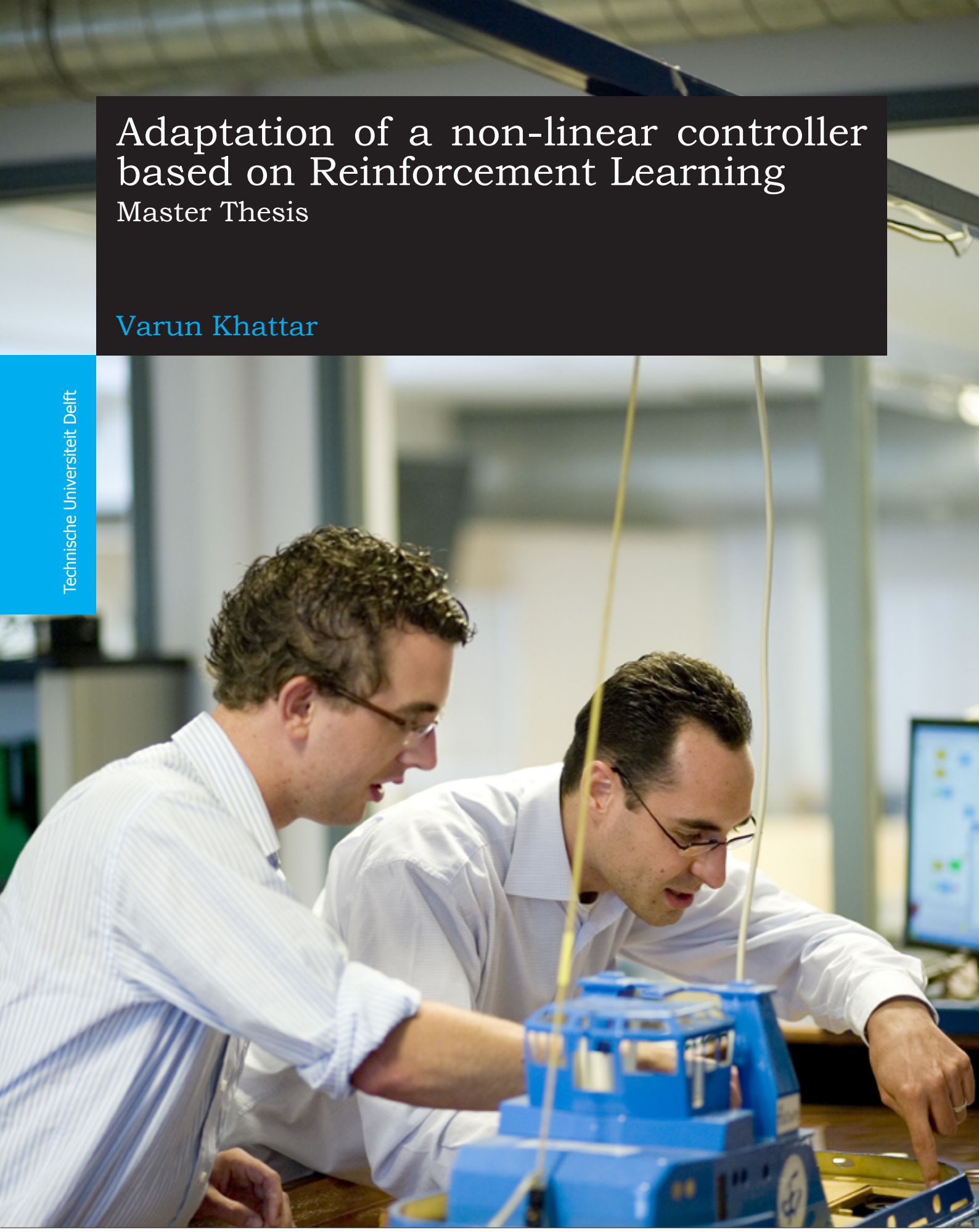
Adaptation of a non-linear controller based on Reinforcement Learning

Master Thesis

Varun Khattar

Technische Universiteit Delft



TU Delft Delft University of Technology

Challenge the future

Adaptation of a non-linear controller based on Reinforcement Learning

Master Thesis

by

Varun Khattar

in partial fulfillment of the requirements for the degree of

Master of Science
in Mechanical Engineering
Track: Vehicle Engineering
Specialization: Dynamics and Controls

at the Delft University of Technology,
to be defended publicly on Wednesday October 31, 2018 at 10:00 am.

Supervisor:	Prof. Dr. Robert Babuska (HL)
Thesis committee:	Dr. Barys Shyrokau (UD)
	Dr. Carlos Celemin Paez (PD)

TU Delft Delft University of Technology

Acknowledgements

I want to thank Professor Robert Babuska for giving me the opportunity to write this thesis under his supervision and his help, guidance and endless patience with my mistakes. His insights have been crucial in leading to successful results.

I also want to thank Dr. Barys Shyrokau and Dr. Jens Kober for their suggestions and help.

I want to thank Dr. Carlos Celemin Paez for agreeing to be part of the thesis committee on such a short notice.

I am grateful to Dr. Riender Happee for letting me know about the opportunity to work under Professor Babuska.

