OPTIMAL CONTROL OF AN AUTONOMOUS WHEEL LOADER

A Thesis

by

SAI SUDEEP REDDY MANDIPALLI

Submitted to the Graduate and Professional School of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Xingyong Song
Committee Members,	Won-Jong Kim
	Yanling Chang
Head of Department,	Guillermo Aguilar

May 2022

Major Subject: Mechanical Engineering

ABSTRACT

Wheel loaders, one of the heavy equipment machinery are widely used in Construction and Mining industries. They typically transport loads over shorter distances. Due to their limited capacity, they have to perform repetitive loading and unloading operations which is called a Short Loading Cycle (SLC). Considering the total energy consumed by heavy machinery, there is a lot of scope to optimize the energy consumed by a wheel loader in an SLC. To analyze the same, a comprehensive control-oriented model of a wheel loader is developed which comprises complex mechanical subsystems such as Engine, Hydraulics, Steering, and Transmission, etc. For optimal fuel consumption, an SLC can be optimized using optimal control techniques such as Dynamic Programming (DP) and numerical optimal control approaches such as Indirect Methods (IMs) or Direct Methods (DMs).

DP is an optimal control technique that guarantees global optimum and yields a global control policy over state and control space. It is preferred for lower-order systems as it is computationally expensive. DMs are computationally efficient for higher-order systems. However, they don't guarantee global minimum and convergence and only provide an optimal open-loop control policy which may not be useful for stochastic systems.

In this thesis, optimization of an SLC of a WL is performed using DP and DM, and their merits and demerits are compared using different metrics namely computational effort, optimal cost, convergence, etc. A customized DP method is proposed to solve the higher-order Optimal Control Problem (OCP) of a WL to address the issue of the curse of dimensionality (COD) and its result is compared to that of DM.

DEDICATION

To my mom, dad and my sister.

ACKNOWLEDGEMENTS

I would like to thank my committee chair, Dr. Song, Xingyong and committee members, Dr. Kim, Won-Jong and Dr. Chang, Yanling for their continuous guidance and support. I would like to specifically thank Dr. Song, Xingyong for being patient with me. I sincerely appreciate his valuable insights and a logical approach towards a problem. I am grateful for being his first masters student.

I would like to thank Dr. Tohid, Sardarmehni for his inputs. I would also like to thank my lab mates for their support.

I would like to express my sincere gratitude to Prof. Golla, Michael for having me as his teaching assistant.

Last but not least, I would like to thank my parents, my sister and my relatives for everything. I would also like to thank my friends, department faculty and staff for an enlightening and a joyful experience at Texas A&M University.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supervised by the thesis committee consisting of Dr. Song, Xingyong (advisor) and Dr. Kim, Won-Jong of the Department of Mechanical Engineering and Dr. Chang, Yanling of the Department of Engineering Technology and Industrial Distribution. All other work conducted for the thesis was completed independently.

Funding Sources

Graduate study was supported through Teaching Assistantship provided by Professor Michael Golla, Department of Engineering Technology and Industrial Distribution.

NOMENCLATURE

ω_e	State variable, Engine speed (rad/s)
V	State variable, Vehicle speed (m/s)
S	State variable, Vehicle displacement (m)
S_{buc}	State variable, Bucket position of wheel loader (m)
V_{buc}	State variable, Bucket velocity (m/s)
X	State variable, X-coordinate of wheel loader (m)
Y	State variable, Y-coordinate of wheel loader (m)
β	State variable, Heading Angle (rad)
δ	State variable, Steering Angle (rad)
U_{mf}	Control input, Fuel mass injection/cycle (mg)
U_{brake}	Control input, Braking force (N)
U_{st}	Control input, Steering input
U_{ab}	Control input, Bucket acceleration (m/s^2)
T_e	Engine Torque $(N-m)$
$\dot{m}_f$	Rate of fuel consumption $(ltrs/s)$
γ	Gear Ratio
r_w	Wheel Radius (m)
P_{load}	Power consumed by the auxiliary components (W)
P_{pump}	Power at the pump side of the torque converter (W)
P_{turb}	Power at the turbine side of the torque converter (W)
P_{steer}	Power consumed by Steering system (W)
P_{lift}	Power consumed by Hydraulic system (W)

P_{trac}	Power consumed in Traction (W)
J_e	Engine Inertia $(kg\, m^2)$
P_{im}	Intake manifold pressure (N/m^2)
F_{trac}	Traction force at wheels (N)
c_r	Rolling resistance coefficient
C_{str}	Steering power coefficient
F_{roll}	Rolling resistance force (N)
M_{total}	Total Mass of the wheel loader (Kg)
M_{load}	Mass of the bucket with load (Kg)
R_{turn}	Turning radius of the wheel loader (m)

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

1. INTRODUCTION AND LITERATURE REVIEW

1.1 Motivation

Mining and Construction industries account for approximately 20% of the energy consumption in the United States [7]. The mining industry uses the heaviest machinery on the planet. The mining industry has to get rid of overburden, material that lies above coal to extract coal from mines. In most countries, coal and mineral ore deposits close to the surface are mined and are not available anymore. Unfortunately, the energy demand has enormously increased and the mining industry has to search deeper into the earth for fossil fuels to meet the rising demands. The construction industry also consumes a massive amount of energy to prepare sites, transport materials, and build structures. All the work mentioned above is performed by either heavy equipment or construction vehicles. Hence, both industries consume a tremendous amount of resources to excavate materials and transport them. Heavy machinery in both industries consume a lot of energy [8] and they often deal with harsh environments such as dust, hard cut-resistant rocks, and loose sand with less ground bearing pressure. They can be very inefficient in operation and can waste a lot of fuel if not well-maintained.

Heavy Earth Moving Machinery (HEMM) such as Dumpers, excavators, wheel loaders (WLs), etc play an important role in the construction and mining industries. They are responsible for loading and transporting material from the working site to the power generators such as power plants, steel industries, etc. Dumpers or load trucks are used to transport materials over a long distance, excavators are used for loading-unloading the material, and WL for both purposes. WLs are used for transportation in workplaces and not for long-range transportation. Excavators and WLs perform similar jobs i.e load and unload. However, shovels have high bucket capacity and are very less mobile. On the other hand, WLs have less bucket capacity and hence have to be more mobile to meet the productivity standards. WLs are the main loading and unloading machines deployed in open-cast mines.

HEMMs across the world consume a lot of fuel delivering the needs of people and industries. Unfortunately, the source of energy is fossil fuels which are not perennial. Some of the machinery operate on electricity but a majority of them use diesel as the primary energy source. They operate continuously throughout the day (14 hrs) for 200 days per year. The majority of the equipment performs a repetitive operation cycle and a minor change in its operation can save a significant amount of time and fuel. Optimization of a cycle of operation of these equipment has been a primary interest for researchers and in the industry. Apart from fuel and time savings, the environment poses several safety hazards to both machines and humans. Rough terrain with a lot of bumps and lack of proper vision in a dusty environment lead to accidents (Toppling of machines and health hazards). Technological advancements in computational power and Artificial Intelligence (AI) are driving the dream of self-driving vehicles towards a reality. Having Autonomous off-road vehicles can significantly reduce these threats coupled with energy savings. Mining and construction related health hazards can be minimized with automation at work sites.

1.2 Literature Review

WLs are indispensable and are the most common type of heavy machinery deployed in the mining and construction industries. WL is typically used to lift a pile of material from a heap and unload it onto a dumper or load truck. Due to the limited capacity of a WL, it has to perform the loading-unloading cycles repetitively. A WL typically has a mass of 30 tons and it is ideally preferred to have a low center of gravity. The center of gravity plays an important role in determining the stability of the vehicle. A typical off-road environment is rough terrain with road bumps or a sandy road with less ground bearing pressure. The integrity of the ground determines the stability of the vehicle. While the operator is moving the bucket up, the center of gravity of the vehicle shifts upward and a bump can drive the vehicle unstable. Hence, WLs are typically slow compared to the on-road vehicles. For on-road vehicles, EPA tests a series of driving routines and prescribes a cycle for better fuel economy. Unfortunately, off-road environments are not generic and it is often difficult to prescribe an operational cycle. Different variants of operating cycles are followed by operators at mining sites. One such cycle of operation is defined as a Short

Loading Cycle (SLC). An SLC is shown in Fig.1.1. It has four phases of operation. It typically starts with the WL with a fully loaded bucket at its lowest position. The first phase is the vehicle accelerating in reverse. The second phase is braking where the vehicle comes to halt. The third phase is accelerating forward followed by braking in the fourth phase to stop at its destination. The bucket will be at the required height to dump the material onto a Dumper or a load truck by the end of phase-4. Manufacturers and owners of a WL are often interested in performing the cycle in minimum possible time with minimum possible fuel. Minimum time means more acceleration and needs more force at the wheels which demand high power requirements and hence high fuel. On the other hand, minimum fuel means less torque generation meaning less force at the wheels and the vehicle takes more time to reach its destination. Due to the contradictory nature of both the goals, one has to settle for a trade-off between the two. Construction and mining sites where a WL is deployed are often dusty and lack proper vision. The sensors have to be very robust and should be able to reject disturbances effectively. It is very expensive to mount robust sensors to take measurements and analyze the system. Data-driven approaches can also be useful but using neural networks and other machine learning algorithms for optimization requires a lot of data in various conditions. Considering the environment the WL operates being dusty, remote and bumpy terrains, obtaining data from sensors becomes more expensive and also noisy. A mathematical model which can considerably capture the dynamics of the cycle operation can save a lot of time and resources than physically capturing the data every time.

Several studies have been performed on analyzing the performance of a WL in an SLC. A systematic approach has been outlined in [9]. It outlines a strategy to minimize fuel in a power-split hybrid WL. The DP-based approach is used for optimization but only drive train and hydraulic system are considered and engine dynamics, steering are neglected. Path planning for a bucket trajectory while scooping has been discussed in [10], [11]. Path planning analysis for a WL has been outlined in [12]. In this paper, DP is used to optimize the trajectory traversed by the WL. However, it only focuses on path planning and not the integrated system. A linear model of a vehicle power train is considered in [13] and Multi-Input, Multi-Output robust controllers such

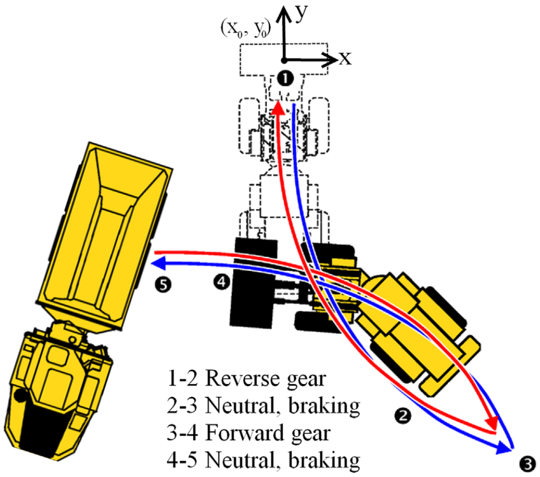


Figure 1.1: Trajectory of a WL with four phases of operation. Picture from [1]

as H_2 and H_∞ are designed. However, the optimality of the result is limited by the linearity assumption. Nonlinear dynamics, trajectory optimization, and vehicle dynamics are ignored in this paper. DP and Pontryagin Maximum Principle (PMP) based optimization are outlined in [2]. In this paper, engine dynamics are considered for minimizing fuel consumption based on torque requirements from subsystems. Subsequently, vehicle dynamics, lift, and tilt dynamics are ignored. Research on finding a global optimum for a WL SLC is outlined in [14]. This paper focuses on scooping and lift-tilt dynamics and optimizing the trajectory based on the DP approach. Engine, vehicle dynamics, and the steering system are not included in this paper. An integrated approach of including all the major subsystems such as Hydraulics, Engine, Drive train, Steering has been outlined in [4]. A comprehensive model of WL with all the subsystems has been developed and a numerical solver (PROPT) has been used to solve the OCP. This software developed in MATLAB uses pseudo-spectral collocation methods to discretize the states and converts an OCP into a Non-Linear Program (NLP) and Sparse Nonlinear OPTimizer (SNOPT) algorithm is used to solve the resulting NLP problem. However, SNOPT is a numerical solver which guarantees local optimum and not global optimum and may have convergence issues. Other works such as [15] and [16] include all the major subsystems and owing to the order of the systems being too high and due to non-linearities and discontinuities, authors have preferred numerical solvers instead of dynamic programming, and indirect methods are also ignored.

In this thesis, an SLC is considered for optimization. A WL consists of highly complex systems such as engine, transmission, steering, hydraulics, etc. The engine is the prime mover which generates power and three subsystems that consume power are Traction, Hydraulics and Steering. Traction can be related to longitudinal motion of the WL, Hydraulics to vertical motion of the bucket, and Steering to the Wl's lateral motion. Modeling of such systems poses several challenges such as high non-linearity and being highly parametric. Such models are usually of very high order and designing a controller is extremely difficult owing to high computational requirements and convergence issues. Hence, from a modeling and control perspective, the mathematical model should be of lower order and should also considerably capture the entire system dynamics. A lot of human factors and environmental factors such as dust and obstacles and uncertainties also play an important role but they are ignored.

1.3 Optimization Techniques

Some of the widely used techniques to solve an Optimal Control Problem (OCP) are Dynamic Programming (DP) [17] and numerical approaches [18], [19],[20]. Numerical techniques for solving OCPs are typically classified into two categories namely direct and indirect methods.

1.3.1 Indirect Methods

Indirect methods are one of the oldest methods. They are based on the calculus of variations. They utilize the maximum principle to derive necessary conditions of optimality, the adjoint equations and a Hamiltonian maximization condition, and the boundary conditions, thus converting an OCP into a boundary value problem. This boundary value problem can be solved by shooting or discretization methods. Some of the advantages and disadvantages of indirect methods are outlined below:

• The optimality conditions derived using the maximum principle are necessary but not sufficient conditions. Hence the solution obtained is locally optimal and may not be globally optimal. Since they discretize the solution after the optimization routine, they provide highly accurate solutions and are typically used in aerospace applications. They are better than DP and direct methods as they provide better error estimates and do not suffer from discretization errors [21], [22], [19].

• They are based on differential calculus and as the approach is analytical, it is often difficult to solve especially higher-order nonlinear systems.

• They require an initial guess for the adjoint variables to solve the necessary conditions of optimality. This is often difficult if the adjoint equations are high in number. Shooting methods suffer from difficulties in finding appropriate initial guess for the adjoint variables [21], [22], [19].

• They have a smaller domain of convergence [21].

• It is difficult to handle path constraints as an initial guess has to be provided for the sequence of constrained/unconstrained arcs of a solution trajectory which is often very hard [21], [22], [19].

1.3.2 Direct Methods

Direct methods, on the other hand, do not rely on maximum principle and directly optimize the objective function without the formulation of necessary conditions. This has a two-step approach – one, transcribing a continuous infinite-dimensional OCP into a discrete finite-dimensional NLP, and two, solving the nonlinear program using existing NLP algorithms such as Sequential Quadratic Programming (SQP) or Interior Point Methods (IPM). Several numerical NLP solvers exist such as SNOPT, IPOPT, KNITRO etc [23], [19].

Transcription is usually performed using direct collocation methods or direct shooting methods. Shooting methods are not practically reliable for higher-order systems. In a direct collocation method, differential equations are discretized using finite difference methods with a fixed or variable time steps. In between those time steps, time-dependent functions (linear or polynomial) are used to approximate the solution. These are local collocation or orthogonal collocation methods. Algorithms typically used for discretization are explicit or implicit schemes. Explicit methods are first-order accurate and computationally very efficient but face stability issues if the integration time step is higher than required. Implicit schemes such as Range Kutta (RK4) and Hermite Simpson methods are fourth-order accurate and hence computationally expensive compared to explicit methods. Global or Pseudo-spectral collocation methods involve approximating the state using a global polynomial and performing collocation at chosen points. Global methods converge at an exponential rate as a function of collocation points [19], [20], [21], [18]. The transcribed OCP in the form of an NLP can be solved using algorithms such as SQP, IPM. Several open-source and closed source software exist such as CasAdi [24], PROPT [25], GPOPS – II [26], DIDO [27] etc. More information about the solvers can be found in a survey papers [19], [22]. Some of the advantages and disadvantages of direct methods are listed as follows:

• They have a larger domain of convergence compared to indirect methods [21].

• They also provide a local solution as indirect methods but have larger leverage on the initial guess compared to indirect methods. The optimal solution is also deterministic i.e the optimal control policy obtained is a single trajectory and not a global policy. In other words, the solution is open-loop and may not be useful for stochastic systems with noise and uncertainties [21]. Methods that guarantee a global optimum have been a subject of interest among researchers. Some of them include multi-start approach, Simulated Annealing (SA)[28], Evolutionary Algorithms (EA)[29], Branch and Bound (B&B), Tabu search, etc. The Multi-start approach is a conventional optimization with different initial guesses. Solving an OCP using this approach is a typical manifestation of the curse of dimensionality (COD). SA is a stochastic method. It also has a trial and error component and its performance is problem specific. It also has a trade-off between convergence time and locality of the search[30]. EA are stochastic optimization techniques based on ideas of natural evolution. They also suffer from the trade-off as mentioned above. Likewise, B&B and Tabu search methods have their drawbacks. No method can guarantee global optimum at least in a finite time. A survey on methods to obtain global optimum in non-linear optimization can be found in [31].

• These methods require smooth objective and constraint functions, differentiable at least twice. Direct methods are gradient-based methods that require computing first and second-order derivatives which can be done analytically or numerically. Analytical derivatives are exact and yield faster optimization results. However, it is practically not feasible to use symbolic differentiation in software development and hence numerical techniques are to be relied upon. Numerical derivatives mostly use forward or central finite difference schemes and often are not accurate for highly nonlinear systems and also require the time step to be small enough. These factors play an important role in the convergence or divergence of the solvers [19].

• Initial guess sensitivity: A decent initial guess is often necessary for the solution to converge to a local minimum or even to a feasible solution. Interior point methods require the initial guess to be strictly feasible which is difficult to provide. Some random component is observed in [32] convergence with relevance to GPOPS – II [26]. It is well known that different initial guesses lead to different local extremum.

• Real-Time Implementation: In the industry, lookup tables are often used rather than analytical functions. Calculating gradients is often tedious and using a numerical approach to calculate gradients may not be accurate and often requires higher sampling for them to be accurate which is practically challenging [14].

