously Variable Transmissions”. IFAC Advances in Automotive Control. Karlsruhe, Germany, March 28-30, 2001. Pp. 79-84.

O’Neil, P. (1995): “Advanced Engineering Mathematics”. 4th ed., International Thomson Publishing.

Ogata, K. (2002): “Modern Control Engineering”. 4th ed., Prentice Hall.

Osumi, T., Ueda, K., Nobumoto, H., Sakaki, M., Fukuma, T. (2002): “Transient analysis of geared neutral type half-toroidal CVT”. JSAE Review. Vol. 23, pp. 49-53.

Pérez, L., Bossio, G., Moire, D., García, G. (2006): “Optimization of power management in an hybrid electric vehicle using Dynamic Programming”. Mathematics and Computers in Simulation. Vol. 73, pp. 244-254.

Pérez, L., Pilotta, E. A. (2007): “Optimal power split in a hybrid electric vehicle using direct transcription of an optimal control problem” Mathematics and Computers in Simulation.

Pfiffner, R., Guzzella, L., and Onder, C.H. (2003): “Fuel-optimal control of CVT powertrains”. Control Engineering Practice. Vol 11, pp 329-336.

Pinch, E. R. (1993): “Optimal Control and the Calculus of Variations”. Oxford University Press.

Powell, B., Zhang, X., and Baraszu, R. (2000): “Computer Model for a Parallel Hybrid Electric Vehicle (PHEV) with CVT”, Proceedings of the American Control Conference, Chicago, Illionis, June 2000. Pp 1011-1015.

Samineni, S., Johnson, B. K., Hess, H. L., Law, J. D. (2003): “Modeling and Analysis of a Flywheel Energy Storage System with a Power Converter Interface”. International Conference on Power Systems Transients. IPST 2003. New Orleans, USA.

Schweitzer, G. (2002): “Active magnetic bearings – chances and limitations” International Centre for Magnetic Bearings, ETH Zurich. http://www.mcgs.ch/web-content/AMB-chances_and_limit.pdf [retrieved 25.9.12]

Setlur, P., Wagner, J. R., Dawson, D. M., and Samuels, B. (2003): “Nonlinear Control of a Continuously Variable Transmission (CVT)”, IEEE Transactions on Control Systems Technology, Vol. 11, pp. 101-108.

Song, X., Zulkefli, M., Sun, Z., Miao, H. (2011): “Automotive Transmission Clutch Fill Control Using a Customized Dynamic Programming Method”. ASME Transactions on Journal of Dynamic Systems, Measurement, and Control, Vol. 133, 054503, September 2011.

Srivastava, N., Haque, I. (2008): “Transient dynamics of metal V-belt CVT: Effects of band pack slip and friction characteristic”. Mechanism and Machine Theory. Vol. 43, pp. 459-479.

Van Mierlo, J., Van den Bossche, P., Maggetto, G. (2004): “Models of energy sources for EV and HEV: fuel cells, batteries, ultracapacitors, flywheels and engine-generators”. Journal of Power Sources. Vol. 128, pp. 76-89.

Wicks, F., Donnelly, K. (1997): “Modeling Regenerative Braking and Storage for Vehicles”. Proceedings of the 32nd Intersociety Energy Conversion Engineering Conference. Vol. 3, pp. 2030-2035.

Won, J., Langari, R., Ehsani, M. (2005): “An Energy Management and Charge Sustaining Strategy for a Parallel Hybrid Vehicle with CVT”. IEEE Transactions on Control Systems Technology. Vol. 13, No. 2, pp. 313-320.

Yang, Y., Liu, J., Hu, T. (2011): “An energy management system for a directly – driven electric scooter”. Energy Conversion and Management. Vol. 52, pp. 621-629.

Ye, M., Jiao, S., Cao, B. (2010): “Energy Recovery for the Main and Auxiliary Sources of Electric Vehicles”. Energies. Vol. 3, pp. 1673-1690. 8th October 2010.

Youmin, W., Penghuang, C. (2009): “The Optimal test PID control for CVT Control System”. Proceedings 2009 IEEE International Conferences on Intelligence and Intelligent Systems: 1-5.

APPENDIX A

Algorithms for numerical integration of Ordinary Differential Equations

Here three algorithms for numerical integration of ODEs are summarised. These algorithms are used in the optimal control application using Dynamic Programming. The algorithms are taken from O'Neil (1995). The general initial value problem is to integrate the first order system of ODEs defined as:

$$\dot{\bar{y}} = f(\bar{x}, \bar{y}) \tag{A.1}$$

With initial conditions $y(x_0) = y_0$. Where the independent variable x, dependent variable y are in general vector quantities which covers a system of differential equations.

Euler Method

The definition of y_{k+1} in terms of y_k is given as:

$$y_{k+1} = y_k + hf(x_k, y_k) \tag{A.2}$$

for $k = 0,\ \ 1,\ \ 2,\ \ ...,\ \ n-1$, and $y(x_0) = y_0$

Modified Euler Method

The definition of y_{k+1} in terms of y_k is given as:

$$y_{k+1} = y_k + hf\left(x_k + \frac{h}{2}, y_k + \frac{hf_k}{2}\right) \tag{A.3}$$

for $k = 0,\ \ 1,\ \ 2,\ \ ...,\ \ n-1$, and $y(x_0) = y_0$

Runge-Kutta Method (RK4)

The definition of y_{k+1} in terms of y_k is given as:

$$y_{k+1} = y_k + \frac{1}{6} h[W_{k1} + 2W_{k2} + 2W_{k3} + W_{k4}] \tag{A.4}$$

where h represents the increments of the independent variable and the terms are given as follows, for the first term:

$$W_{k1} = f_k \tag{A.5}$$

for the second term

$$W_{k2} = f\left(x_k + \frac{h}{2}, y_k + \frac{hW_{k1}}{2} \right) \tag{A.6}$$

for the third term

$$W_{k3} = f\left(x_k + \frac{h}{2}, y_k + \frac{hW_{k2}}{2} \right) \tag{A.7}$$

finally, for the fourth term

$$W_{k4} = f\left(x_k + h, y_k + hW_{k3} \right) \tag{A.8}$$

for $k = 0,\ \ 1,\ \ 2,\ \ ...,\ \ n-1$, and $y(x_0) = y_0$

End Appendix A