Finally, I want to thank my girlfriend, family and friends, without whose support this would not have possible.

Varun Khattar

Contents

1 Introduction **1**
1.1 Goal of the thesis . . . 2
1.2 Outline . . . 2

2 Preliminaries **3**
2.1 Model-Based Reinforcement Learning . . . 3
2.2 The Bellman equation . . . 4
2.3 Fuzzy V Iteration . . . 4
2.4 Robust methods for value function estimation . . . 4
2.4.1 Average method . . . 5
2.4.2 Max-Min method . . . 5
2.5 Policy Formulation . . . 5
2.5.1 Hill climbing policy . . . 5
2.5.2 Interpolated policy . . . 5
2.6 Actor-only RL methods . . . 6
2.6.1 Standard policy gradient . . . 6
2.6.2 Natural policy gradient . . . 7
2.6.3 Episode based actor-only methods . . . 7
2.7 Policy Learning by Weighting Exploration with the returns (PoWER) . . . 8
2.8 Summary . . . 10

3 ABS model and baseline controller **11**
3.1 Single corner wheel dynamics . . . 11
3.2 Pacejka Tire Model . . . 12
3.3 Current control methods for ABS . . . 13
3.3.1 Performance Criteria . . . 15
3.4 Proportional-Integral controller for pure slip control . . . 15
3.4.1 Proportional control with gain scheduling . . . 19
3.5 Summary . . . 19

4 ABS control through offline Reinforcement Learning **21**
4.1 Fuzzy-V iteration and system parameters . . . 21
4.2 Results for non-linear interpolation . . . 22
4.2.1 Initial speed = 80 km/h . . . 22
4.3 Results for piecewise linear approximated policy . . . 25
4.3.1 Initial speed = 80 km/h . . . 26
4.3.2 Initial speed = 60 km/h . . . 29
4.4 Summary . . . 30

5 Policy and PI controller adaptation results **31**
5.1 Algorithm parameters . . . 31
5.2 Piecewise linear policy adaptation results . . . 32
5.2.1 Adaptation of wet asphalt policy to dry asphalt . . . 33
5.2.2 Adaptation of average policy to dry asphalt . . . 35
5.2.3 Adaptation of dry asphalt policy to wet asphalt . . . 38
5.2.4 Adaptation of average policy to wet asphalt . . . 40
5.3 Adaptive Proportional-Integral (PI) control . . . 43
5.3.1 Fixed slip setpoint, adaptive proportional gain and integral gain . . . 44
5.3.2 Adaptive slip setpoint, proportional gain and integral gain . . . 46
5.3.3 Adaptive slip setpoint, fixed proportional gain and integral gain . . . 49
5.3.4 Adaptive slip setpoint, fixed proportional gain . . . 51
5.3.5 Adaptive slip setpoint and proportional gain . . . 54
5.4 Summary . . . 56

6 Conclusions and Future Work **57**
6.1 Conclusions 57
6.2 Future Work 58
6.2.1 ABS (Anti-lock Braking System) 58
6.2.2 Lateral and vertical stability control of a vehicle 58
6.2.3 Motion control of a robot 58

References **59**

List of symbols

Description	Variable	Unit
State of the agent	x	-
State space domain	$\mathcal{X}$	-
Control input/ action	u	-
Control input/ action domain	$\mathcal{U}$	-
State transition function	$f(\cdot)$	-
Reward function	$\rho(\cdot)$	-
Total reward/ return	R	-
Control policy	π	-
Discount factor	γ	-
Value function	$V(\cdot)$	-
Optimal value function	$V^*(\cdot)$	-
Vector of value function parameters	θ	-
Vector of basis functions for value function	ϕ_f	-
Threshold for fuzzy V-iteration	ϵ_f	-
Number of transition models	n_m	-
Vector of ideal control inputs/ actions	p	-
Vector of policy parameters	ψ	-
Vector of basis functions for control policy	φ	-
Total Cost	J	-
Learning rate of actor	α_a	-
Riemannian metric tensor	$G(\tilde{\psi})$	-
Trajectory/ Episode/ Rollout	τ_r	-
k^{th} rollout	k_r	-
Number of rollouts	K	-
State-action value function	$Q(\cdot)$	-
Random Gaussian exploration	ε	-
Number of time steps in episode	N	-
k^{th} time step	k	-
Covariance matrix	$\hat{\Sigma}$	-
PoWER metric	M	-
Reward offset	ρ_{off}	m
Braking distance	d_x	m
Longitudinal wheel slip	κ	-
Linear wheel/chassis velocity	v_x	m/s
Wheel angular speed	ω	rad/s
Effective wheel rolling radius	r_t	m
Normalized wheel deceleration	η_w	-
Wheel mass moment of inertia	J_w	kgm^2
Wheel angular acceleration	$\dot{\omega}$	rad/s^2
Single corner mass	m	kg
Linear wheel/chassis acceleration	a_x	m/s^2
Acceleration due to gravity	g	m/s^2
Gravitational force due to slope	F_g	N
Longitudinal force on the tire	F_x	N
Braking torque on the tire	T_b	Nm
Lateral force on the tire	F_y	N
Vertical force on the tire	F_z	N