1.3.3 Dynamic Programming

Dynamic Programming: It is a recursive technique employed to solve Bellman's principle of optimality [17]. The solution obtained is a sequence of decisions that yields a global optimal policy and a trajectory. If there exists an optimal sequence of decisions, then every subsequence must also be optimal. For continuous systems, the DP algorithm boils down to solving a partial differential equation, the Hamilton Jacobi Bellman (HJB) equation. If a solution exists, then it is guaranteed to be a global optimum. However, an analytical solution exists only for a limited class of OCPs that are linear or close to being linear which is not the case with most practical situations. Hence, the solution has to be found numerically and a discrete DP algorithm is used [33]. Sampling of state, control and time space remains a challenge as higher sampling means high accuracy but high computational cost. Lower sampling leads to less accurate solutions and may not yield a global optimum at all. Integration with respect to time poses the same challenge as seen in direct methods that explicit methods are fast and but less stable and accurate than implicit methods. A trade-off between computational cost vs accuracy means that DP is mostly effective only for lower dimensional systems. Some of the advantages and disadvantages of DP are outlined as follows:

• Dynamic Programming finds the optimal solution by performing an exhaustive search through the entire state and control space. Hence, the result obtained will be a global optimum if it exists.

• The sequence of decisions obtained from DP is dependent on the current state of the system and hence, the control solution is a feedback control or closed-loop control. In practice, there are model/parameter uncertainties, noise in sensors, etc. Any system in practice is stochastic and DP can handle stochastic aspects of the control problem [21].

• Curse of Dimensionality (COD): Dynamic Programming being discrete suffers from the COD. A higher sampling of the state-space system leads to a higher computational load and increases exponentially. The order of computations depends exponentially on the number of states and control variables. Hence, DP is computationally feasible for lower-order optimal control systems [21].

Considering the merits and demerits of the above approaches, indirect methods are ignored as the problem dealt with in this analysis is of higher-order and has path constraints and it also relies heavily on initial guess for the adjoint states and also has a smaller domain of convergence. For this thesis, direct methods and dynamic programming are explored with relevance to the WL problem discussed in the modeling section. The direct method has already been successfully applied on the fully integrated WL system in [4]. The authors have ignored dynamic programming citing the COD as the reason. However, considering the merits of DP over direct methods, we aim to explore the trade-offs between these algorithms in the context of optimization of a WL system. Considering the challenges posed by direct methods such as convergence issues, local minimum solutions, initial guess sensitivity, and deterministic solution or open-loop control solution, we consider exploring DP to solve the system as it does not face any of the issues mentioned above. However, the major challenge is to overcome the COD as the order of the system is high and is not computationally feasible as-is and the model has to be modified to account for the same. Considering the assumptions and modifications, it remains to be seen how DP fares against direct methods because the solution obtained from DP may not be global owing to the assumptions/ modifications to be discussed in section 4.

Dynamic Programming for the fully integrated system results in an OCP of a higher-order system and is not computationally feasible. Two approaches to resolving this issue could be to have a Reduced Order Model (ROM) or a split stage. Since all the state equations are not independent, ROM can be a solution and will also be explored.

ROM has been a powerful tool in computational fluid dynamics and the Finite Element Method (FEM). ROM works on the premise that the states are correlated as we see in continuous systems. For discrete systems, studies on reduced order modeling have not gained enough attention. The feasibility of a ROM of the entire system is investigated. Several reduced-order methodologies are investigated such as Proper Orthogonal Decomposition (POD) and Smooth Orthogonal Decomposition (SOD) and Discrete Empirical Interpolation Method (DEIM). All the methods try to find a fundamental basis of all the states and the states can later be retrieved by the basis. The basis is usually of a lower order than the system order. However, a basis is of a lower order only if there is a strong correlation between states. WL subsystems are usually independent and do not have a strong correlation between them. Reduction of system order is not beneficial as without correlation the accuracy drops significantly with reduced order. With a system of high nonlinearities and discontinuities, a reduced-order approach may render a WL system unstable due to high inaccuracies.

With a split-stage optimization approach, the integrated system is split into two subsystems and an optimization routine is performed on both of the systems independently and later the optimal policies obtained from both the routines are integrated using a flexible approach to obtain the final solution. Keeping the inaccuracies of ROM in mind, the split-stage approach is found to be more suitable for this application and is chosen for this thesis.

A block diagram of integrated and split-stage optimization approaches is shown in Fig.3.2 in chapter III. Integrated OCP suffers from the COD with the order of the system being high. In a split-stage approach, the full order OCP is split into three stages - path planning, hydraulics optimization and integration of both the stages. This reduces the number of operations at each time step by a factor of 1000 depending upon the coarseness of the discretization. The first stage focuses on path planning i.e an optimal trajectory for the WL to traverse to a desired final destination is found without considering hydraulic power consumption. This stage gives us the distance that the WL has to traverse in order to reach the final position. The final distance obtained from stage - 1 becomes a constraint in stage - 2 optimization which finds the optimal hydraulic power consumption profile while obeying the distance constraint set from stage - 1. Finally, the stage - 3 integrates both the above stages and provides an optimal state and control profile. This approach is discussed in detail in chapter 3.

The novelty of this thesis is the Split-Stage Optimization strategy, a customized DP based approach which reduces the computational complexity significantly and enables the application of DP to solve a higher order WL Optimal Control Problem to be more computationally feasible while not compromising the optimality of the solution. The formulated OCP is solved using this novel strategy and compared to the optimal solution obtained from the DM. This thesis also explores the initial guess sensitivity and convergence issues encountered while solving the OCP using the DM.

1.4 Thesis Overview

This thesis targets to explore the merits and demerits of Dynamic Programming and Direct Methods of optimal control in the context of autonomous construction machinery, WL to be specific. A customized Dynamic Programming method is also proposed to address the issue of the COD. This thesis is divided into 5 main chapters.

Chapter 2 focuses on developing a mathematical model to represent the typical operation of a WL in the mining and construction sites. Each subsystem is analysed and modeled in detail separately.

Chapter 3 formally introduces the optimal control problem with an objective function, state dynamics and constraints etc. It also briefly outlines the approach to the OCP with Dynamic Programming.

Chapter 4 discusses a DP based approach to solve the OCP in detail. The formulated OCP is solved using DP and the results are presented and analysed in detail.

Chapter 5 discusses the direct numerical approach to solve the COP in detail. The OCP is solved using an open-source solver namely MPOPT (Multi-Phase OPTmizer). The results are presented and various factors influencing the results are discussed and analysed in detail.

Finally, Chapter 6 is about comparing and analysing the results obtained from both the methods and their advantages and disadvantages in the context of the current OCP are thoroughly discussed in detail. This chapter also details the contribution and and a novel approach to solve higher order OCPs using DP. It concludes this phase of research and proposes possible research ideas to explore in future.

Wheel loaders have complex mechanical components such as a turbocharged diesel engine, hydraulic system for lift and tilt mechanism of the bucket, torque converter, and vehicle's lateral and longitudinal dynamics. A mathematical representation of these components is highly parametric and non-linear. Designing a controller for such a mathematical model is usually difficult because of its high order and non-linearity. Hence, a control-oriented model preferably low in order, and can significantly capture the dynamics of all the systems is preferred. The majority of the modeling work is borrowed from [4]. Fig.2.1 shows the overview of all the subsystems constituting a wheel loader.

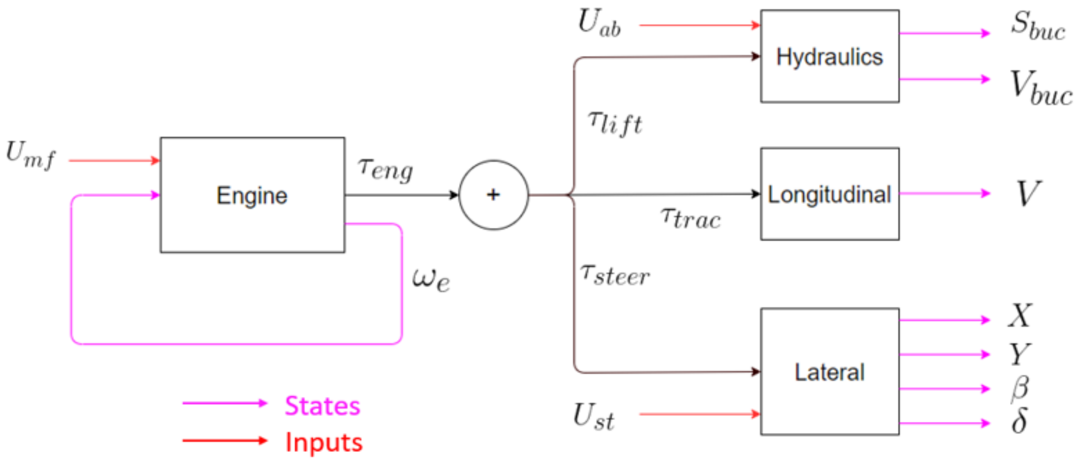


Figure 2.1: Component systems in a wheel loader

2.1 Engine Modeling

A turbocharged/supercharged diesel engine model consists of several sub-models of components such as air filter, compressor, throttle, turbine, etc which is usually a higher order nonlinear model and is also highly parametric. A validated mean value model of an engine developed in

MATLAB/Simulink is available at [2] with experimental data from [34]. Considering optimal control, a control-oriented engine model is developed which is lower in order and can significantly capture the engine dynamics. (T_e), engine torque is assumed to be a function of engine speed (W_e), fuel injection per cycle (U_{mf}) and intake manifold pressure (P_{im}). Training data is obtained from the simulink model for a wide range of inputs of varying pulse-width in steps of $0.1s$, $0.5s$, and $1.5s$ and the simulation was run for $300s$. The generated data is used to fit a linear regression model with basis function $T_e : \mathbb{R}^3 \to \mathbb{R}$.

$$\hat{T}_e(\hat{\omega}_e, \hat{P}_{im}, \hat{U}_{mf}) = W^T[1; \hat{\omega}_e; \hat{\omega}_e{}^2; \hat{U}_{mf}; \hat{P}_{im}]$$

Here $W^T = [w_1, w_2, w_3, w_4, w_5]$ denotes the transposed vector of the weights of variables. The Root Mean Square Error (RMSE) of normalized torque between the predicted and actual data is found to be about $1.71e-2$ for the training data and $1.68e-2$ for the test data. The RMSE for the model with intake manifold pressure variable is $9.5e-3$. The difference between both the above cases with relevance to the inclusion of P_{im} is very little. Hence, the model is assumed to be independent of intake manifold pressure. This means that the torque loss because of the low intake manifold pressure is ignored. Fig.2.2 shows the comparison of the predicted and actual data.

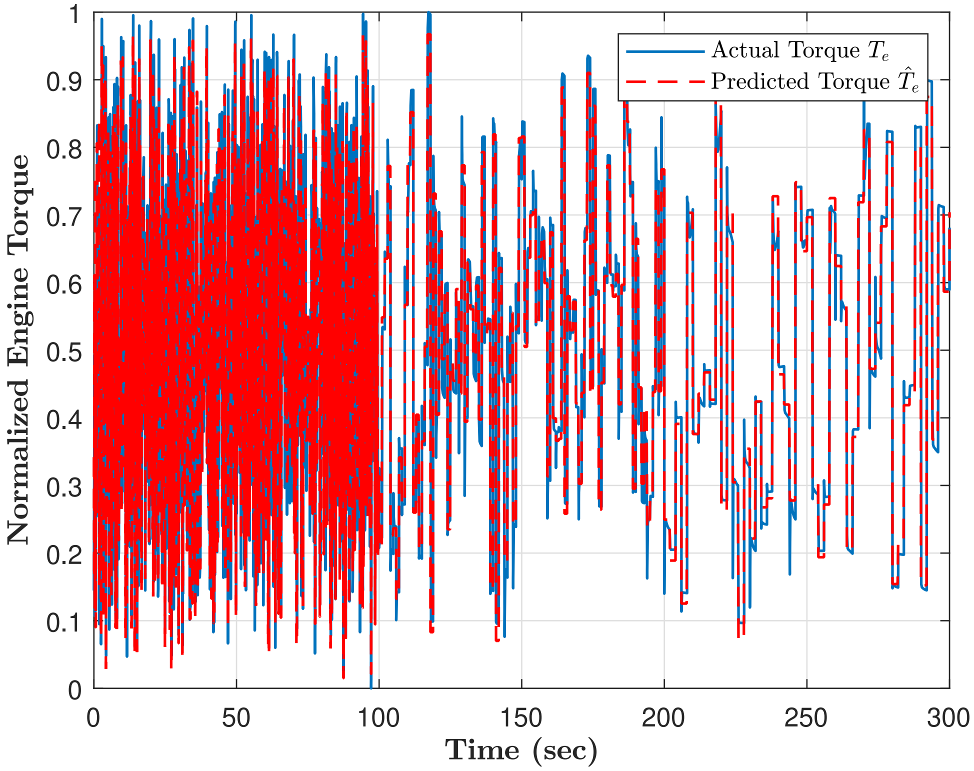


Figure 2.2: Normalized Predicted and Actual (validated Simulink model [2]) Engine Torque for test data.

The generated model has a parabolic profile (concave down) with engine speed ω_e and has a positive correlation with U_{mf}. In practice, an engine cannot generate enough torque at lower engine speeds even with high fuel injection. This is also true when the engine is operating at high speeds with high fuel injection. Considering the polynomial nature of the function, it does not perform well at both ends of the engine speed. This can be addressed by introducing a limiting torque profile for the generated model. The limiting profile is borrowed from [3]. Fig.2.3 shows the engine model torque and the limiting profile obtained from [3]. The engine speed dynamics depend on the torque generated and torque utilized by subsystems such as hydraulics, traction, etc.

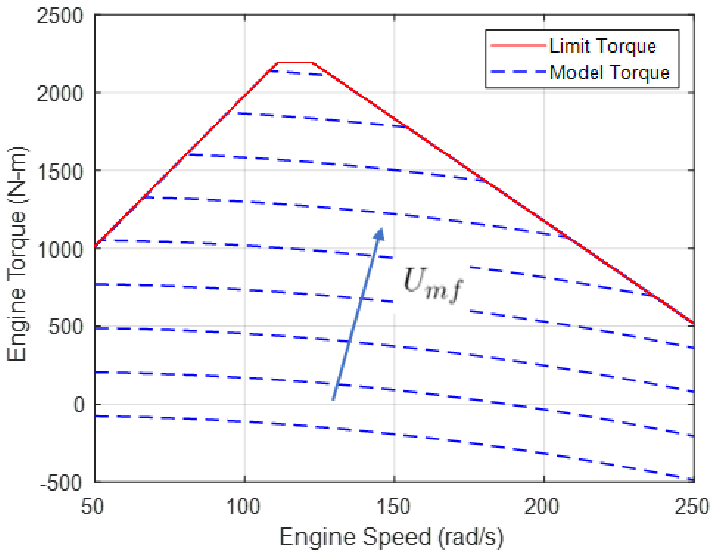


Figure 2.3: Engine model Torque and the limiting Torque profile [3].

$$\frac{d\omega_e}{dt} = \frac{1}{J_e}(T_e - \frac{P_{load}}{\omega_e}) \tag{2.1}$$

where $P_{load} = P_{trac} + P_{lift} + P_{steer}$ represents the sum of torque requirements from all the subsystems assuming all the subsystems are connected to the crank shaft.

The rate of fuel consumed by the engine (6 cylinder) is calculated as

$$\dot{m_f} = \frac{3\ *\ 10^{-6}\, U_{mf}\, \omega_e}{8\pi} \tag{2.2}$$

2.1.1 Driveline

A typical off-road vehicle driveline is modeled with a torque converter, gearbox, and wheels. A wheel loader is an off-road vehicle that usually traverses on rough terrain with road bumps, potholes, etc. The transmission system of the vehicle should be able to dampen the vibrations and shock loads arising from rough terrains. A torque converter provides smooth gear shifting while multiplying the torque. A torque converter transmits power through fluid and hence has high losses through viscous dissipation compared to a manual transmission. Alternatives to torque converter have also been proposed by researchers such as in [35]. Despite its losses and alternatives, torque converters are widely used in off-road vehicles due to their merits.

A torque converter consists of a pump and turbine where it is assumed that the pump shaft spins at the speed of the engine and the turbine shaft is directly connected to the gearbox. A torque converter can transmit power in both directions i.e from the engine to driveline or vice-versa depending upon the speed ratio ϕ which is defined as

$$\phi(\omega_e, V) = \frac{\gamma V}{r_w \omega_e} \tag{2.3}$$

The direction of power transfer depends on the value of ϕ. Power is transmitted to the driveline from the engine if the pump shaft rotates faster than the turbine shaft $(\phi < 1)$ or from the driveline to the engine if $(\phi > 1)$. This means that in the reverse operation of a torque converter, the power available from the deceleration of wheels can be used for hydraulics or engine dynamics. however, in the reverse operation, the efficiency of the torque converter reduces to a third of its normal efficiency. Modeling of a torque converter is complex due to the presence of lock-up clutches and split torque modes. Numerous studies have been performed on modeling torque converters. A dynamic torque converter model was introduced in [5] for restricted operating conditions which are extended to all operating conditions in [36]. Models of torque converters range from higher-order differential conditions to polynomial-based functions as in [37]. Bond graph-based method is also developed in [36]. Keeping DP and its curse of dimensionality in mind, a model which is lower in order while preserving accuracy is preferred.

A polynomial with lookup table-based model is developed by [37]. In this model, the torque converter is modeled using two characteristic curves, one each for torque ratio and the torque on the pump side of the TC vs the speed ratio. As mentioned in [4], the transition at $\phi = 1$ is non-smooth and discontinuous. Hence, an efficiency-based modeling approach developed in [37] is used to avoid discontinuity. The model with characteristic curves is as follows:

$$P_{pump} = c_1 \, T_{pump}(\phi) \, \omega_e^{a_1} \tag{2.4}$$

$$P_{turb} = c_2 \, P_{pump} \, \eta(\phi) \, \omega_e^{a_2} \tag{2.5}$$

where c_1, c_2 are normalization constants, a_1 and a_2 are tuning parameters, γ is the gear ratio of the gear box, engine speed ω_e is in (rps). Fig.2.4 represents the TC characteristics used in this model.