Description	Variable	Unit
Self aligning torque on the tire	M_z	Nm
Camber angle	γ_c	rad
Slip angle	α_s	rad
Curve fitting constants	B, C, D, E	-
Mixed slip control variable	ϵ_w	-
Reference longitudinal wheel slip	$\overline{\kappa}$	-
Reference normalized wheel deceleration	$\overline{\eta}_w$	-
Reference mixed slip control variable	$\overline{\epsilon}_w$	-
Mixed slip control constant	α_w	-
Braking torque function	$\zeta(\cdot)$	Nm
Maximum Braking Torque	T_b	Nm
Proportional gain	K_p	-
Integral gain	K_i	-
Constants	c_1, c_2, c_3, c_4	-

1

Introduction

Closed-loop control systems, which utilize output signals for feedback to generate control inputs, can achieve high performance and robustness against system changes and uncertainties. However, robustness of feedback control loops can be lost if system changes and uncertainties are too large. Adaptive control combines the traditional feedback structure with providing adaptation mechanisms that adjust a controller for a system with parametric, structural and environmental uncertainties to achieve desired system performance [1]. The design of control methods like PID, pole placement, optimal or nonlinear control methods is based on certain knowledge of the system parameters. In contrast, adaptive controllers do not need such knowledge. They adapt to parameter uncertainties by using performance error information on line. They can also be implemented on top of robust and optimal control methods, which are of fixed gain nature. While robust control is a powerful method to overcome parameter variations of the system model, it also depends on the range of uncertainty domain itself [2]. For example, sometimes a large amount of uncertainty can be handled, while another time only a small amount of uncertainty can be accommodated.

Interest in adaptive controls started growing in the 1950s due to the need of high performance flight control systems. Model reference adaptive control (MRAC) was developed to solve the autopilot problem in [3]. An adaptive pole placement scheme based on the optimal linear quadratic problem was given by [4]. State space techniques and stability theory based on Lyapunov were introduced in the 1960s. Developments in Dynamic Programming [5] and stochastic control, system identification and parameter estimation reformulated adaptive control methods. In the 1970s, the development and progress in computers and electronics made the implementation of complex controllers feasible and contributed to an increased interest in applications of adaptive control. A Lyapunov stability based design approach was used to design and analyze MRAC schemes [6]. By the mid 1980s, robust adaptive control methods had been developed to counteract the lack of robustness in adaptive control. In the 1990s, efforts were made to extend results for linear systems to non linear systems. Also, neural networks were used approximators of unknown nonlinear functions, which led to the use of online parameter estimators to train or update the weights of the neural networks. In the 2000s, Reinforcement Learning (RL) had started being used for adaptive control.

A major constraint for adaptive controllers is that to adapt to uncertainties and converge to new parameters successfully, signals which are generated inside the time varying feedback loop of the unknown plant i.e. the regression vectors, must be persistently exciting. In practice, even after ensuring persistence of excitation, the estimated parameter may converge to some unexpected position due to measurement noise. Another drawback is that they do not always adapt well to non-linear systems. Also, most of the adaptive control methods work well primarily for linear systems. In addition, there are always unmodeled dynamics at high frequencies, which lead to lack of initial stability.

In this thesis, RL has been used for an adaptive controller for an Anti-Lock Braking System (ABS) controller. In contrast to [7], which gives a data-driven method to apply model free Q-learning for ABS control, model based RL has been used here. Model based Adaptive Dynamic Programming (ADP) using Value Iteration has also been used in [8] for wheel slip control. But the difference is that [8] uses a neural network as the value function approximator, while this thesis uses a different function approximator for

the value function, which is a simplified implementation of the method given by [9]. This is because the disadvantage of approximators like expansions with fixed or adaptive basis functions, regression trees, local linear regression and deep neural networks is that they are difficult to tune for convergent learning. Another difference is that the focus in [8] is to find an optimal switching schedule, while the focus of this thesis is to obtain a smooth control input with minimum chattering and steady state error. [9] uses fuzzy sets, which also have been used by [10] for ABS control. The advantage of fuzzy control is that it is able to handle the uncertainties in the slip dynamics very well.

1.1. Goal of the thesis

Given a baseline nonlinear controller of an ABS (Anti-lock Braking System) of a passenger car, obtained as a symbolic approximation of an optimal control law derived as a solution to the Bellman equation, we want to develop a robust and convergent method to adapt online its parameters to dry asphalt and wet asphalt, in order to take care of mild process variations or a model-plant mismatch e.g the road surface might suddenly change from wet asphalt to dry asphalt or vice versa.

1.2. Outline

The thesis has been outlined in the following way:

1. Chapter 2 explains the preliminaries i.e. model based Reinforcement Learning, the Bellman equation and the RL methods used for robustness and adaptation.
2. Chapter 3 explains the wheel dynamics, tire model, current control principles and the performance criteria used for ABS. It also shows results for a hand tuned Proportional-Integral (P-I) controller and a hand tuned proportional controller.
3. Chapter 4 explains the application procedure of RL to ABS and the results for initial robustness of the controller on dry asphalt and wet asphalt.
4. Chapter 5 explains the details of the adaptation of the control policy and results for the adaptations of both nominal and robust policies to dry asphalt and wet asphalt. It also shows results for adaptive P-I control e.g. adaptive or fixed proportional and integral gains with fixed or adaptive slip setpoint.
5. Chapter 6 explains the conclusions and the future work possible.

2

Preliminaries

This chapter presents the concepts of Reinforcement Learning that will be used to obtain the control policy, and make it robust and adaptive.

2.1. Model-Based Reinforcement Learning

Reinforcement learning (RL) is a machine learning method that can be used for self tuning adaptive control and optimal control. It is based on the model used by humans for learning. In the RL setting, there is an agent and an environment, and they are explicitly separated from each other. The environment represents the system in which the task is defined. The agent is a decision maker, whose goal it is to accomplish the task. The problem is solved by letting the agent interact with the environment. Each action of the agent changes the state of the environment. The environment responds by giving the agent a reward for what it has done. Based on this reward, the agent adapts its behavior. The agent then observes the state of the environment and determines what action it should perform next. In this way, the agent learns to act, such that its reward is maximized or minimized. RL can be model-based or model-free. If the model of the system is unknown, RL is model-free. In this thesis, model-based RL has been used since the dynamic system of interest is known to be described by the state transition function:

$$x_{k+1} = f(x_k, u_k) \tag{2.1}$$

where $x_k, x_{k+1} \in \mathcal{X} \subset \mathbb{R}^n$ are the current and next state respectively, and $u_k \in \mathcal{U} \subset \mathbb{R}^m$ is the current input. This function does not have to be stated by explicit equations; it can be e.g. a generative model given by a numerical simulation of complex differential equations. The control goal is specified through a reward function which assigns a scalar reward $r_{k+1} \in \mathbb{R}$ to each state transition from x_k to x_{k+1}:

$$r_{k+1} = \rho(x_k, u_k, x_{k+1}) \tag{2.2}$$

This function is defined by the user and typically calculates the reward based on the difference between the current state and a given constant reference state x_r that should be attained. The goal is to find an (approximately) optimal control policy $\pi : \mathcal{X} \to \mathcal{U}$ such that in each state it selects a control action so that the expected cumulative discounted reward over time, called the return, is maximized:

$$R^\pi = E\left\{\sum_{k=0}^{\infty} \gamma^k \rho\big(x_k, \pi(x_k), x_{k+1}\big)\right\} \tag{2.3}$$

Here $\gamma \in [0, 1]$ is a discount factor and the initial state x_0 is drawn uniformly from the state space domain $\mathcal{X}$ or its subset. The return is approximated by the value function $V^\pi : \mathcal{X} \to \mathbb{R}$ defined as:

$$V^\pi(x) = E\left\{\sum_{k=0}^{\infty} \gamma^k \rho\big(x_k, \pi(x_k), x_{k+1}\big) \,\middle|\, x_0 = x\right\} \tag{2.4}$$

2.2. The Bellman equation

Equation 2.4 can be written as a recursive equation between the value function at one time step and the value function at the next time step, known as the Bellman equation [5]:

$$V(x) = \rho\big(x, u, f(x,u)\big) + \gamma V\big(f(x,u)\big) \tag{2.5}$$

An approximation of the optimal V-function, denoted by $\hat{V}^*(x)$, can be computed by solving the Bellman optimality equation:

$$\hat{V}^*(x) = \max_{u \in \mathcal{U}} \Big[\rho\big(x, u, f(x,u)\big) + \gamma \hat{V}^*\big(f(x,u)\big)\Big]. \tag{2.6}$$

In discrete state and action space, the number of states and actions are finite. With continuous-valued state and input spaces, the number of states and actions are infinite. Policy iteration and value iteration cannot be used directly to solve the Bellman equations. A solution can be found if one obtains an approximation to the optimal value function instead of the exact optimal value function i.e. using approximate dynamic programming (ADP) or Reinforcement learning (RL) [11]. As given in [9], function approximators like expansions with fixed or adaptive basis functions, regression trees, local linear regression and deep neural networks can be used to represent the model and its dynamics i.e. policy mappings. The disadvantage of these approximators is that they are difficult to tune for convergent learning and can also affect the control performance negatively e.g. chattering control signals and steady state errors. Thus, focus is required not only on finding the control policy but also on achieving good control performance.

2.3. Fuzzy V Iteration

To obtain the optimal value function $V^*(x)$, [9] uses Fuzzy-V iteration method based on Fuzzy Q iteration [12]. Triangular membership functions are defined which are centered at points $C = \{c_1, c_2, ..., c_N\}$ distributed over a rectangular grid in state space such that

$$\begin{aligned} \phi_{f,j}(c_i) &= 1 \quad \text{for} \quad j = i \\ \phi_{f,j}(c_i) &= 0 \quad \text{for} \quad j \neq i \end{aligned} \tag{2.7}$$