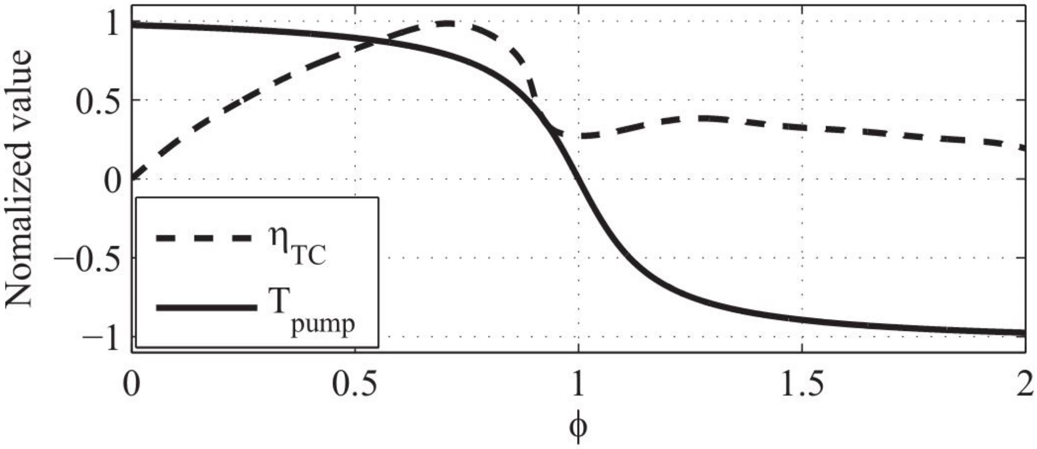


Figure 2.4: Torque Converter characteristic curves (Normalized) for efficiency and Pump shaft Torque [4] .

The gear ratios of the gearbox are assumed to be $\gamma = \{-60, 0, 60, 0\}$ for four phases of SLC respectively.

Vehicle speed dynamics is affected by the forces acting on it. The forces involved are traction force, rolling resistance and braking if any. Air drag force is neglected in this thesis. Considering mass of the loaded vehicle, the dynamics of vehicle speed are as follows:

$$\frac{dV}{dt} = \frac{F_{trac} - sign(V)\,(F_{roll} \, + \, U_{brake})}{M_{total}} \tag{2.6}$$

$$\frac{dS}{dt} = V \tag{2.7}$$

where F_{trac} is the traction force available, F_{roll} is the rolling resistance force on the vehicle, S

is the displacement of the vehicle and U_{brake} is the braking force (a control input) limited to phases 2 and 4 in this thesis. Traction and rolling resistance force are calculated as shown:

$$F_{trac} = \frac{P_{turb}}{\eta_{gb}\, V} \tag{2.8}$$

$$F_{roll} = c_r\, M_{total}\, g \tag{2.9}$$

where η_{gb} is the efficiency of the gearbox, c_r is the rolling resistance coefficient. While braking in phase 2 and phase 4, the vehicle is in a neutral state and there is no power transmission from engine to drive train and hence power required for traction is calculated as

$$P_{trac} = P_{pump}\ \mid sign(\gamma) \mid \tag{2.10}$$

2.1.2 Steering Dynamics

As the wheel loader traverses back and forth through rough terrain at work sites, it is beneficial to have an articulated wheel loader as opposed to a traditional wheel loader. An articulated wheel loader is split into two parts and then joined using a revolute joint. Fig.2.5 shows a schematic of an articulated wheel loader.

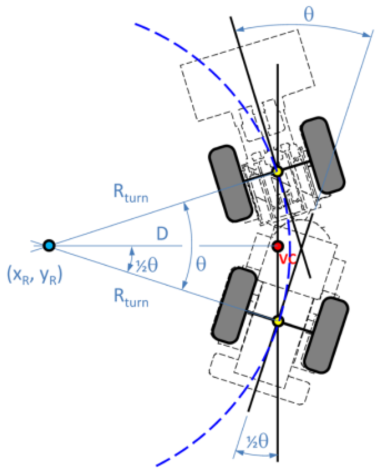


Figure 2.5: Schematic diagram of an articulated wheel loader [5].

This mechanism improves the maneuverability of the wheel loader in tight spatial conditions compared to a traditional one. Articulated wheel loader's steering is handled through rear wheels which frees the front wheels to do their best while scooping and traversing. Numerous studies have been performed on path planning and trajectory optimization of a wheel loader in SLC e.g [5], [38], [39]. In this thesis, a kinematic model is considered for the position, steering, and heading angles of the wheel loader. The governing equations are as follows:

$$\frac{dx}{dt} = V\ cos(\beta) \tag{2.11}$$

$$\frac{dy}{dt} = V\ sin(\beta) \tag{2.12}$$

$$\frac{d\beta}{dt} = \frac{V}{R_{turn}(\delta)} \tag{2.13}$$

$$\frac{d\delta}{dt} = U_{st} \tag{2.14}$$

where $[X, Y, \beta, \delta]$ represent the (X, Y) position of the vehicle, heading, and steering angles respectively. $R_{turn}(\delta)$ is the turning radius of the vehicle which is defined as

$$R_{turn}(\delta) = \frac{L}{2\,tan(\delta/2)} \tag{2.15}$$

where L (mtr) is the wheelbase obtained from [3].

Power required to steer the wheel loader is assumed to be proportional to square of the steering input (U_{st}) as

$$P_{steer} = C_{st}\,U_{st}^2 \tag{2.16}$$

where C_{st} is the steering power coefficient.

2.1.3 Hydraulic Lift and Tilt System

A hydraulic system typically involves a prime mover such as a pump which provides energy in the form of kinetic and pressure energy in the fluid, proportional valves to control the direction and flow rate in the system, an actuator such as a hydraulic cylinder or motor depending upon the application. Various pumps that come under Positive Displacement (PD) pumps and Non-positive Displacement (NPD) pumps have been modeled by researchers for different applications. A model for the centrifugal pump model is developed by authors in [40]. For an off-road vehicle that deals with high-pressure applications, a positive displacement is a better choice. Various positive pumps with and without pressure compensation or load sensing capabilities have been modeled. A load-sensing pump model is developed in [9]. A full and reduced-order model with a constant pressure source is developed as in [41]. The majority of the industrial equipment uses positive displacement pumps such as gear, vane, piston pumps, etc which are flow source-based and have a constant or variable displacement based on the nature of the pump. The pressure in the system develops because of external load or internal resistance such as friction etc. Usually, in on-road vehicles, all the subsystems receive power simultaneously as the load requirements are less whereas, in an off-road vehicle, substantial power is required to perform loading/unloading operations and steer the vehicle to its destination. All the subsystems cannot function at the same time depending upon power from the engine and the load requirements from subsystems.

A typical hydraulic system in a wheel loader has lift and tilt functions. Lifting serves the purpose of raising and lowering the bucket depending upon the height of the load and the load truck and tilting serves the purpose of loading and unloading the bucket. A SLC in this thesis is assumed to start from a loaded position and hence the tilt dynamics are neglected. In this thesis, the majority of the modeling of lift dynamics is imported from [9]. The dimensions required for lift cylinders are obtained from Volvo's brochure [3]. The schematic diagram of the model is shown in Fig.

The above figure represents one of two lift cylinders arranged in a parallel combination for stability. The following are assumed:

- Bulk Modulus is assumed constant.
- Friction between cylinder and piston is neglected.
- Changes in density and viscosity are neglected.
- Fluid is incompressible.
- Pump shaft is connected to the crankshaft of the engine.

The torque T_p and flow rate Q_p of the pump are given by

$$T_p = \frac{P_p\, D_p}{\eta_p} \tag{2.17}$$

$$Q_p = \frac{D_p\, \omega_e}{\eta_v} \tag{2.18}$$

where P_p is the output pressure of the PD pump, D_p is the variable displacement of the pump, η_p is the mechanical efficiency of the pump, η_v is the volumetric efficiency.

The governing equations representing the hydraulic system are as follows:

$$\dot{P}_a = \frac{B_{mod}}{V_{a0} + A_a x_1}(Q_a - A_a \dot{x_1}) \tag{2.19}$$

$$\dot{P}_b = \frac{B_{mod}}{V_{b0} + A_b(Str - x_1)}(Q_b + A_b\dot{x_1}) \tag{2.20}$$

where P_a and P_b are pressures at the piston and rod sides of the cylinder respectively, B_{mod} is the bulk modulus, V_{a0} and V_{b0} are the clearance volumes on both sides of the cylinder, A_a and A_b are areas of cross-sections on piston and rod end side of the cylinder, x_1 is the displacement of the piston with Str is the stroke of the piston, Q_a and Q_b are the flow rates in/out of the cylinder which are defined as

$$Q_a = \begin{cases} z_1 C_d A\sqrt{\dfrac{2(P_s - P_a)}{\rho}} & \text{if } z_1 > 0, \\ z_1 C_d A\sqrt{\dfrac{2(P_a - P_T)}{\rho}} & \text{if } z_1 < 0, \\ 0 & \text{if otherwise;} \end{cases} \tag{2.21}$$

$$Q_b = \begin{cases} -z_2 C_d A\sqrt{\dfrac{2(P_b - P_T)}{\rho}} & \text{if } z_2 > 0, \\ -z_2 C_d A\sqrt{\dfrac{2(P_s - P_b)}{\rho}} & \text{if } z_2 < 0, \\ 0 & \text{if otherwise;} \end{cases} \tag{2.22}$$

where z is the spool valve displacement (m), C_d is the orifice discharge coefficient, A is the orifice area (m^2) and ρ is the density of the fluid in the hydraulic system.

Finally, the dynamics of the piston in a cylinder is modeled as

$$m_p\ddot{x_1} = P_a A_a - P_b A_b - F_{load} \tag{2.23}$$

where m_p is the mass of the piston in the lift cylinder, F_{load} is the load acting on the piston.

2.1.4 Reduced Order Model

2.1.4.1 Overview

Proper Orthogonal Decomposition (POD) is a statistical tool employed to reduce the order of a mechanical system by projecting high dimensional data into low dimensional data. This method identifies a correlation between states of the dynamical system and creates a new set of uncorrelated variables much smaller than the number of original states. The new states are physically irrelevant as they are a combination of different states and they are purely mathematical. The main advantage of POD is a reduction of computational effort owing to less number of states. Control of mechanical systems becomes easier with lesser states. This is highly useful in nonlinear systems. POD has applications in various fields such as fluid mechanics, finite element modeling, and structural dynamics. A nonlinear system is considered and the detailed procedure is outlined in [1] Consider a nonlinear system that is non-dimensionalized:

$$\frac{dy}{dt} = Ay(t) + F(y(t)); \quad y(0) = y_0 \tag{2.24}$$

Simulate the system for a time t $\in$ [0,T] and the solution at each time is appended to create a *snapshot* matrix $\mathbf{Y}$ = $[y_0(t), y_1(t), ..., y_n(t)] \subset \Re^n$. Now, we perform singular value decomposition (SVD) to find the dominant vector space in which our data lies. SVD of

$$\mathbf{Y} = U\Sigma V^T \tag{2.25}$$

where **U** is the orthonormal basis of our data and Σ is the singular value matrix with values in descending order. The singular values signify the dominance of the corresponding mode in obtained data. To reduce our model, we consider first $k << n$ dominant modes and form the orthonormal basis as

$$\phi = [U(:,1), U(:,2), ..., U(:,k)] \tag{2.26}$$

Now we create our reduced order states $\hat{y}$ as

$$y = \phi\hat{y} \text{ or } \hat{y} = \phi^T y \text{ with } \phi^T\phi = I. \tag{2.27}$$

The original equation reduces to

$$\phi\frac{d\hat{y}}{dt} = A\phi\hat{y}(t) + F(\phi\hat{y}(t)) \tag{2.28}$$

which can be written as

$$\frac{d\hat{y}}{dt} = \phi^T A\phi\hat{y}(t) + \phi^T F(\phi\hat{y}(t)) \quad \text{with} \quad \hat{y}(0) = \phi^T y. \tag{2.29}$$

We simulate the reduced order system and the original features are extracted using the relation $y = \phi\hat{y}$

2.1.4.2 Applications to Offroad Vehicles

With Construction and Mining industries using a lot of energy daily, it is certain that a small optimization in operation can save a significant amount of energy. Also mining and construction industries are deemed to be hazardous for the workforce, automation of offroad vehicles can prevent injuries for the workforce. With this motivation, there has been a lot of research in energy and cycle time optimization of a single offroad vehicle such as wheel loaders used in mining. However, there has been very little research on the automation of the continuous autonomous operation of wheel loaders. This involves mathematical modeling of various powertrain and drivetrain components such as turbocharged engine, torque converter, and also the hydraulics (Lift and Tilt system) along with the longitudinal dynamics which gives us the order of the system as ~ 15. Considering optimization of series of vehicles with the above number of states, the model is highly computationally intensive to control. This demands a simpler overall model to be controlled with a simple model for engine, torque converter, and hydraulics. An experimentally validated engine model is approximated to be linear and validated with test data is used. To further simplify the model, the order of the whole system can be reduced for less computational effort with a trade-off inaccuracy.

For POD to be effective, the correlation between the states has to be high. Here, the engine, velocity, hydraulics, and longitudinal kinematics have less correlation between them. Hence, the effective possible order reduction is in the hydraulic system as it correlates with lift and tilt system pressures at piston and rod ends. A 25 % reduction is possible in the hydraulic system which can play a huge role in the optimization of several vehicles running in parallel. A simple analysis of Model Order Reduction (MOR) using POD is done for a simple hydraulic system which reduces the order of the system by 25% and the result is attached:

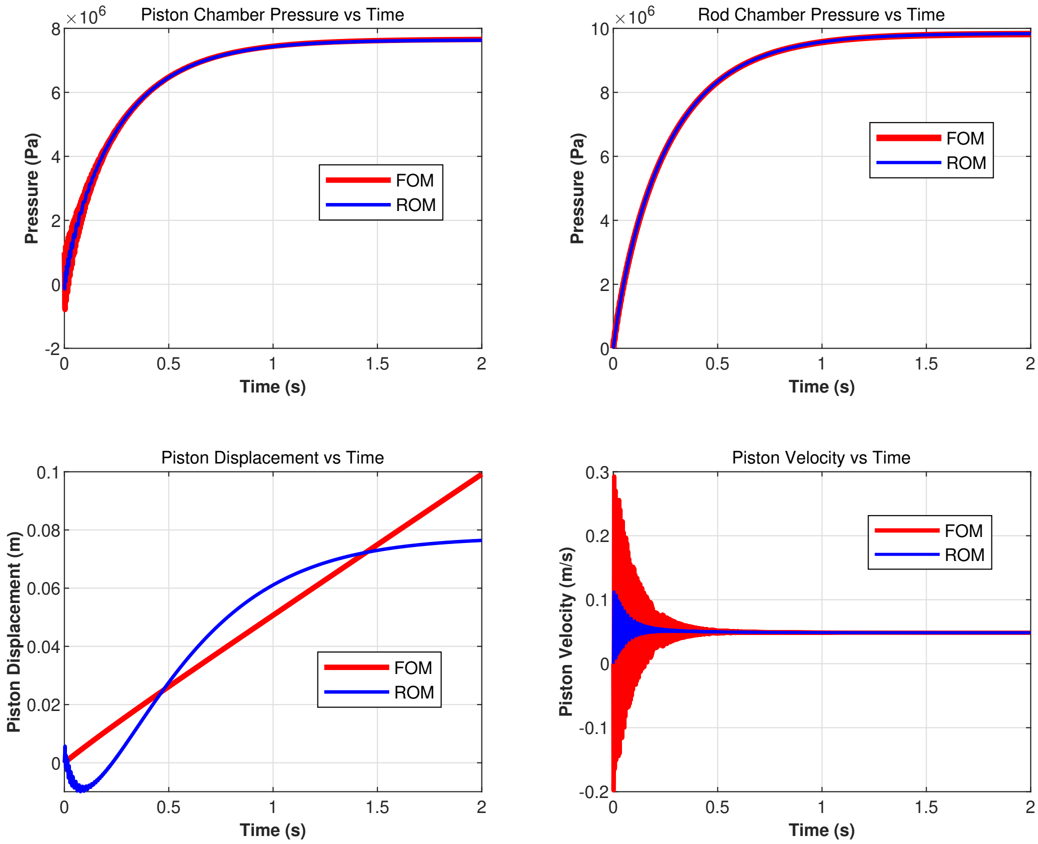


Figure 2.6: Full and Reduced-Order Model of Hydraulic System.

2.1.4.3 Observations and conclusions

Results show a considerable match between the Full Order Model (FOM) and Reduced-Order Model (ROM) for chamber pressures and velocity of the system. However, the piston displacement in ROM does not match well with the FOM. Piston displacement also shows non-minimum phase behavior in the ROM. Piston displacement needs to be highly accurate as it determines the position of the bucket. It cannot be guaranteed that the ROM will be faster than the FOM in all cases because of the cost of conversion of states and the governing equations from the FOM to the ROM domain. Owing to the above reasons, a reduced-order model is not beneficial in this case.

2.1.5 Simplified Hydraulic Model

A comprehensive model is developed to capture the dynamics of the lift and tilt mechanism of the bucket. To use dynamic programming as a tool, it is preferred to have fewer states to avoid complexities and the curse of dimensionality. A simplified model would be a better option for dynamic programming. However, the comprehensive model can be considered for numerical approach based optimization. A simplified version of lift dynamics is considered for modeling the bucket dynamics in [4]. It is as follows:

$$\frac{dS_{buc}}{dt} = V_{buc} \tag{2.30}$$

$$\frac{dV_{buc}}{dt} = U_{lift} \tag{2.31}$$

where S_{buc} is the bucket position, V_{buc} is the bucket velocity, and U_{lift} is the acceleration of the bucket (Control input).

The power required to lift the bucket P_{lift} is calculated as follows

$$F_{load} = M_{load}(g + U_{ab}) \tag{2.32}$$

$$P_{lift} = \frac{F_{load}V_{buc}}{\eta} \tag{2.33}$$

where M_{load} is the mass of the bucket loaded, $g = 9.81 m/s^2$, η is the efficiency of the hydraulic system.