The functions are normalized i.e. $\sum_{j=1}^{N} \phi_{f,j}(x) = 1$ and for a state variable j are defined as:

$$\begin{aligned} \phi_{f,1}(x_j) &= \max\left(0, \min\left(1, \frac{c_2 - x_j}{c_2 - c_1}\right)\right) \\ \phi_{f,N_j}(x_j) &= \max\left(0, \min\left(1, \frac{x_j - c_{N_j-1}}{c_{N_j} - c_{N_j-1}}\right)\right) \\ \phi_{f,i}(x_j) &= \max\left(0, \min\left(\frac{x_j - c_{i-1}}{c_i - c_{i-1}}, \frac{c_{i+1} - x_j}{c_{i+1} - c_i}\right)\right) \end{aligned} \tag{2.8}$$

for $i = 2, 3, 4, ...N_{j-1}$. The value function is approximated as $V(x) = \theta^\top \phi_f(x)$ where $\phi_f = [\phi_{f,1}\ \phi_{f,2}, ...\phi_{f,N}]^\top$, and $\theta = [\theta_1\ \theta_2\ .....\theta_N]^\top$ is a parameter vector found by the iteration

$$\theta_i \leftarrow \max_{u \in \mathcal{U}} \Big[\rho\big(c_i, u, f(c_i,u)\big) + \gamma \theta_i^\top \phi_f\big(f(c_i,u)\big)\Big] \quad \text{until} \quad \|\theta_l - \theta_{l-1}\|_\infty \leq \epsilon_f \tag{2.9}$$

where $\rho(\cdot)$ is the reward function, $f(\cdot)$ is the state transition function, $\gamma < 1$ is the discount factor, $\mathcal{U} = \{u_1, u_2, u_3....u_M\}$ and ϵ_f is a user defined threshold.

2.4. Robust methods for value function estimation

The value function can be obtained in a robust way to account for uncertainties in the transition model. Also, due to multiple possible transition models with each having its own uncertainty, one input for the current state can lead to multiple states:

$$x_{k+1,j} = f_j(x_k, u_k) \quad \text{for } j = 1, 2, ...n_m \tag{2.10}$$

where n_m is the number of transition models. If the controller is trained for all such transitions or a combination of such transitions, initial robustness before adaptation can be achieved.

Algorithm 1: Fuzzy V - Iteration for discrete actions and continuous states

Input: x_0, ρ, γ, ϵ_f, $\mathcal{X}$, $\mathcal{U}$

Define $C = \{c_1, c_2, ..., c_N\}$ for each state variable in $\mathcal{X}$
Define $\phi_f = [\phi_{f,1}\ \phi_{f,2}, ...\phi_{f,N}]^\top$ with $\sum_{j=1}^{N} \phi_{f,j}(x) = 1$ for each state variable in $\mathcal{X}$
$\theta_0 \leftarrow 0$
$l \leftarrow 0$
do
  for $i = 1 \dots N$ **do**
    $\theta_i \leftarrow \max_{u \in \mathcal{U}} \left[\rho\big(c_i, u, f(c_i, u)\big) + \gamma \theta^\top \phi_f\big(f(c_i, u)\big)\right]$
  end
  $l \leftarrow l + 1$
while $\|\theta_l - \theta_{l-1}\|_\infty \leq \epsilon_f$;

Output: $\theta^* = \theta_k$

2.4.1. Average method

The optimal value function can be obtained from maximizing the average of the RHS of the Bellman equation of all the models involved to make the controller more robust:

$$V^*(x) = \max_{u \in \mathcal{U}} \frac{1}{n_m} \sum_{j=1}^{n_m} \left[\rho(x, u, f_j(x, u)) + \gamma V^*(f_j(x, u))\right] \tag{2.11}$$

2.4.2. Max-Min method

Another way to make the controller robust is to obtain the optimal value function by choosing the maximum of the minimum of the RHS of the Bellman equation among all the models i.e maximize performance for the worst case:

$$V^*(x) = \max_{u \in \mathcal{U}} \left(\min_{f_1, f_2, \dots f_{n_m}} \left(\left[\rho(x, u, f_1(x, u)) + \gamma V^*(f_1(x, u))\right], \dots \dots, \left[\rho(x, u, f_{n_m}(x, u)) + \gamma V^*(f_{n_m}(x, u))\right] \right) \right) \tag{2.12}$$

2.5. Policy Formulation

There are two primary methods to derive the control policy from the value function [9]: online and offline maximization.

2.5.1. Hill climbing policy

The first method is based on an online maximization of the RHS of the Bellman optimality equation:

$$u^* = \underset{u \in \mathcal{U}}{\operatorname{argmax}} \left[\rho(x, u, f(x, u)) + \gamma V\left(f(x, u)\right)\right] \tag{2.13}$$

The advantage of this method is that stability is guaranteed since it is hill climbing the Lyapunov function [11]. The first disadvantage is that maximization is a computationally intensive process; using the simplest method produces only discrete actions. The second disadvantage is that the process model must be available for online use; if the process model is computationally intensive, then the process takes more time.

2.5.2. Interpolated policy

The second method applies the Bellman equation off-line and uses basis functions to interpolate online (interpolated policy). This method is used for this thesis. For all states c_i, $i = 1, 2, \dots, N$, the optimal

control action p_i is computed offline:

$$p_i = \underset{u \in \mathcal{U}}{\operatorname{argmax}} \left[\rho(c_i, u, f(c_i, u)) + \gamma \theta^\top \phi_f \left(f(c_i, u) \right) \right] \tag{2.14}$$

and the control actions are collected in a vector: $p = \left[p_1, \ldots, p_N \right]^\top \in U^N$. In an arbitrary state x, the corresponding control action is then obtained by interpolation:

$$u(x) = p^\top \phi_f(x) \tag{2.15}$$

where $\phi_f(x)$ are the same basis functions as defined for $V(x)$. The advantage of this method is its computational simplicity: most computations are done off-line (vector p is actually obtained for free as a byproduct of the fuzzy value iteration algorithm) and the online interpolation is computationally cheap. Another advantage is that (2.15) directly produces continuous control actions. However, the control signal is not necessarily smooth and the interpolation can also result in a steady-state error. Therefore, [9] proposes a symbolic approximation method which is computationally effective and also yields smooth controls. A simplified version of this method is applied here. The policy is approximated analytically. For a typical optimal control problem, the policy surface can be split into saturated parts where the control signal attains the minimal or maximal possible value, and a rather steep transition between the two parts. The transition is generally nonlinear, but often can be well enough approximated by a linear function. The overall policy is then described by:

$$u(x) \quad = \quad \text{sat}\big(\psi \varphi(x)\big) \tag{2.16}$$

with ψ obtained by using linear regression on samples of the steep transition augmented with samples on the boundaries between the transition and the saturated hyper planes, and $\varphi(x)$ being the basis function vector. The function $\text{sat}(\cdot)$ defined as follows:

$$\text{sat}(z) = \max\left(U_{\text{min}}, \min\left(U_{\text{max}}, z\right)\right)$$

The control policy is approximated as a piece-wise linear function with n parameters and n basis functions:

$$u(x) = \text{sat}(\begin{bmatrix} \psi_1 & \psi_2 & \psi_3 & . & . & . & \psi_{n+1} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ . \\ . \\ x_n \\ 1 \end{bmatrix}) = \text{sat}(\psi_1 x_1 + \psi_2 x_2 + \ldots . \psi_n x_n + \psi_{n+1}) \tag{2.17}$$

where ψ_1, ψ_2,.....,ψ_{n+1} are the policy parameters and x_1, x_2,....., x_n are the state variables.

2.6. Actor-only RL methods

A model of the value function can be learned from samples corresponding to single steps in the trajectories of the agent. Such a model is called a critic and the policy is called the actor. As given in the review [13], RL algorithms which search for a policy in the state action space can be divided into into three groups: actor-only, critic-only and actor-critic methods. The majority of actor-only algorithms work with a parameterized family of policies and optimize the cost defined directly over the parameter space of the policy. A major advantage of actor-only methods over critic-only methods is that they allow the policy to generate actions in the complete continuous action space.

2.6.1. Standard policy gradient

A policy gradient method is generally obtained by parameterizing the policy π by the parameter vector $\psi \in \mathbb{R}^p$. Assuming that the parameterization is differentiable with respect to ψ, the standard gradient of the cost function J with respect to ψ is described by

$$\nabla_\psi J = \frac{\partial J}{\partial \pi_\psi} \frac{\partial \pi_\psi}{\partial \psi} \tag{2.18}$$

By using standard optimization techniques, a locally optimal solution of the cost J can be found. The standard gradient $\nabla_\psi J$ is estimated per time step and the parameters are then updated in the direction

of this gradient. For example, a simple gradient ascent method would yield the policy gradient update equation

$$\psi_{k+1} = \psi_k + \alpha_{a,k} \nabla_\psi V_k \tag{2.19}$$

where $\alpha_{a,k}$ is the learning rate of the actor. The main advantage of actor-only methods is their strong convergence property, which is naturally inherited from gradient descent methods. Convergence is obtained if the estimated gradients are unbiased and the learning rates $\alpha_{a,k}$ satisfy

$$\sum_{k=0}^{\infty} \alpha_{a,k} = \infty \quad \text{and} \quad \sum_{k=0}^{\infty} \alpha_{a,k}^2 < \infty \tag{2.20}$$

A drawback of the actor-only approach is that the estimated gradient may have a large variance. Also, every gradient is calculated without using any knowledge of past estimates.

2.6.2. Natural policy gradient

Standard gradient descent is most useful for cost functions that have a single minimum and whose gradients are isotropic in magnitude with respect to any direction away from its minimum [14]. In practice, these two properties are almost never true. The existence of multiple local minima of the cost function is a known problem in RL and is usually overcome by exploration strategies. Also, the performance of methods that use standard gradients relies heavily on the choice of a coordinate system over which the cost function is defined. In robotics, it is common to have a curved state space e.g. because of the presence of angles in the state. A cost function will then usually be defined in that curved space too, possibly causing inefficient policy gradient updates to occur. The natural gradient incorporates knowledge about the curvature of the space into the gradient. It is a metric based not on the choice of coordinates, but on the space that those coordinates parameterize. If a function $J(\psi)$ is parameterized by ψ in Euclidean space, the squared Euclidean norm of an increment $\Delta\psi$ is given by