2.1.6 Full State Dynamics

A complete model of wheel loader consists of three control inputs and 8 states (simplified Hydraulics model). Refer to nomenclature for state and control description.

$$X = [\omega_e, V, S_{buc}, V_{buc}, X, Y, \beta, \delta] \tag{2.34}$$

$$U = [U_{mf}, U_{ab}, U_{st}] \tag{2.35}$$

The governing equations for the entire system are as follows:

$$\frac{d\omega_e}{dt} = \frac{1}{I_e}(T_e - \frac{P_{trac} + P_{steer} + P_{lift}}{\omega_e}) \tag{2.36}$$

$$\frac{dV}{dt} = \frac{F_{trac} - sign(V)\ (F_{roll}\ +\ U_{brake})}{M_{total}} \tag{2.37}$$

$$\frac{dS_{buc}}{dt} = V_{buc} \tag{2.38}$$

$$\frac{dV_{buc}}{dt} = U_{ab} \tag{2.39}$$

$$\frac{dx}{dt} = V\ cos(\beta) \tag{2.40}$$

$$\frac{dy}{dt} = V\ sin(\beta) \tag{2.41}$$

$$\frac{d\beta}{dt} = \frac{V}{R_{turn}(\delta)} \tag{2.42}$$

$$\frac{d\delta}{dt} = U_{st} \tag{2.43}$$

2.1.7 Results and Validation

Experimental validation was not possible due to the lack of data. However, the model developed in [6] is validated. The same control inputs U_{mf}, U_{ab} are applied to the model having 5 states excluding the lateral kinematics. The state profile obtained from the current model matches considerably with the validated state profile obtained in [6]. Fig. 2.7 shows the control profile and the state profile for comparison. The excluded part of the model that is lateral kinematics is not parametric and is the same as in [4].

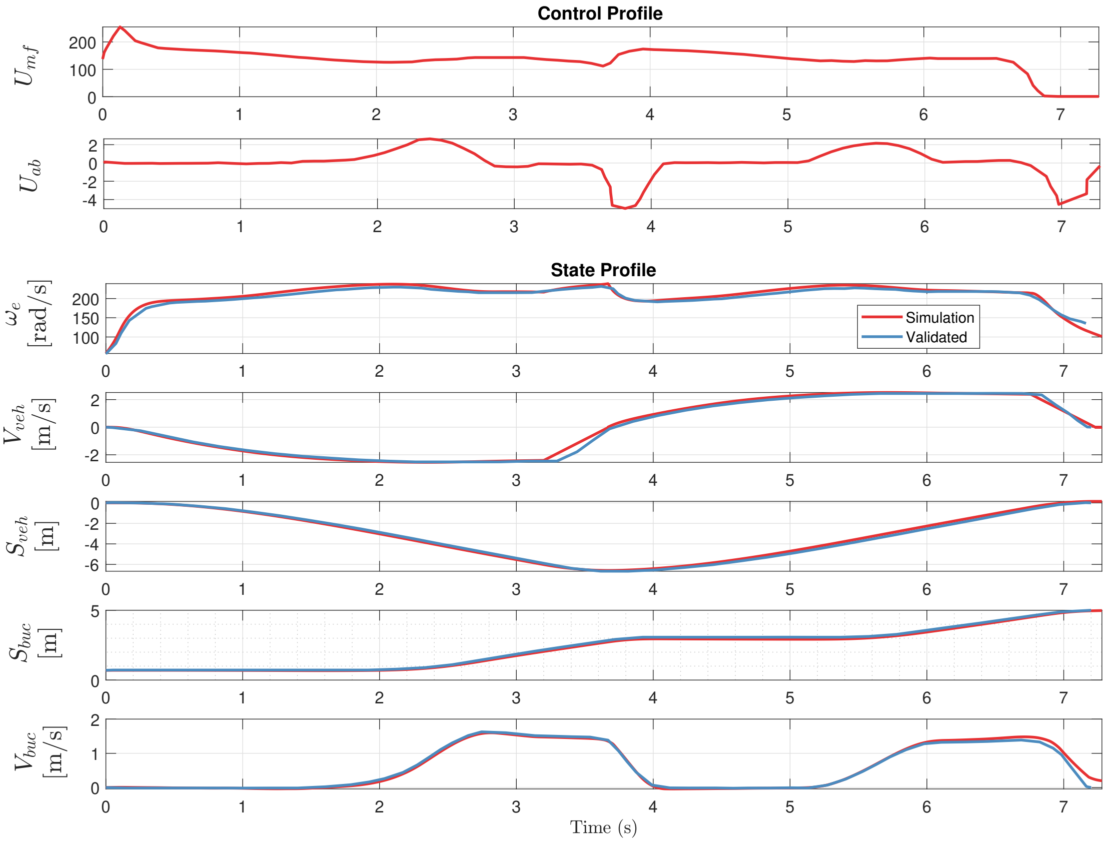


Figure 2.7: Control profile imported from [6]. Resulting state profiles from the current developed model and from the paper [6]

3. Problem Outline

3.1 Formulation - Optimal Control Problem

The augmented wheel loader system dynamics consist of engine, vehicle dynamics, steering, and hydraulic subsystems. We assume that the power consumed by the steering system is negligible compared to traction and hydraulics. The system has 8 states, 3 inputs as outlined in the modeling section. Two cases of optimization of an SLC are considered for comparing DP and DM. An Optimal Control Problem (OCP) for the two cases is formulated and the approach to solve the OCP using DP is detailed in this section. DP is focused more in this section as we adopt a split-stage optimisation approach in this research. The idea is discussed in detail. In case of DM, a conventional pseudo-spectral collocation based method is adopted. The objective function, phases in an SLC, constraints in both cases followed by detailed approach to solve the system using DP and DM are defined.

The objective functions for the optimization problem can be as follows:

- Time Efficiency - The objective of this analysis is to perform the short loading cycle as early as possible without any regard to the fuel consumed. This can be fairly easily achieved by injecting maximum fuel into the engine. The vehicle has higher torque available for maximum acceleration which drives the vehicle faster to the desired destination.

- Fuel Efficiency - Time taken by the wheel loader to perform an SLC is not a concern in this analysis. The primary purpose of this approach is to have the wheel loader consume as little fuel as possible.

- Productivity - Here, to maximize productivity, the objective function is a weighted average of the above objectives. A fixed time is chosen which is between the time taken for the above two contradictory objectives and then the cycle is optimized for fuel-efficiency. It is assumed that there is no time delay between any of the phases in an SLC.

In this research, the latter of the three cost functions is considered. We consider a fixed final time and optimize the system for fuel consumption. The short loading cycle (SLC) consists of four phases as described earlier which are differentiated by a parameter γ, gear ratio. However, the convention is briefly discussed below.

1. Deceleration - In this phase, the wheel loader travels with a reverse gear of gear ratio ($\gamma = -60$). This phase of SLC is assumed to be done in time t_1. The bucket position is assumed to be at its lowest position at the start of this phase.

2. Braking - This phase starts at t_1 and ends when the vehicle reaches a stop. During braking, it is assumed that the driveline is disengaged from the power train. The gear ratio for this phase is ($\gamma = 0$). The duration of phase - 2 depends on the velocity of the vehicle at time t_1.

3. Acceleration - After momentarily being at a halt, the wheel loader accelerates forward with a gear ratio ($\gamma = 60$). This phase starts immediately with the end of phase - 2 and ends at the time (t_3).

4. Braking - This phase is similar to phase - 2. It starts at t_3 and ends when the vehicle comes to rest.

The cost function for this OCP only contains an integral term and hence can be expressed in the lagrange form [42]. The problem is subjected to several limits and constraints such as state dynamics, path, boundary and box constraints etc. The governing equations for the model can be found in the modeling section.

Cost function:

$$\phi = \min_{u_{mf}, u_{st}, u_{ab}} \quad (\int_0^{t_f} \dot{m_f} dt) \tag{3.1}$$

States:

$$x = [\omega_e, V, S_{buc}, V_{buc}, X, Y, \beta, \delta] \tag{3.2}$$

Controls:

$$u = [U_{mf}, U_{ab}, U_{st}] \tag{3.3}$$

State dynamics:

$$\dot{x}(t) = f(x, u, t) \tag{3.4}$$

Constraints:

$$k_{min} \leq f(x) \leq k_{max} \tag{3.5}$$

$$x_{min} \leq x(t) \leq x_{max} \tag{3.6}$$

$$u_{min} \leq u(t) \leq u_{max} \tag{3.7}$$

In this analysis, the phase times are assumed to be known and to be same in both cases. The phase times for this multi-phase optimization problem can be considered as variables as considered in [4]. This increases the complexity of the problem. However, the computations can be run in parallel and do not cause an increase in run time and that is considered as part of future work. The initial and final conditions for the optimal control problem for both cases are tabulated in Fig. 3.1 and 3.2. The two cases considered here only differ by the final position the wheel loader has to reach. The objective is to find the 3 control inputs which lead the wheel loader to the desired final states while obeying the constraints.

The majority of the box constraints are borrowed from the model in [4]. Constraints for engine speed and lateral kinematics are imported from Volvo brochure [3]. Box constraints for all the state and control variables are listed in Table.3.3.

This OCP can be solved using different methods as outlined in the review section. This research focuses on DP and DM. An approach to solve the OCP along with the procedure is discussed in the following section.

Case:1

Variables	Phase - 1		Phase - 2		Phase - 3		Phase - 4	
	t_0	$t_1 = 4s$	$t_1 = 4s$	t_2	t_2	$t_3 = 9s$	$t_3 = 9s$	t_4
$\omega_e\ (rad/s)$	57	$-$	$-$	$-$	$-$	$-$	$-$	$-$
$V\ (m/s)$	0	$-$	$-$	0	0	$-$	$-$	0
$S_{buc}\ (m)$	0.7	$-$	$-$	$-$	$-$	$-$	$-$	5.0
$X\ (m)$	0	$-$	$-$	$-$	$-$	$-$	$-$	-2.5
$Y\ (m)$	0	$-$	$-$	$-$	$-$	$-$	$-$	2.5
$\beta\ (rad)$	0	$-$	$-$	$-$	$-$	$-$	$-$	$\frac{\pi}{2}$
$\delta\ (rad)$	$\frac{\pi}{16}$	$-$	$-$	$-$	$-$	$-$	$-$	$-$
$U_{brake}\ (N)$	0	0	$1e5$	$1e5$	0	0	$1e5$	$1e5$

Table 3.1: Case:1 - Constraints for the Optimal Control Problem.

Case:2

Variables	Phase - 1		Phase - 2		Phase - 3		Phase - 4	
	t_0	$t_1 = 4s$	$t_1 = 4s$	t_2	t_2	$t_3 = 9s$	$t_3 = 9s$	t_4
$\omega_e\ (rad/s)$	57	$-$	$-$	$-$	$-$	$-$	$-$	$-$
$V\ (m/s)$	0	$-$	$-$	0	0	$-$	$-$	0
$S_{buc}\ (m)$	0.7	$-$	$-$	$-$	$-$	$-$	$-$	5.0
$X\ (m)$	0	$-$	$-$	$-$	$-$	$-$	$-$	-3.0
$Y\ (m)$	0	$-$	$-$	$-$	$-$	$-$	$-$	3.5
$\beta\ (rad)$	0	$-$	$-$	$-$	$-$	$-$	$-$	$\frac{\pi}{2}$
$\delta\ (rad)$	$\frac{\pi}{16}$	$-$	$-$	$-$	$-$	$-$	$-$	$-$
$U_{brake}\ (N)$	0	0	$1e5$	$1e5$	0	0	$1e5$	$1e5$

Table 3.2: Case:2 - Constraints for the Optimal Control Problem.

3.2 Approach - Dynamic Programming

The continuous system is discretized into finite dimensional system. It is widely known that DP suffers from curse of dimensionality. The finer the discretization, the accurate the solution and more the time it takes. The computational effort to solve the problem with DP increases exponentially with states and control inputs. DP is usually applied for systems with number of

Parameters	Value
ω_e	$[50, 250]\ (rad/s)$
V	$[-3, 3]\ (m/s)$
S_{buc}	$[0.7, 5]\ (m)$
V_{buc}	$[0, 2]\ (m/s)$
δ	$[-0.8, 0.8]\ (rad)$
U_{brake}	$\{0, 1e5\}\ (N)$
U_{mf}	$[1, 250]\ (mg/cycle)$
U_{st}	$[-1.2, 1.2]\ (rad/s)$
R_{turn}	$[R_{min}, R_{max}]\ (m)$

Table 3.3: Box constraints for state and control variables.

states less than 5 and inputs less than 3. The current OCP has 8 states and 3 inputs which in this state is computationally infeasible to apply DP. Fig.3.1 shows the number of operations at each time stamp vs number of states or control inputs.

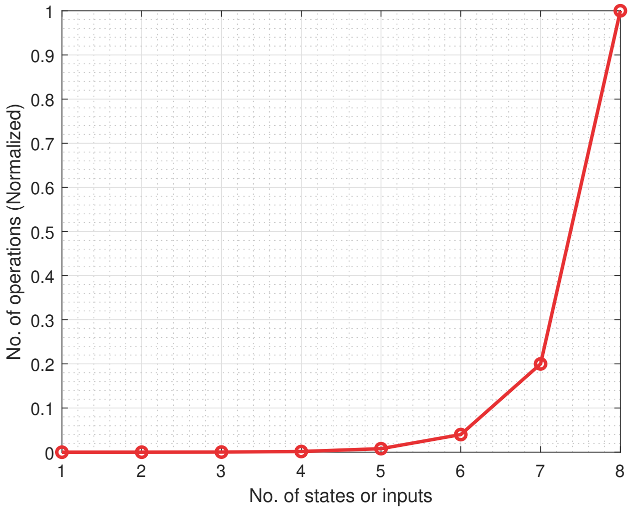


Figure 3.1: Curse of Dimensionality in DP.

For computational feasibility, one can have coarse discretization which compromises the accuracy or split the problem into multiple subproblems. Here, the OCP is split into three stages and later integrated back together to yield an optimal result. We call this as the split stage approach.

This approach can compromise the global optimality of the solution. However, the stages are integrated with flexibility. This reduces the maximum number of states and inputs in each stage of optimisation are 6 and 2 respectively. Let's assume that the state and control space is discretized into k grids each. The integrated approach has 11 in total whereas the split stage approach has three stages with a maximum of 8. The split stage approach reduces the amount of operations at each time stamp by a factor of k^3. Table.3.4 shows the number of operations at each time stamp for both the approaches. The framework is shown in Fig. 3.2

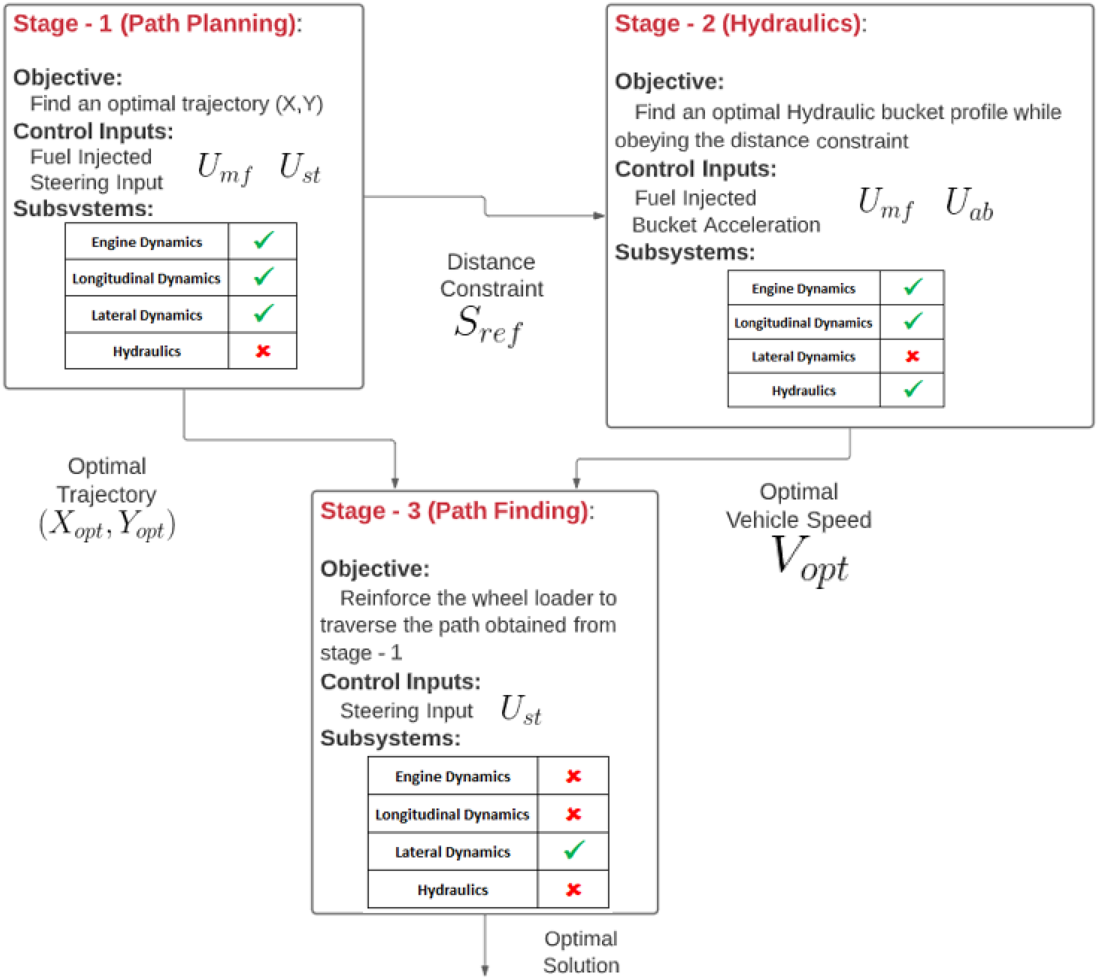


Figure 3.2: Flow chart for Split-Stage Optimization approach.

	Integrated	**Stage - 1**	**Stage - 2**	**Stage - 3**
States and Control Inputs	8 & 3	6 & 2	5 & 2	4 & 1
Subsystems	1. Engine 2. Longitudinal 3. Lateral 4. Hydraulics	1. Engine 2. Longitudinal 3. Lateral 4. Hydraulics	1. Engine 2. Longitudinal 3. Lateral 4. Hydraulics	1. Engine 2. Longitudinal 3. Lateral 4. Hydraulics
No. of operations	k^{11}	k^{8}	k^{7}	k^{5}

Table 3.4: Characteristics of Integrated and split-stage optimization approaches. (Blue - Included ; Red - Ignored)

Considering 20 grid divisions for each state and control input, the number of operations increase by 20^3 times for the current OCP compared to the usual number for which DP is applied. Fig.3.3 shows an estimated reduction in the number of operations in a logarithmic scale.

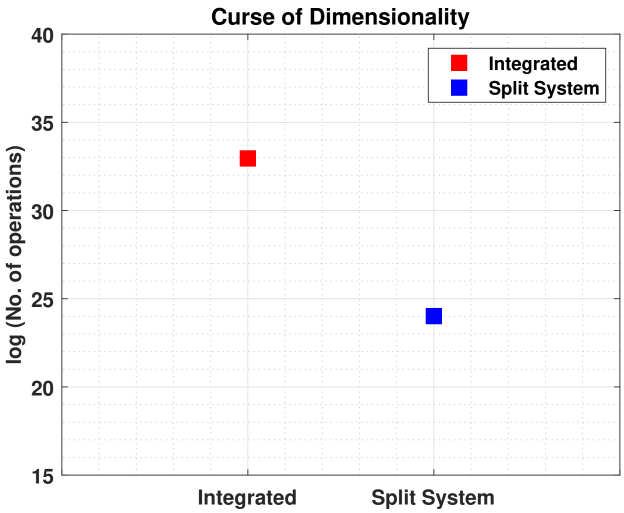


Figure 3.3: Comparison of number of operations in the Integrated and Split-stage approaches

From the above analysis, split-stage approach seems to be a good trade-off between accuracy and computational time. Hence, we decided to explore more of this approach. The three stages are described as follows:

- Stage - 1 involves path planning which includes engine, vehicle dynamics, and steering. A block diagram of the states and control inputs for stage -1 is shown in Fig.3.4. The primary purpose of this stage is to find the approximate distance a wheel loader has to traverse in an SLC. The hydraulic subsystem is not included in this stage of optimization. The path a wheel loader should traverse for a weighted fuel and time efficiency case is found with the following state-space system.

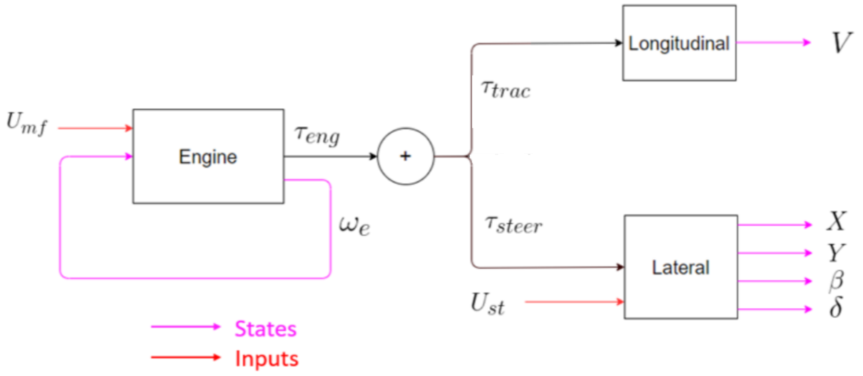


Figure 3.4: State and control flow block diagram for stage - 1 optimization (Path planning)

States:

$$X = [\omega_e, V, X, Y, \beta, \delta] \tag{3.8}$$

Controls:

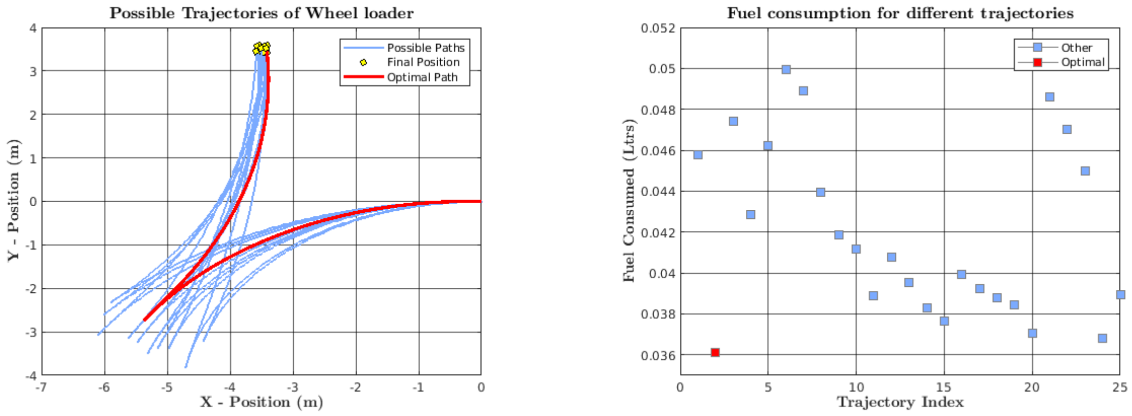
$$U = [U_{mf}, U_{st}] \tag{3.9}$$

Cost Function:

$$\phi = \min_{u_{mf},u_{st}} \left(\int_0^{t_1} \dot{m}_f dt + \int_{t_1}^{t_2} \dot{m}_f dt + \int_{t_2}^{t_3} \dot{m}_f dt + \int_{t_3}^{t_4} \dot{m}_f dt\right) \tag{3.10}$$

where t_1 and t_3 are assumed to be known. The second phase starts at t_1 and ends when the vehicle comes to a halt and similarly phase - 4 starts at t_3 and ends when the vehicle reaches its destination. A sample of possible trajectories in an SLC is shown in Fig. 3.5.

- Stage-2 involves optimization of hydraulics which includes engine, vehicle dynamics, and



(a) Possible Trajectories

(b) Fuel Consumption

Figure 3.5: A sample of possible trajectories with fuel consumption in a SLC

hydraulics. A block diagram for this stage of optimization is shown in Fig.3.6. From stage-1 optimization, the path to be traversed by a wheel loader and also the distance traversed is obtained. An extra state (displacement of the wheel loader) is added to integrate stage-1 to stage-2 optimization. A distance constraint is enforced in this stage to ensure the WL traverses the same distance as it had in stage - 1 at the end of phase-2 and phase-4 of the problem as shown in Fig. 3.2. The resulting state-space system is as follows:

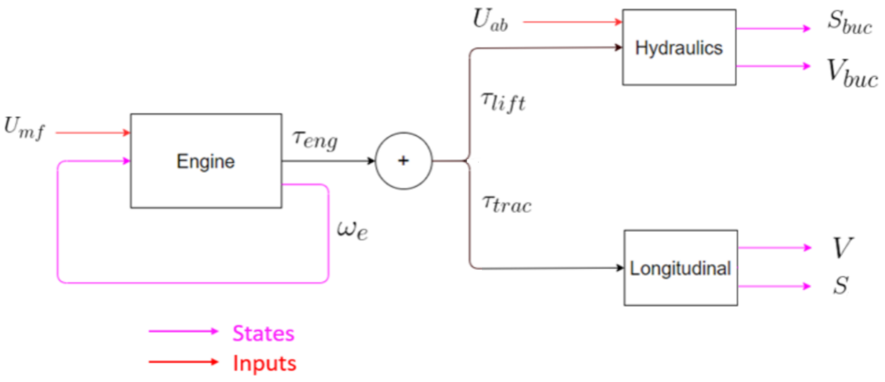


Figure 3.6: State and control flow block diagram for stage - 2 optimization (Hydraulics)

States:

$$X = [\omega_e, V, S, S_{buc}, V_{buc}] \tag{3.11}$$

Controls:

$$U = [U_{mf}, U_{ab}] \tag{3.12}$$

Cost Function:

$$\phi = \min_{u_{mf}, u_{ab}} \int_{0}^{t_1} \dot{m}_f dt + \int_{t_1}^{t_2} \dot{m}_f dt + \int_{t_2}^{t_3} \dot{m}_f dt + \int_{t_3}^{t_4} \dot{m}_f dt \tag{3.13}$$

where t_1 and t_3 are the same as in stage - 1.

- Finally, Stage-3 involves finding the steering input which leads to the wheel loader to the desired final position. This is not a conventional optimization routine but finding a path that is close enough to the path obtained in stage-1. The resulting state space system is as follows:

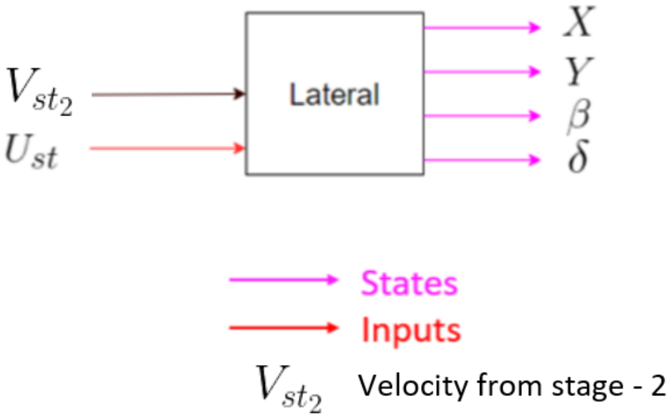


Figure 3.7: State and control flow block diagram for stage - 3 (Integration of stage 1&2)

States:

$$X = [X, Y, \beta, \delta] \tag{3.14}$$

Controls:

$$U = [U_{st}] \tag{3.15}$$

Cost Function:

$$\dot{\psi} = [(X(t) - X_{ref}(t))^2 + (Y(t) - Y_{ref}(t))^2] \tag{3.16}$$

$$\phi = \min_{u_{st}} \quad \int_0^{t_1} \dot{\psi} dt + \int_{t_1}^{t_2} \dot{\psi} dt + \int_{t_2}^{t_3} \dot{\psi} dt + \int_{t_3}^{t_4} \dot{\psi} dt \tag{3.17}$$

where t_1 and t_3 is chosen to be the same as in stage-2.

4. Dynamic Programming

In this chapter, we solve the OCP with a split-stage approach using DP as outlined in the previous section. Solving the continuous differential equations is tedious as the system is of higher order and nonlinear. Numerical integration techniques such as explicit, implicit, Range-kutta (RK4), Hermite-Simpson or Dormand-Prince (DOPRI), etc can be employed to solve the discrete OCP. Some of them are briefly discussed here.

4.1 Discretization

To perform a discrete search through the reachable state space for optimal control inputs, the states and controls have to be discretized. There are many algorithms to discretize differential equations such as explicit, implicit, Range-Kutta (RK4), Dormand Prince (DOPRI) methods, etc. The explicit integration method usually is unstable for highly stiff systems and the discrete-time step h has to be as low as possible to ensure stability. RK4 method is widely used as it is a fourth-order method whereas Euler's explicit method is the first order. While RK4 is widely used for its accuracy, it takes more time to estimate a solution. For a non-stiff system, the explicit method is more useful as it provides an accurate solution in less time. For a stiff system, the time step h has to be small enough to capture the system dynamics accurately. This increases the time it takes to estimate the solution. Sample numerical simulations show that the explicit method does not have any convergence issues and hence it is preferred over the RK4 method for better computational efficiency.

- Euler Explicit algorithm:

$$X[k+1] = X[k] + h\, f(t_k, X_k) \quad X(0) = X_0 \tag{4.1}$$

where $X[k]$ is the state value at time step k, h denotes the discrete time step.

- Runge-Kutta (RK4) algorithm:

$$X[k+1] = X[k] + \frac{h}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right) \quad X(0) = X_0 \tag{4.2}$$

where $X[k]$ is the state value at time step k, h denotes the discrete time step and (k_1, k_2, k_3, k_4) can be obtained from

$$\begin{cases} k_1 = f(X[k], U[k], t[k]), \\ k_2 = f(X[k] + \frac{h}{2}k_1, U[k], t[k] + \frac{h}{2}), \\ k_3 = f(X[k] + \frac{h}{2}k_2, U[k], t[k] + \frac{h}{2}), \\ k_4 = f(X[k] + h\, k_3, U[k], t[k] + h), \end{cases} \tag{4.3}$$

4.2 Procedure

4.2.1 Iterative search using DP with region-based gridding

This problem can be solved with either forward-DP or backward-DP. Forward DP is useful only for deterministic systems while backward-DP is useful also for stochastic systems [43]. However, both the methods yield the same optimal result. Since this research assumes the system to be deterministic, forward-DP and backward-DP are no different. Forward-DP based approach is chosen for convenience as the optimum is the same in both the methods ideally. For real-time implementation, backward DP is better as it can handle stochastic systems. After discretizing the state and control space, for every possible state value at each time step, an exhaustive search is performed over the control space to find the reachable state space at the next time step. Suppose, at time $t[k]$, the reachable state space is known and possible reachable states at $t[k+1]$ is found by projecting the state $x[k]$ by applying all the possible discretizing control inputs $U_1[1:i], U_2[1:j], U_3[1:k]$ at time $t[k]$. The possible reachable state values often may not exactly correspond to the grid points chosen at $t[k+1]$ and fall between the grid points. A schematic of the approach is shown in Fig. 4.1 below:

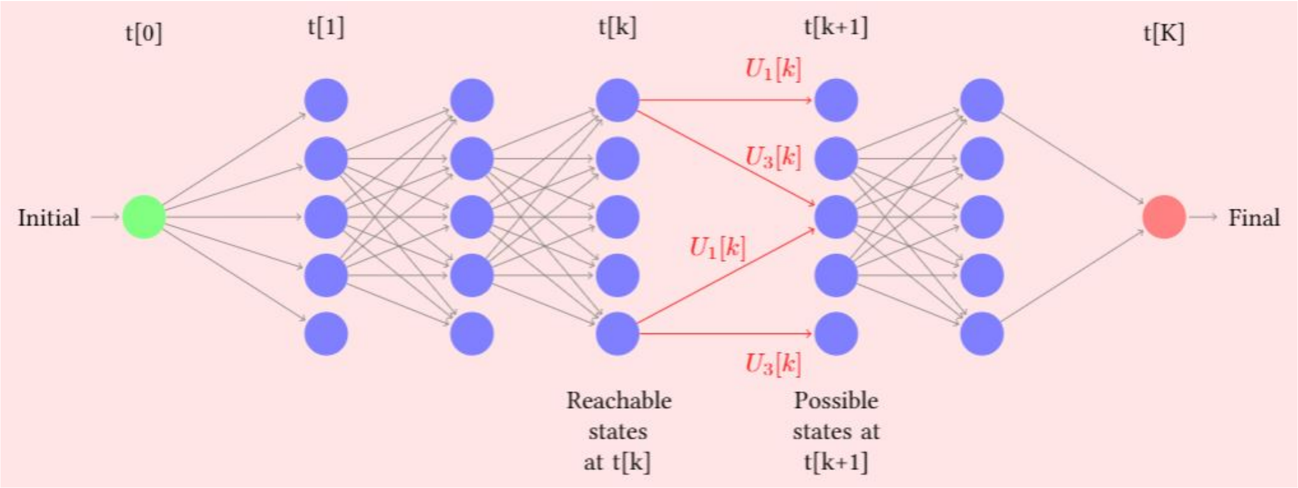


Figure 4.1: Schematic of forward-DP approach.

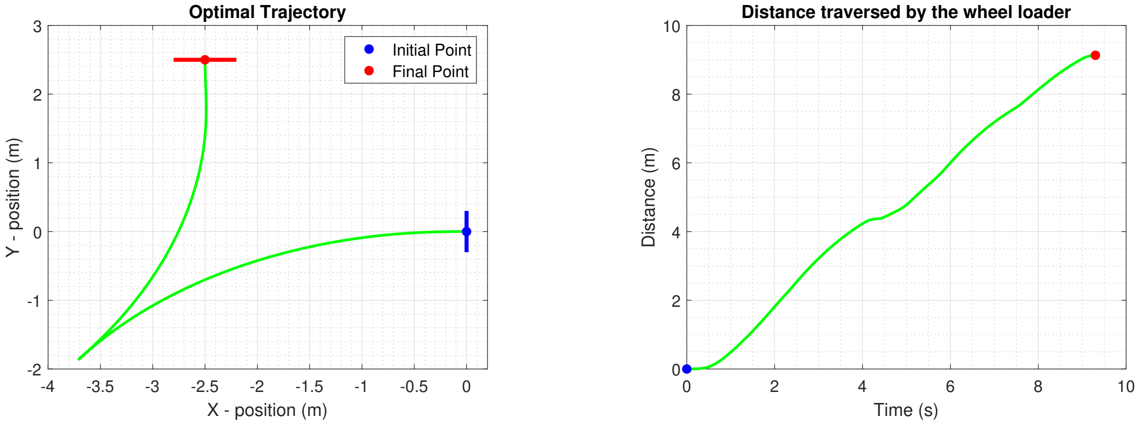
Another approach to this forward DP is to divide the state space into finite regions as opposed to grids. The reachable state space is calculated by projecting the state space at $t[k]$ to time $t[k+1]$ by applying all the discrete control inputs. If multiple projections fall into the same region, the control input and the corresponding state which has minimum local cost to that region are chosen. A point to be noted is that the number of the regions has to be higher enough to avoid local minimum. In the current problem which has 6 states, the number of grids reaches approximately 20^6 assuming 20 grids per state which is still a high number. Reducing the number of grids would reduce the complexity but would compromise the accuracy of the result. One possible way to reduce the number of grids without compromising with accuracy is to identify the possible regions of each state where the optimal could lie and perform region-based searching. In this approach, based on the sample results from simulations, the extremum of states is varied with each time step. This maintains accuracy as the optimal state trajectory always lies in the selected region of interest. In the current problem, we can ignore the velocity (V), X and Y positions, heading (β), and steering angle (δ) profiles which certainly could not steer the vehicle to its destination. The same cannot be done for states which exhibit a high degree of transient behavior throughout the entire time such as engine speed (ω_e) and velocity of the bucket (V_{buc}) and hence the grid points for these type of variables have to be dense for an accurate solution.

4.3 Results

First, we present the results for both cases (1&2) obtained from DP - all three stages individually and the final profile.

4.3.1 Case - 1:

- Stage-1 of DP focuses on trajectory optimization along with engine and vehicle dynamics. This section ignores power consumed by the hydraulics unit. The optimal trajectory obtained from DP is shown in Fig. 4.2(a) along with few other possible trajectories. The optimal path is chosen according to the final conditions of the problem and minimum cost. The distance traversed by the wheel loader and also the optimal trajectory is also shown in Fig. 4.2(b) which forms as a reference for the stage-2 optimization.