$$\|\Delta\psi\|_E^2 = \Delta\psi^\top \Delta\psi \tag{2.21}$$

When ψ is transformed to other coordinates $\tilde{\psi}$ in a non-Euclidean space, the squared norm of the increment $\Delta\tilde{\psi}$ with respect to that Riemannian space is given by:

$$\|\Delta\tilde{\psi}\|_R^2 = \Delta\tilde{\psi}^\top G(\tilde{\psi}) \Delta\tilde{\psi} \tag{2.22}$$

where $G(\tilde{\psi})$ is the Riemannian metric tensor, a $n \times n$ positive definite matrix characterizing the intrinsic local curvature of a particular manifold in an n-dimensional space. For Euclidean spaces, $G(\tilde{\psi})$ is the identity matrix. Standard gradient descent for the new parameters $\tilde{\psi}$ would define the steepest descent with respect to the norm $\|\Delta\tilde{\psi}\|^2 = \Delta\tilde{\psi}^\top \Delta\tilde{\psi}$. This would result in a different gradient direction, despite keeping the same cost function and only changing the coordinates. The natural gradient avoids this problem and always points in the "right" direction by taking into account the Riemannian structure of the parameterized space over which the cost function is defined. Now $\tilde{J}(\tilde{\psi} + \Delta\tilde{\psi})$ is minimized while keeping $\Delta\tilde{\psi}$ small. This results in the natural gradient $\tilde{\nabla}_{\tilde{\psi}} \tilde{J}(\tilde{\psi})$ which is a linear transformation of the standard gradient $\nabla_{\tilde{\psi}} \tilde{J}(\tilde{\psi})$ by the inverse of $G(\tilde{\psi})$

$$\tilde{\nabla}_{\tilde{\psi}} \tilde{J}(\tilde{\psi}) = G^{-1}(\tilde{\psi}) \nabla_{\tilde{\psi}} \tilde{J}(\tilde{\psi}) \tag{2.23}$$

For a manifold of distributions, the Riemannian tensor is the Fisher Information Matrix (FIM) [15].

2.6.3. Episode based actor-only methods

A majority of suitable actor-only methods found in literature are based on episodic RL. In episodic RL, the agent executes a task until a terminal state is reached. Executing a policy from an initial state until the terminal state, called a Monte Carlo roll-out, leads to a trajectory τ_r which contains information about the states visited, actions executed,and rewards received i.e. $\tau_r = [x_{1\ldots N+1}, u_{1\ldots N}]$, where N is the number of discrete time steps, $x_{1\ldots N+1} = [x_1, x_2, x_3 \ldots x_{N+1}]$ is the state vector and $u_{1\ldots N} = [u_1, u_2, u_3 \ldots u_N]$ is the control input/action vector. The policy is executed K times with the same parameters ψ. The expected return to be maximized is:

$$J(\psi) = \sum_{\mathbb{K}} p(\tau_r) R(\tau_r) \tag{2.24}$$

where $\mathbb{K}$ is the set of all episodes, $p(\tau_r)$ is the probability of a episode and $R(\tau_r)$ is the return of an episode given by:

$$p(\tau_r) = p(x_1) \prod_{k=1}^{N} p(x_{k+1}|x_k, u_k)\pi(u_k|x_k, k) \qquad \text{and} \qquad R(\tau_r) = \frac{1}{N} \sum_{k+1}^{N} \rho(x_k, u_k, x_{k+1}) \tag{2.25}$$

where $p(x_1)$ is the initial state distribution, $p(x_{k+1}|x_k, u_k)$ is the next state distribution, and $\rho(x_k, u_k, x_{k+1})$ is the immediate reward. To find the lower bound on the return $J(\psi)$, the episodes are weighed with the returns $R(\tau_r)$ and are matched with a new policy parameterized by ψ' [16]. This matching of the success-weighted path distribution is equivalent to minimizing the Kullback- Leibler divergence $D(p_{\psi'}(\tau_r)||p_{\psi}(\tau_r)R(\tau_r))$ between the new path distribution $p_{\psi'}(\tau_r)$ and the reward-weighted previous one $p_{\psi}(\tau_r)R(\tau_r)$. This results in a lower bound on the expected return using Jensen's inequality and the concavity of the logarithm [16], [17]:

$$\begin{aligned} \log J(\psi') = \log \sum_{\mathbb{K}} \frac{p_{\psi}(\tau_r)}{p_{\psi}(\tau_r)} p_{\psi'}(\tau_r) R(\tau_r) \geq \sum_{\mathbb{K}} p_{\psi}(\tau_r) R(\tau_r) \log \frac{p_{\psi'}(\tau_r)}{p_{\psi}(\tau_r)} \\ \propto -D(p_{\psi}(\tau_r)R(\tau_r)||p_{\psi'}(\tau_r)) = L_{\psi}(\psi') \end{aligned} \tag{2.26}$$

where $D(p(\tau_r)||q(\tau_r)) = \sum_{\mathbb{K}} p(\tau_r) \log(p(\tau_r)/q(\tau_r))$ is the Kullback-Leibler divergence. As pointed out in [16], $p_{\psi}(\tau_r)R(\tau_r)$ is an improper probability distribution i.e the immediate costs sum to a constant number and are always positive. The policy improvement step is equivalent to maximizing the lower bound on the expected return, resulting in EM (Expectation Maximization) algorithms.