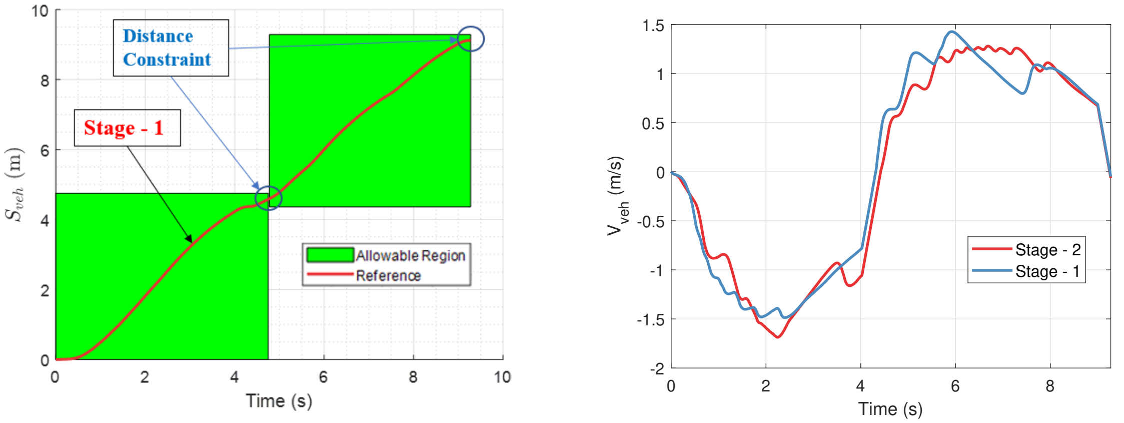


(a)

(b)

Figure 4.2: Optimal Trajectory and Distance traversed by the wheel loader

- Stage-2 focuses on optimization of the entire system considering the wheel loader to traverse the same distance as in stage - 1. For the path traversed by the wheel loader to be the same as in stage-1, we impose a constraint on the distance traversed by the vehicle at the end of phase -2 and phase -4 to be the same as obtained in stage -1 with some flexibility as shown in Fig.4.3.



(a) Distance state space and constraint for stage - 2

(b) Optimal velocity profile from both stages

Figure 4.3: Distance state space and velocity profiles from stage - 1 and stage - 2

As the algorithm progresses with time, we consider the states to be reachable with distance traversed lying in the possible distance region (green in the Fig.4.3). Possible state-space outside this region is ignored as they won't lead to the same distance as in stage - 1. The difference between stage - 1 and stage - 2 comes in the form of a velocity profile. Although the distance profiles would be similar, the constraints in stage - 2 are only applied at the end of phase - 2 and phase - 4, and the velocity profile is flexible in stage - 2 to some extent. The optimal velocity profiles in stage - 1 and stage - 2 are shown in Fig.4.3.

The state profile obtained from the optimization routine is shown below in Fig.4.4.

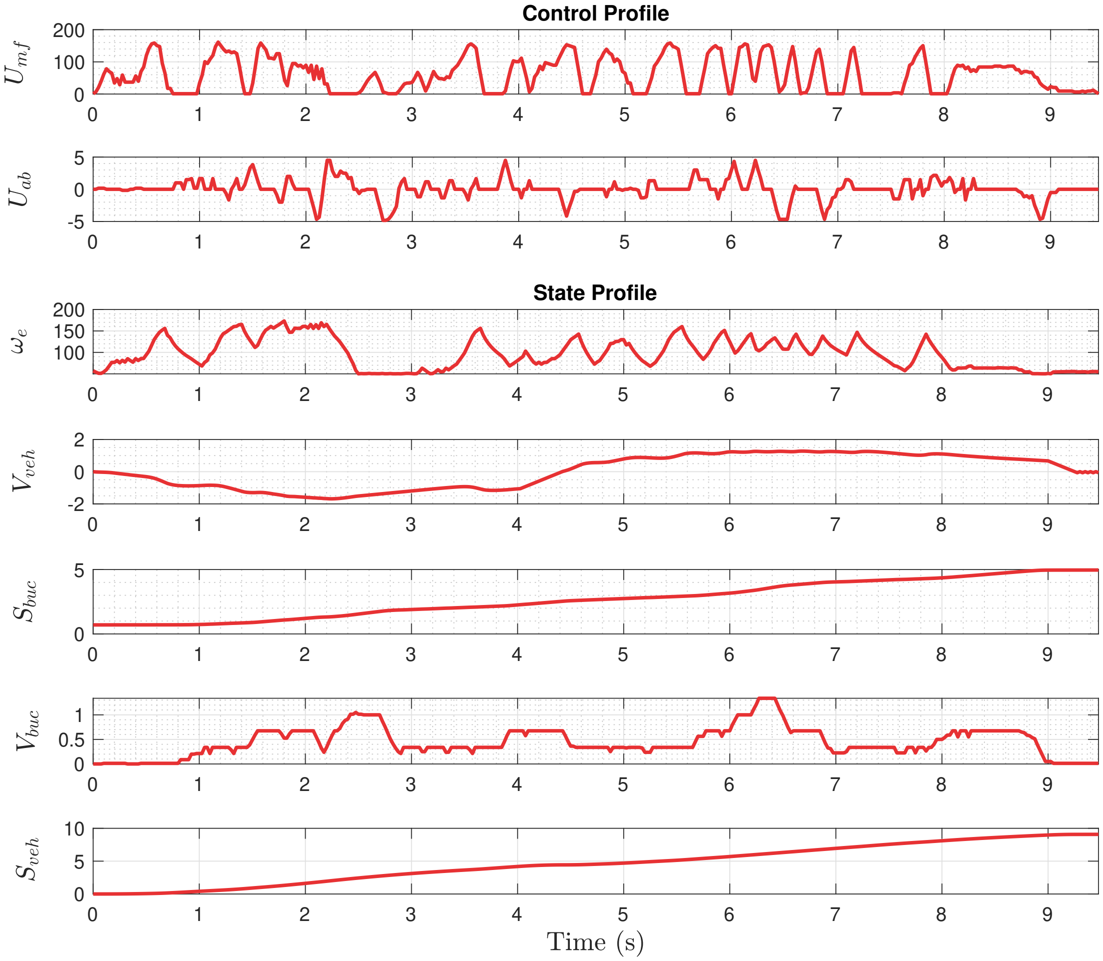


Figure 4.4: Optimal state and control profile obtained from stage - 2

- Stage-3 focuses again on path planning. This stage is not essentially an optimization routine but just finding a steering input that leads the vehicle to the desired final position. The inputs to this stage are the velocity profile obtained from stage-2. The optimal trajectory obtained is shown in Fig.4.5. The heading angle and steering angle profiles are also shown below in Fig.4.6

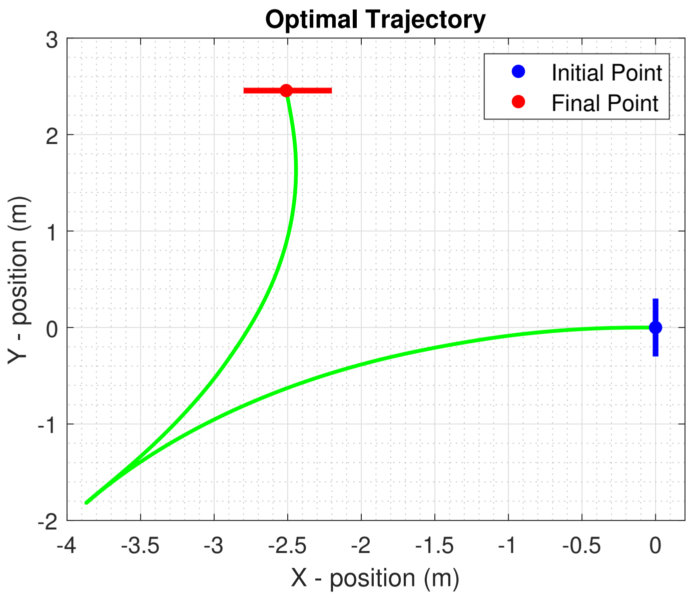


Figure 4.5: Optimal trajectory obtained from stage - 3

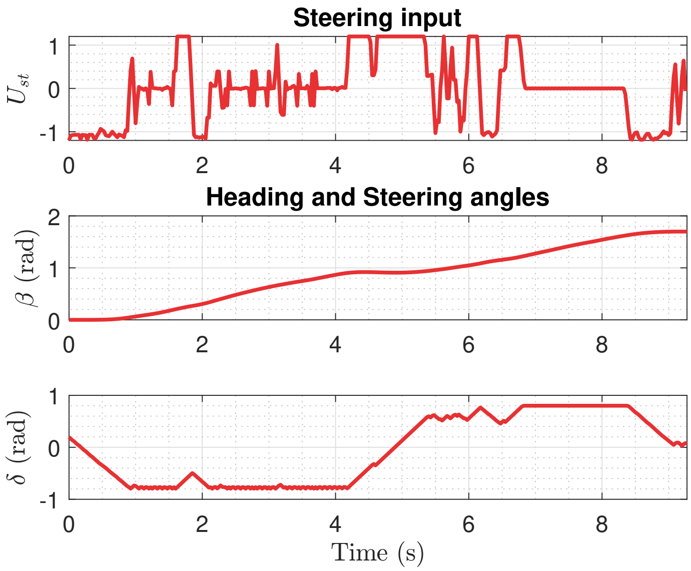


Figure 4.6: Steering and Heading angle profile from stage - 3

4.3.2 Case - 2:

This case is quite similar to case - 1 except for the final position of the wheel loader.

- Stage - 1: In this case, the distance traversed by the wheel loader will be more than it has in case - 1. Fig.4.7 shows the optimal trajectory of the wheel loader and distance traversed by it ignoring hydraulic fuel consumption. Stage-1 of DP focuses on trajectory optimization along with engine and vehicle dynamics. The distance profile forms a reference for stage - 2.

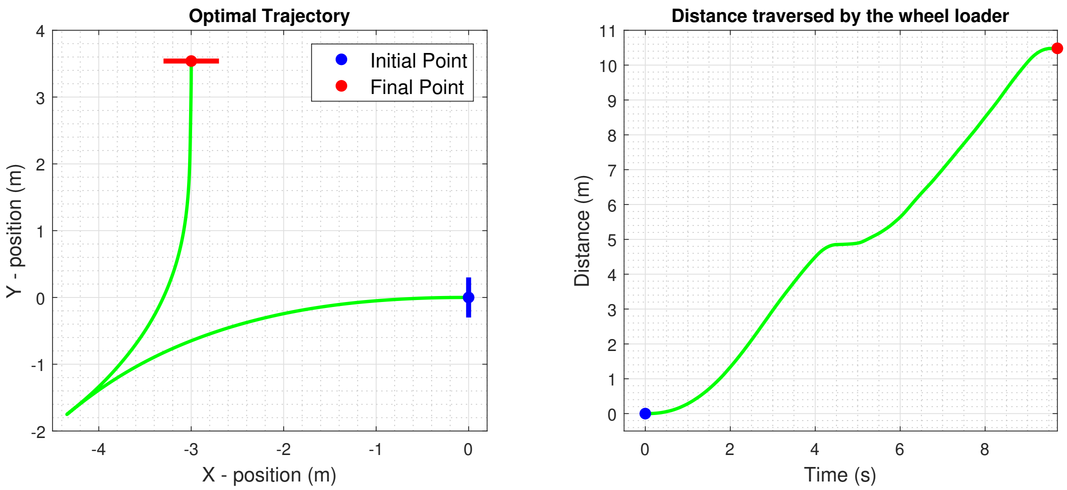
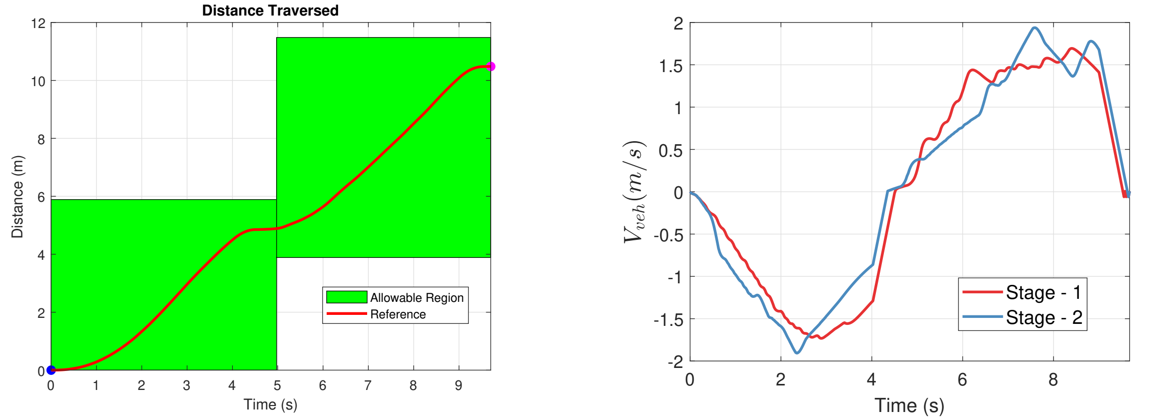


Figure 4.7: Optimal Trajectory and Distance traversed by the wheel loader

- Stage - 2: In this case, it is expected that the fuel consumed in this case would be higher than case - 1. Hydraulics power consumption will be similar to case - 1 whereas traction unit consumes more power in case - 2 as the distance traversed in more. Fig.4.8 shows the constraint on distance traversed and the flexibility of velocity profile of the wheel loader.

  The state profile obtained from the optimization routine is shown below in Fig.4.11.

- Stage - 3: As in case - 1, this stage plans the path to be traversed by the wheel loader taking



(a) Distance state space and constraint for stage - 2

(b) Optimal velocity profile from both stages

Figure 4.8: Distance state space and velocity profiles from stage - 1 and stage - 2

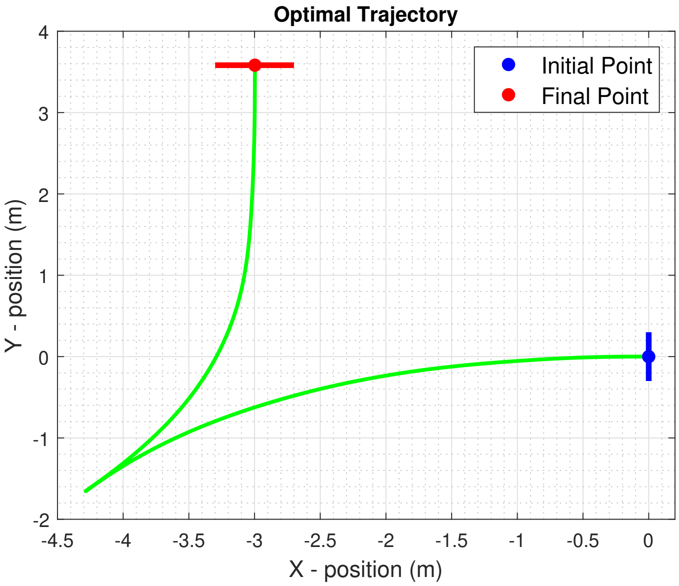


Figure 4.9: Optimal trajectory obtained from stage - 3

the velocity profile from stage - 2. Fig.4.9 shows the final path traversed by the wheel loader. The heading angle and steering angle profiles are also shown below in Fig.4.10.

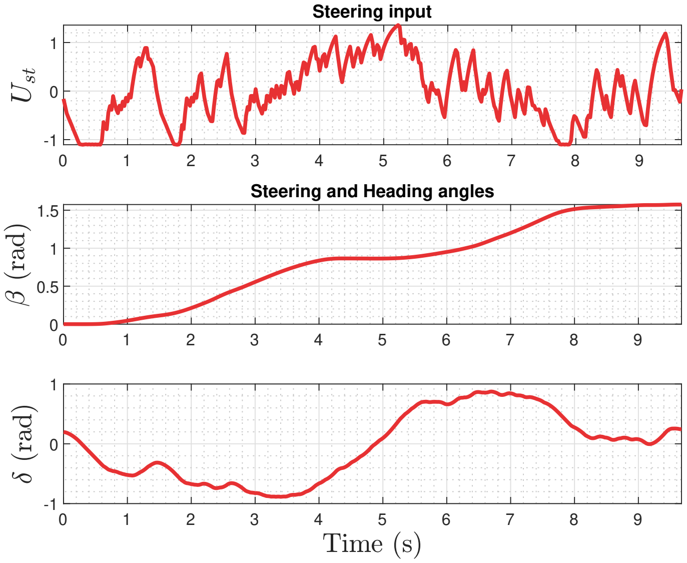


Figure 4.10: Steering and Heading angle profile from stage - 3

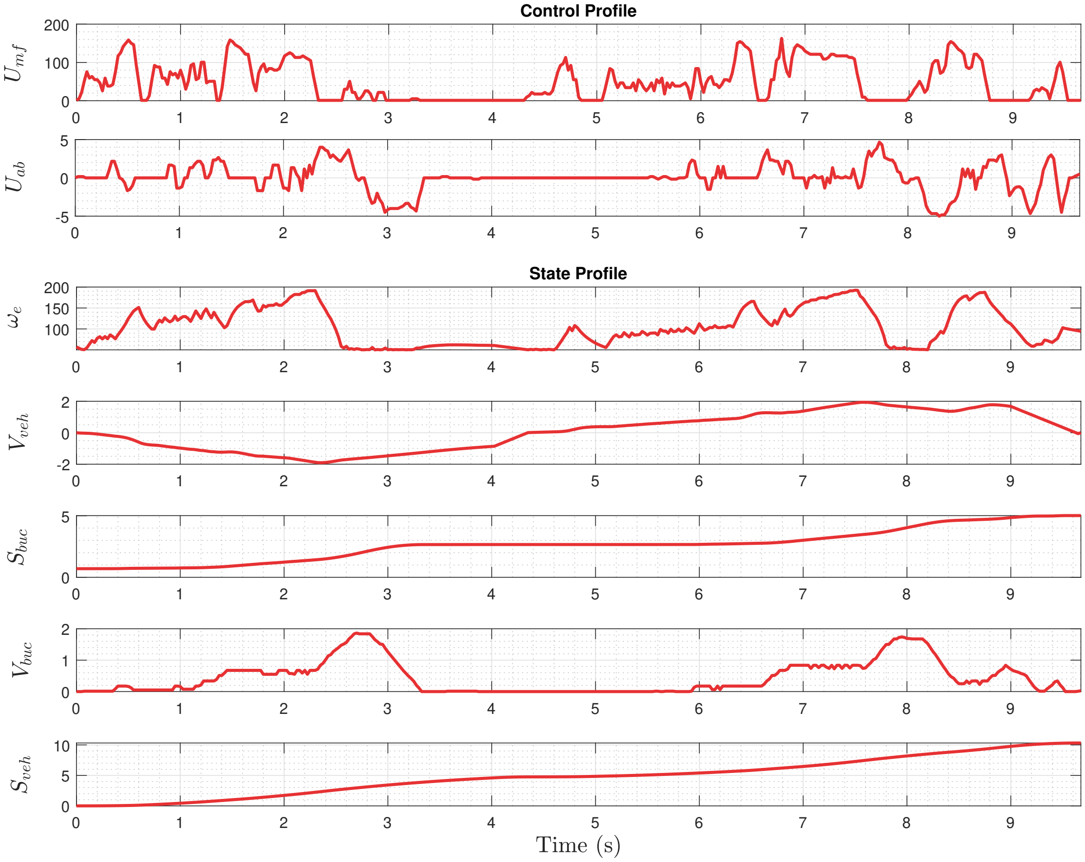


Figure 4.11: Optimal state and control profile obtained from stage - 2

5. Direct Method

5.1 Procedure

Numerical methods are divided into two categories - direct and indirect methods. Due to the disadvantages of indirect methods as described in the introduction section, we explore the DM in this research. GPOPS-II, PROPT are some of the widely used commercial software but require a license. We use a software named MPOPT (Multi-Phase Optimizer) [44], an extensible pseudo-spectral collocation based multiphase non-linear optimal control problem solver mainly it is open source and uses CasAdi, a widely-used open-source software framework for numerical optimization.

Accuracy is often better in DP and Indirect Methods compared to Direct Methods. NLP solvers search locally and converge to a local minimum and do not guarantee a global minimum. For faster runtimes and convergence, initial guess selection is vital. This method requires the constraint and objective functions to be smooth and their first and second derivatives must exist. In practice, lookup tables are often used based on practical data. Smooth functions have to be fit to the data obtained. Often, this leads us to a trade-off between bias and variance. A smooth function helps converge the algorithm faster but suffers a loss of accuracy. On the other hand, if a function is close to discontinuity, the solver takes a longer time to converge or may not converge at all. Lastly, the solution determined by numerical approaches are deterministic and usually provide open-loop controls. DP, on the other hand, converges globally and yields a closed-loop feedback control. This is particularly useful in a real-time implementation where there are model/parameter uncertainties, external disturbances, sensor noises, etc.

The procedure of the DM is briefly outlined here. Interested readers can refer to [19] for detailed procedure. The structure of the optimal control problem to be transcribed to NLP is as follows. Suppose, the control inputs $u(t)$ have to chosen to minimize the cost function

$$\phi = \min_{u_{mf}, u_{ab}, u_{st}} \quad \int_{0}^{t_1} \dot{m}_f dt + \int_{t_1}^{t_2} \dot{m}_f dt + \int_{t_2}^{t_3} \dot{m}_f dt + \int_{t_3}^{t_4} \dot{m}_f dt \tag{5.1}$$

subjected to

Dynamic Constraints:

$$\dot{x} = f(x, u, t) \tag{5.2}$$

Path Constraints:

$$g(x, u, t) \geq 0 \tag{5.3}$$

State Constraints:

$$x_{min} \leq x(t) \leq x_{max} \tag{5.4}$$

Input Constraints:

$$u_{min} \leq u(t) \leq u_{max} \tag{5.5}$$

where $x(t), u(t)$ are state and control vector at time t respectively.

The above continuous domain optimal control problem is transcribed into an NLP problem as following. We discretize the time domain (t_I, t_F) into n finite time intervals which yields

$$t_I < t_k < t_F \tag{5.6}$$

where $k = 1......, n - 1$ and each time interval h = $\frac{t_F - t_I}{n}$.

The state and control variables in the continuous domain become the variables to be optimised in the nonlinear program. Variables:

$$y = [x_1, u_1, x_2, u_2, ..., x_F, u_F] \tag{5.7}$$

Dynamic Constraints:

$$\frac{y_{k+1} - y_k}{h} = f(y_k, u_k, k) \tag{5.8}$$

Path Constraints:

$$g(y_k, u_k, k) \geq 0 \tag{5.9}$$

State and Input Constraints:

$$y_{min} \leq y_k \leq y_{max} \tag{5.10}$$

where y is a vector of all the state and control inputs at time steps t_k. The transcription shown here is for a single phase and can be easily extended to multiple phases. The reason for having multiple phases in this problem is because the vehicle dynamics is discontinuous with a $sign$ function. For NLP, the functions have to be smooth. Hence, we separate the problem into multiple phases and have smooth functions as required by the software framework. Here, dynamic constraints are discretized using an explicit algorithm for computational advantage. A more accurate algorithm such as RK4 can also be employed if more accuracy is needed or the model encounters convergence issues. Numerical solvers use higher-order discretization methods and not the explicit method for accuracy. This NLP can be solved using several open-source nonlinear optimization software routines such as SNOPT, IPOPT, KNITRO, etc.

The current OCP has already been solved using numerical approaches in [16]. The authors in this paper have used PROPT as the solver algorithm for this OCP. PROPT is a closed-source software and requires a license to be purchased. Hence, in this thesis, to explore both methods, we use MPOPT, an open-source method that transcribes the OCP into an NLP and has a builtin interface with CasAdi. CasAdi requires all the constraints and objective functions to be smooth. In this problem, the engine model has a max function which can be dealt with easily by introducing a path constraint. Another function to be dealt with is the torque converter model as it is a lookup table and hence needs to be smoothed. Fourier functions of higher order are curve fit to the torque converter profiles. Care should be taken that the extent of the curve fit should be close but at the same time not highly nonlinear and does not have local extremum. Since NLP does not suffer from the curse of dimensionality and is highly effective in dealing with higher-order optimal control problems, we consider the full system of 8 states and 3 control inputs rather than splitting the problem into stages as been done in DP. Since this is a non-convex problem, initial guess selection is often vital in determining the optimal solution and for convergence. Random initial guesses have been chosen and the best solution among the obtained results is selected for comparison. Fig.5.2 shows the best optimal state profile obtained from MPOPT. Fig. 5.1 shows the optimal trajectory.

5.2 Results:

Since DM is computationally efficient for higher order systems too unlike DP, there is no need for a split-stage optimization approach. The full order system is considered for the OCP and solved using MPOPT. The optimal state and control profile obtained from NLP varies based on the chosen initial guess. The profiles for both cases shown below are the best possible local optimum obtained by randomly varying initial guesses. Influence of initial guess on the optimal solution is analysed later in this chapter.

5.2.1 Case - 1:

The optimal control and state profiles are shown in Fig.5.2.

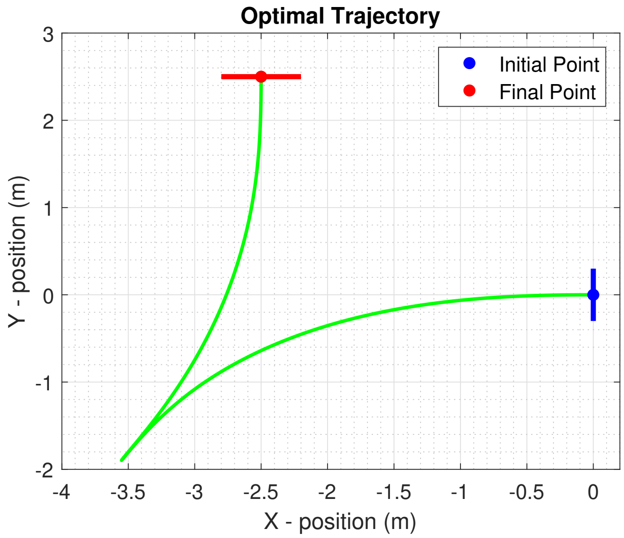


Figure 5.1: Optimal Trajectory - DM (Case - 1).

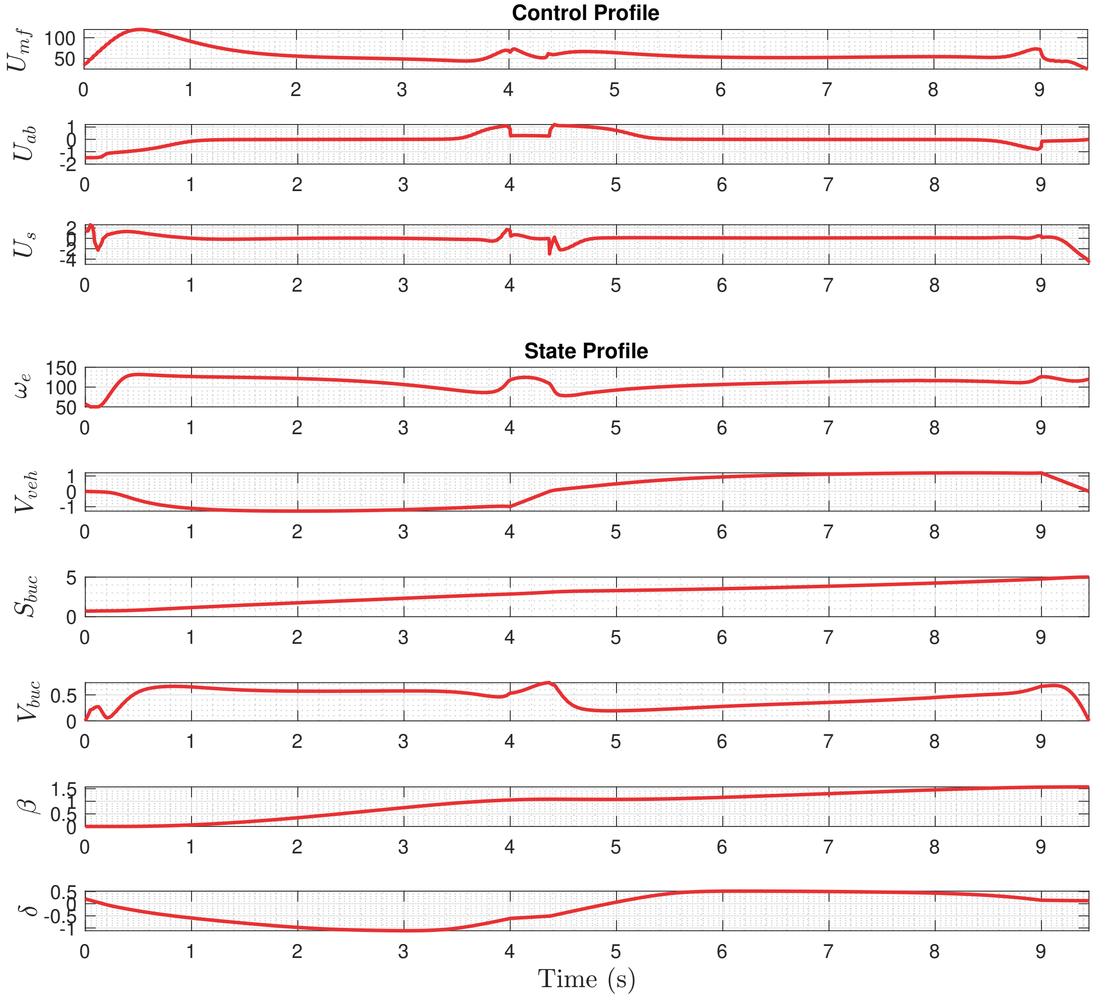


Figure 5.2: Optimal state and control profile obtained from DM (Case - 1).

5.2.2 Case - 2:

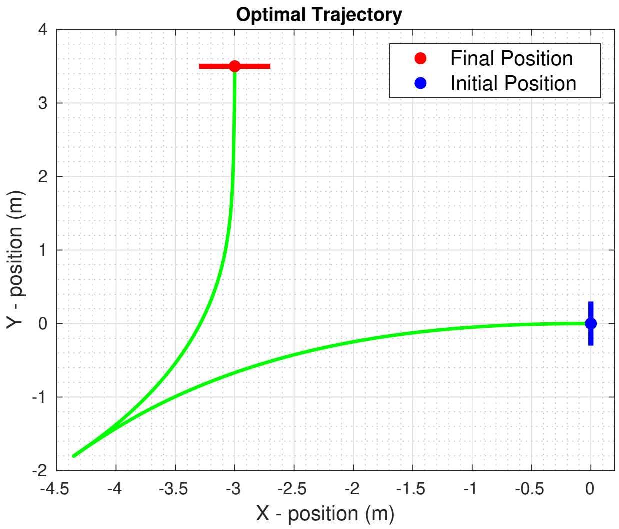


Figure 5.3: Optimal Trajectory - DM (Case - 2).

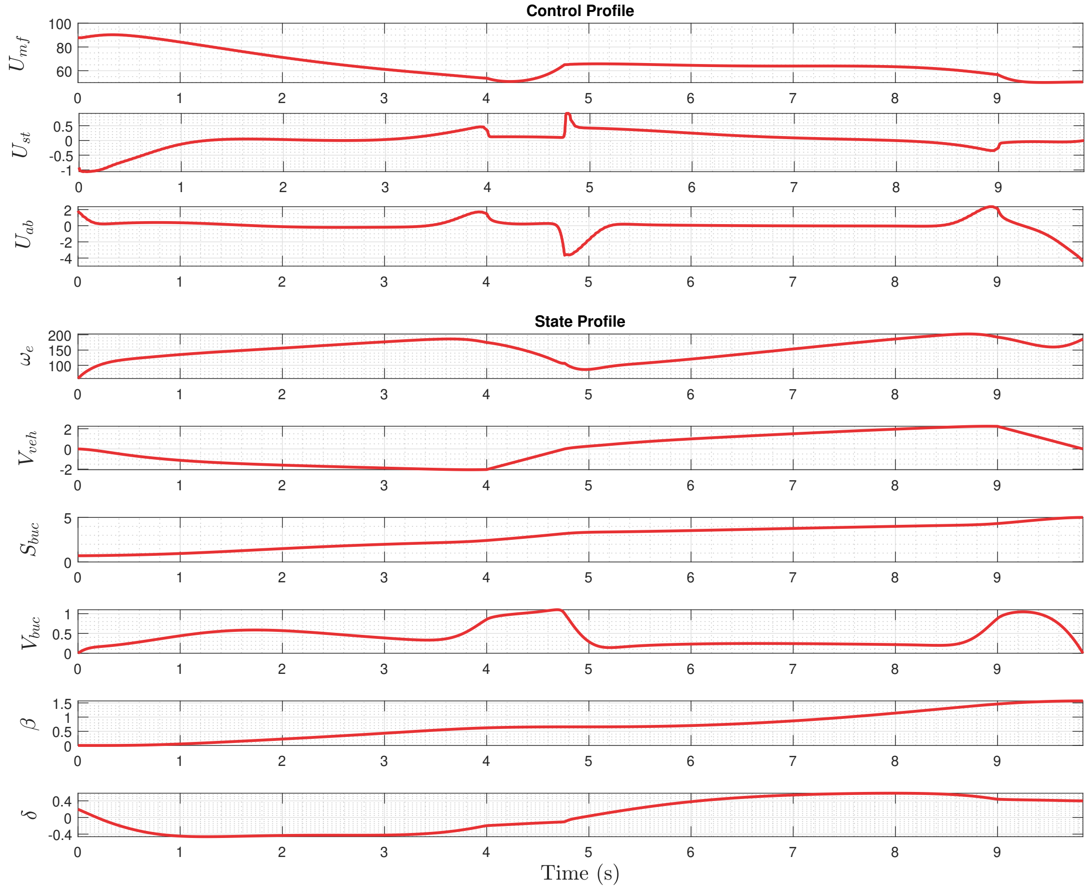


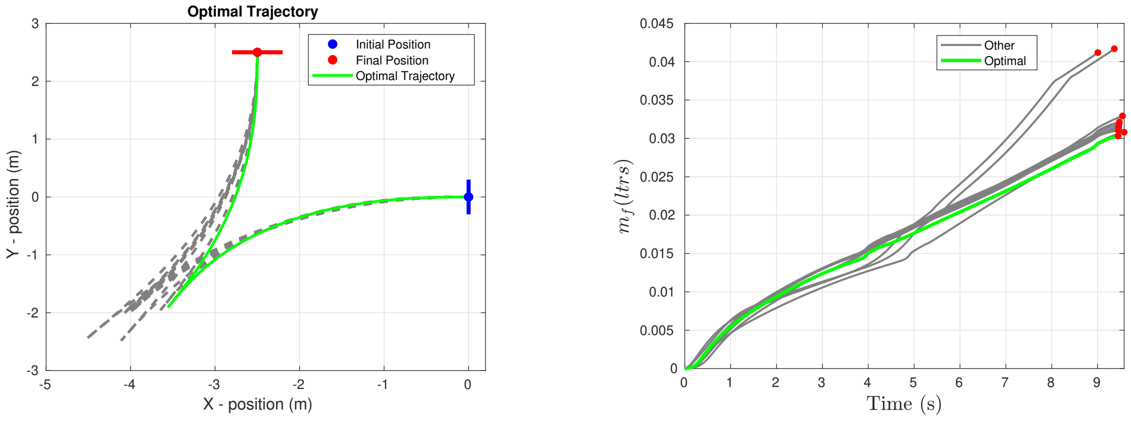
Figure 5.4: Optimal state and control profile obtained from DM (Case - 2).

5.3 Initial Guess Sensitivity

It is widely known that initial guess influences the convergence of a NLP to a local optimum. For interior point based methods, the chosen initial guess must be feasible. Ensuring an initial guess is also feasible is often difficult. In other methods, a bad or infeasible initial guess may lead the solution to diverge. For the current OCP, we consider several different initial guesses and analyse the convergence behavior of the NLP.

Fig.5.5 and 5.6 shows different trajectories obtained from various initial guesses in both cases. Longer trajectories means that the wheel loader has to traverse longer distances which would consume more fuel. One such path can be seen the Fig.5.5 and 5.6 which consumes significantly high amount of fuel compared to other trajectories. The key to avoid bad local minimum traps is to run the simulation through various initial guesses and pick the best optimum among all. This process would be computationally more expensive but yields the best possible local optimum. Hence, a good initial guess is often required to ensure the best local optimum.

Case - 1:

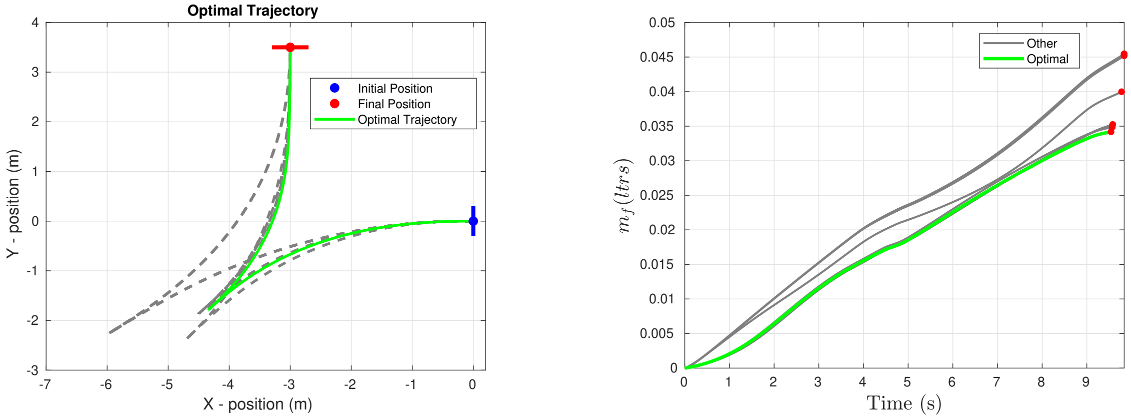


(a) Initial guess sensitivity on optimal trajectories

(b) Initial guess sensitivity on optimal fuel consumption

Figure 5.5: Influence of initial guess on optimal solution (Case - 1)

Case - 2:



(a) Initial guess sensitivity on optimal trajectories

(b) Initial guess sensitivity on optimal fuel consumption

Figure 5.6: Influence of initial guess on optimal solution (Case - 2)

Case No:	Variation
1	38 %
2	32 %

Table 5.1: Variation in fuel consumed based on initial guess

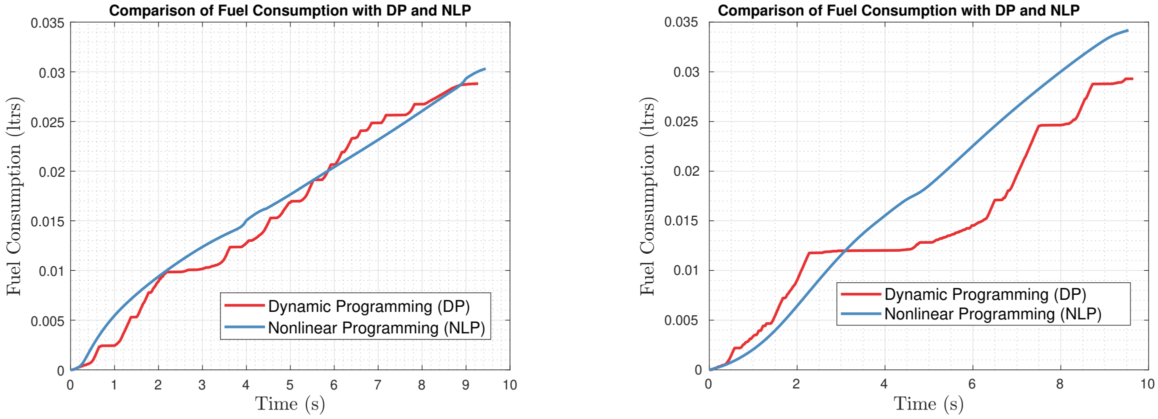
From figs 5.6 and 5.5, there is a significant difference in fuel consumption between the best local optimum and other optimum. Table.5.1 shows the maximum possible increase in fuel consumption with a bad initial guess compared to the best optimal fuel consumption.

6. SUMMARY AND CONCLUSIONS

6.1 Comparison

Dynamic Programming is a computationally intensive algorithm that performs an iterative search through the entire state and control space. The solution obtained from DP is expected to be globally optimal which means the optimal cost from DP should be approximately equal to or lower than the optimal cost from NLP. However, DP is practically feasible for lower-order systems. Considering the order of the optimal control problem in this thesis, we have split the problem into three stages. The effectiveness of the splitting strategy has to be evaluated as this can shorten the search region of DP and may not include the optimal result in the search profile.

• Fig. 6.1shows the comparison of the fuel consumed vs time for DP and NLP. Results show that fuel consumption by DP is lower than NLP by 7% in case - 1 and 15% in case - 2 respectively. The fuel consumption profile shown for NLP is the best that we have obtained from random initial guesses and ignoring the diverging cases. In the case of NLP, fuel consumption is fairly linear with time whereas, in DP, it varies significantly.

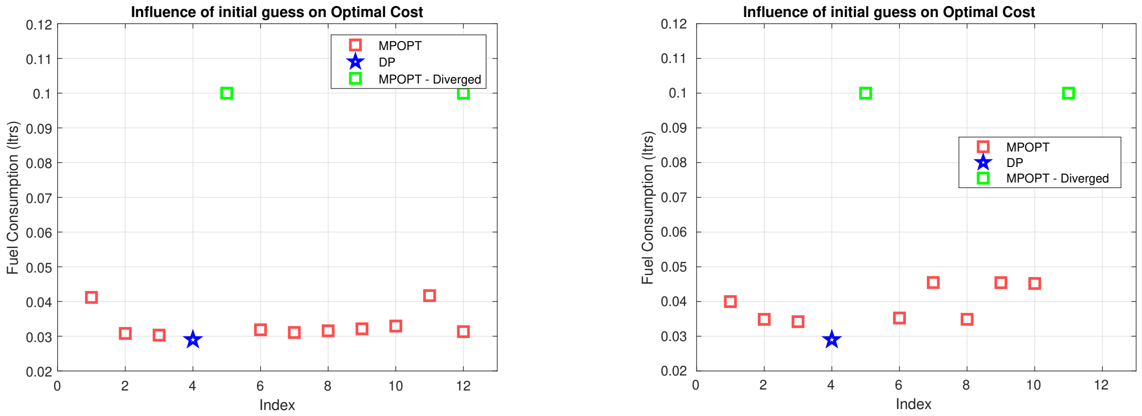


(a) Case - 1

(b) Case - 2

Figure 6.1: Fuel consumption in DP and NLP

• Also, problems with path constraints pose a challenge in robotics applications as observed [21]. Fig. 6.2 shows several other unique solutions obtained from NLP. For most cases, NLP converges fairly efficiently to a local minimum. However, the optimal cost and trajectory are different and dependent on the initial guess. The influence of the initial guess for the solution is highlighted on [32]. Convergence and a local minimum solution to an NLP problem can be highly sensitive to initial guesses. Even small problems can exhibit hundreds of unique solutions. For example, in the Fig. 6.2, two of the cases do not converge to a feasible point and the cost is expressed as a higher value i.e 0.1. Most of the cases converged and the optimal cost lies between 0.03 to 0.05 liters for a cycle. Comparing the best case scenario from MPOPT to DP, one might conclude that DP does not have any advantage over MPOPT. It must be taken into consideration that the solution obtained from MPOPT is the best-case scenario and not usual. The optimal cost value may not seem high initially. The energy savings are pretty high if we consider the number of vehicles and the loading cycles for each vehicle operating around the world.



(a) Case - 1

(b) Case - 2

Figure 6.2: Optimal Cost obtained from MPOPT (with different initial guesses) and DP.

• For real-time implementation, the NLP solution is deterministic and provides a single trajectory through state and control policy and not a global policy as obtained from DP. DP yields a feedback control solution that can be used directly for systems with imperfect models, parameter uncertainties and sensor noises, etc. On the other hand, NLP is an open-loop and has to be run in real-time where the algorithm has to run faster than the actual time step. Considering the convergence issues and locally optimal solutions and the order of the system, DP may be preferable as it can be run offline and can be incorporated into a stochastic system too.

• DP, however, is not beneficial for a longer time horizon such as optimization for only fuel consumption which is obtained in ≈ 27s as in [6]. DP becomes extremely tedious considering the time frame. In cases where the problem cannot be split and require higher simulation times, NLP may be a better choice considering the computational effort. However, in the cases with moderate simulation times, DP can be a better choice.

• Discretization errors can also become influential for highly nonlinear dynamics. To improve accuracy and stability, higher-order integration methods can be used. However, that would increase computational load manyfold, and considering DP's structure, it is not practically feasible. Thus for a highly nonlinear and a higher-order system, DP may not be a computationally viable option.

• A single simulation with DP works for a wide range of final conditions whereas Direct Method has to be modified and run again for the changed final conditions. In this thesis, the results in both the cases are obtained from a single DP run but the initial guesses in DM had to be modified to generate the optimal result.

The proposed method solving a higher dimensional optimal control problem using DP by split-stage optimization strategy shows great promise. Although it still suffers from the curse of dimensionality, this only has to be run once and offline. This thesis used forward DP methodology and this can be easily implemented in a backward fashion which yields an optimal feedback control policy that can be used for deterministic and stochastic systems as well.

6.2 Contribution

We proposed a method to solve a higher dimensional optimal control problem using a split-stage optimization approach using DP. The solution obtained is for a short loading cycle of a wheel loader system consisting of 8 states, 3 control inputs, and 4 phases. This method is an effective way to overcome the well-known issue, the curse of dimensionality that DP encounters while solving higher-order systems. This guarantees global optimal solution does not face convergence issues and also provides a global control policy.

6.3 Conclusions and Future work

The method proposed shows a great promise in the context of wheel loader optimization. The potential energy savings using this method is a lot considering the number of wheel loaders and the number of cycles in the construction and mining industries. Dynamic Programming is better than Direct methods such as NLP in ways such as global optimum, convergence and feedback control policy, etc. Direct methods on the other hand are better equipped to handle higher-order systems and are computationally efficient. The version of DP implemented in this thesis is a forward DP approach. To implement this method in practice in an efficient way, backward DP is a better choice as it yields a feedback control. Also, the analysis done in this thesis is for comparison and fixed times were chosen. We look to apply this algorithm for varying times and that may increase the computational complexity but that has a negligible effect on the time taken if run parallelly on multiple cores. Another possible way to efficiently apply DP is through the Reduced Order Model approach provided the desired accuracy is achieved. A ROM can significantly reduce the computational complexity and makes DP practically feasible whilst compromising on the accuracy. The model considered in this thesis is derived from [16] which is practically validated. DM might be a better option for longer run-time optimization as DP might be computationally infeasible. Also, in practice, we often deal with lookup tables rather than smooth functions. DP can deal with raw data fairly easily without any modifications. NLP, on the other hand, requires the functions and their gradients to be smooth. Fitting the data to a smooth function is often a tedious process that can although improve convergence and run-times but compromises the accuracy of the solution. This remains to be seen if this is the case observed. Thus, practical implementation of backward-based DP for varying times can be explored in a future study to see how both direct methods and DP fare against each other.

REFERENCES

[1] G. Kerschen, J.-C. Golinval, A. VAKAKIS, and L. BERGMAN, “The method of proper orthogonal decomposition for dynamical characterization and order reduction of mechanical systems: An overview,” *Nonlinear Dynamics*, vol. 41, pp. 147–169, 08 2005.

[2] T. Nilsson, A. Fröberg, and J. Åslund, “Minimizing fuel use during power transients for naturally aspirated and turbo charged diesel engines,” 2014.

[3] “Volvo 220h wheel loader brochure,” `https://www.volvoce.com/-/media/volvoce/global/products/wheel-loaders/wheel-loaders/brochures/brochure_l150h_l180h_l220h_t4f_en_na_22_voe2220009440.pdf?v=aA5EPw`.

[4] V. Nezhadali, B. Frank, and L. Eriksson, “Wheel loader operation—optimal control compared to real drive experience,” *Control Engineering Practice*, vol. 48, pp. 1 – 9, 2016.

[5] R. Filla, “Optimizing the trajectory of a wheel loader working in short loading cycles,” 06 2013.

[6] V. Nezhadali, L. Eriksson, and A. Fröberg, “Modeling and optimal control of a wheel loader in the lift-transport section of the short loading cycle,” *IFAC Proceedings Volumes*, vol. 46, no. 21, pp. 195 – 200, 2013. 7th IFAC Symposium on Advances in Automotive Control.

[7] “Use of energy in the industry,” `https://www.eia.gov/energyexplained/use-of-energy/industry.php`.

[8] “Us mining industry energy bandwidth study,” `https://www.energy.gov/sites/prod/files/2013/11/f4/mining_bandwidth.pdf`.

[9] F. Wang, M. A. M. Zulkefli, Z. Sun, and K. A. Stelson, “Energy management strategy for a power-split hydraulic hybrid wheel loader,” *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 230, no. 8, pp. 1105–1120, 2016.

[10] S. Sarata, W. Yossawee, and T. Tsubouchi, “Approach path generation to scooping position for wheel loader,” *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pp. 1809–1814, 2005.

[11] S. Sarata, H. Osumi, Y. Kawai, and F. Tomita, “Trajectory arrangement based on resistance force and shape of pile at scooping motion,” in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA ’04. 2004*, vol. 4, pp. 3488–3493 Vol.4, 2004.

[12] B. Hong and X. Ma, “Path planning for wheel loaders: A discrete optimization approach,” in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1–6, 2017.

[13] R. Zhang, A. Alleyne, and E. Prasetiawan, “Modeling and h_2/h_∞ mimo control of an earthmoving vehicle powertrain,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 124, no. 4, pp. 625–636, 2002.

[14] B. Frank, J. Kleinert, and R. Filla, “Optimal control of wheel loader actuators in gravel applications,” *Automation in Construction*, vol. 91, pp. 1–14, 7 2018.

[15] V. Nezhadali and L. Eriksson, “Optimal control of wheel loader operation in the short loading cycle using two braking alternatives,” in *2013 IEEE Vehicle Power and Propulsion Conference (VPPC)*, pp. 1–6, 2013.

[16] V. Nezhadali and L. Eriksson, “Optimal lifting and path profiles for a wheel loader considering engine and turbo limitations,” *Lecture Notes in Control and Information Sciences*, vol. 455, pp. 301–324, 03 2014.

[17] R. Bellman, *Dynamic Programming*. Dover Publications, 1957.

[18] J. T. Betts, “Survey of numerical methods for trajectory optimization,” *Journal of Guidance, Control, and Dynamics*, vol. 21, no. 2, pp. 193–207, 1998.

[19] A. Rao, “A survey of numerical methods for optimal control,” *Advances in the Astronautical Sciences*, vol. 135, 01 2010.

[20] L. Paiva, *Numerical Methods in Optimal Control and Model Predictive Control.* PhD thesis, 12 2014.

[21] B. Passenberg, "Theory and algorithms for indirect methods in optimal control of hybrid systems," 2012.

[22] H. Febbo, P. Jayakumar, J. L. Stein, and T. Ersal, "Nloptcontrol: A modeling language for solving optimal control problems," *CoRR*, vol. abs/2003.00142, 2020.

[23] P. E. Gill, W. Murray, and M. A. Saunders, "Snopt: An sqp algorithm for large-scale constrained optimization," *SIAM Review*, vol. 47, no. 1, pp. 99–131, 2005.

[24] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi – A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.

[25] P. Rutquist and E. Marcus, "Propt — matlab optimal control software," 01 2010.

[26] M. Patterson and A. Rao, "Gpops-ii," *ACM Transactions on Mathematical Software (TOMS)*, vol. 41, pp. 1 – 37, 2014.

[27] I. M. ROSS, "A beginner's guider to dido: A matlab application package for solving optimal control problem," *http://www. elissar. ziz (2007)*, 2007.

[28] S. Kirkpatrick, C. Gelatt, and M. Vecchi, "Optimization by simulated annealing," *Science (New York, N.Y.)*, vol. 220, pp. 671–80, 06 1983.

[29] Z. Michalewicz, "Genetic algorithms + data structures = evolution programs," in *Artificial Intelligence*, 1992.

[30] L. Ingber, "Simulated annealing: Practice versus theory," *Mathematical and Computer Modelling*, vol. 18, pp. 29–57, 12 1993.

[31] O. Nelles, *Nonlinear System Identification*, vol. 13. 01 2001.

[32] S. Campbell, J. Betts, and C. Digirolamo, "Initial guess sensitivity in computational optimal control problems," 10 2018.

[33] D. Bertsekas, *Dynamic Programming and Optimal Control.* 1995.

[34] J. Wahlström and L. Eriksson, “Modelling diesel engines with a variable-geometry turbocharger and exhaust gas recirculation by optimization of model parameters for capturing non-linear system dynamics,” *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 225, no. 7, pp. 960–986, 2011.

[35] X. Liu, D. Sun, D. Qin, and J. Liu, “Achievement of fuel savings in wheel loader by applying hydrodynamic mechanical power split transmissions.,” 2017.

[36] D. Hrovat and W. Tobler, “Bond graph modeling and computer simulation of automotive torque converters,” *Journal of the Franklin Institute*, vol. 319, no. 1, pp. 93 – 114, 1985.

[37] L. Guzzella and A. Sciarretta, *Vehicle Propulsion Systems: Introduction to Modeling and Optimization.* 01 2007.

[38] B. Alshaer, T. Darabseh, and M. Alhanouti, “Path planning, modeling and simulation of an autonomous articulated heavy construction machine performing a loading cycle,” *Applied Mathematical Modelling*, vol. 37, p. 5315–5325, 04 2013.

[39] H. Takahashi and Y. Konishi, “Path generation for autonomous locomotion of articulated steering wheel loader,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 16, no. 3, pp. 159–168, 2001.

[40] F. Goppelt, T. Hieninger, and R. Schmidt-Vollus, “Modeling centrifugal pump systems from a system-theoretical point of view,” 12 2018.

[41] M. Ruderman, “Full- and reduced-order model of hydraulic cylinder for motion control,” 05 2017.

[42] M. Kelly, “An introduction to trajectory optimization: How to do your own direct collocation,” *SIAM Rev.*, vol. 59, p. 849–904, Jan. 2017.

[43] D. P. Bertsekas, “Dynamic programming and optimal control, vol. ii,” 1976.

[44] D. Thammisetty, “Development of a multi-phase optimal control software for aerospace applications (mpopt),” 2021.

APPENDIX A

Parameters	Value
J_e	$2.5\ Kgm^2$
w_1	101.38
w_2	-0.655
w_3	0.009
w_4	14.12
w_5	0
r_w	$0.7\ m$
c_1	0.0280
c_2	0.0126
a_1	2
a_2	2
c_{roll}	0.03
M_{total}	$42000\ Kg$
η_{gb}	0.9
L	$3.7\ (m)$
C_{st}	100
M_{load}	$10000\ Kg$
η	0.9

Table A.1: Parameters of the WL model.