2.7. Policy Learning by Weighting Exploration with the returns (PoWER)

REINFORCE [18] estimates the standard gradient, which is not robust when noisy, discontinuous utility functions are involved. Also, it requires the manual tuning of the learning rate α_{α}, which is not straightforward, but critical to the performance [19], [17]. The PoWER algorithm [17] addresses these issues by using reward based averaging. Reward-weighted averaging follows the natural gradient, without having to actually compute the gradient or the Fisher Information Matrix. Differentiating the function $L_{\psi}(\psi')$ given in equation 2.26 gives:

$$\nabla_{\psi'} L_{\psi}(\psi') = \sum_{\mathbb{K}} p_{\psi}(\tau_r) R(\tau_r) \nabla_{\psi'} \log p_{\psi'}(\tau_r) \tag{2.27}$$

where $\nabla_{\psi'} \log p_{\psi'}(\tau_r) = \sum_{t=1}^{N} \nabla_{\psi'} \log \pi(u_t|x_t, t)$ is the log derivative of the path distribution. Substituting Equation 2.25 in Equation 2.27 gives:

$$\nabla_{\psi'} L_{\psi}(\psi') = \mathbb{E}\Big(\sum_{k=1}^{N} \nabla_{\psi'} \log \pi(u_k|x_k, k) Q^{\pi}(x, u, k) \Big) \tag{2.28}$$

where $Q^{\pi}(x, u, k)$ is the state action value function.

Methods like REINFORCE and [20] use state independent, Gaussian noise i.e. $\varepsilon_k \sim \mathcal{N}(0, \sum)$. Reward-Weighted Regression is obtained for episodic RL by setting Equation 2.28 to 0 and solving for ψ'. This naturally yields a weighted regression method with the state-action values $Q^{\pi}(x, u, k)$ as weights. [17] takes the stochastic policy $\pi(u|x, k)$ to be $u_k = \psi^{\top}\varphi(x, k) + \varepsilon(\varphi(x, k))$, where the perturbation is approximated as $\varepsilon(\varphi(x, k)) = \varepsilon_k^{\top}\varphi(x, k)$ like in [21]. This gives $u_k = (\psi^{\top} + \varepsilon_k^{\top})\varphi(x, k)$. Thus, PoWER implements a policy perturbation scheme where the parameters of the policy rather than its output are perturbed i.e $\pi_{\psi+\varepsilon_k}(x)$ rather than $\pi_{\psi}(x) + \varepsilon_k$. The policy $u \sim \pi(u_k|x_k, k) = \mathcal{N}(u|\psi^{\top}\varphi(x, u), \hat{\sum}(x, k))$ is substituted in Equation 2.28 and solved after setting it to 0. The update rule obtained is:

Algorithm 2: Policy learning by Weighting Exploration with the returns (PoWER)

Input: initial policy parameters ψ_0

repeat

Sample: Perform episode(s) using $u_k = (\psi^\top + \varepsilon_k^\top)\varphi(x,k)$ with $\varepsilon_k \sim \mathcal{N}(0, \sigma_{ij}^2)$ and collect all $(k, x_k, u_k, x_{k+1}, \varepsilon_k, r_{k+1})$ for $k = 1, 2 \ldots, N+1$

Estimate: Use unbiased estimate $R(x,u,k) = \frac{1}{N}\sum_{k=1}^{N} \rho(x_k, u_k, x_{k+1})$

Reweight: Compute importance weights and reweight episodes, discard low importance roll-outs

Update policy using $\psi_{k+1} = \psi_k + \langle \sum_{k=1}^{N} \varepsilon_k R(x,u,k)\rangle / \langle \sum_{k=1}^{N} R(x,u,k)\rangle$

until $\psi_{i+1} \approx \psi_i$;

$$\begin{aligned}\delta\psi &= \left(\mathbb{E}\left(\sum_{k=1}^{N} M_k Q_{k,k_r}^{\pi}(x,u,k)\right)\right)^{-1}\left(\mathbb{E}\left(\sum_{k=1}^{N} M_k \varepsilon_k^{k_r} Q_{k,k_r}^{\pi}(x,u,k)\right)\right) \\ &\approx \left(\sum_{k_r=1}^{K}\sum_{k=1}^{N} M_k Q_{k,k_r}^{\pi}(x,u,k)\right)^{-1}\left(\sum_{k_r=1}^{K}\sum_{k=1}^{N} M_k \varepsilon_k^{k_r} Q_{k,k_r}^{\pi}(x,u,k)\right)\end{aligned} \tag{2.29}$$

where $M_k = \varphi_k \varphi_k^\top \left(\varphi_k^\top \sum \varphi_k\right)$. The parameter vector ψ is updated as $\psi \leftarrow \psi + \delta\psi$.

Algorithm 3: Adaptation of ABS control policy parameters by PoWER (Policy Learning by Weighting Exploration with Returns) with constant parameter variance

Input: x_0, ρ_{off}, ψ_0, σ^2, Initial surface, Final Surface, ψ^*, $N_{\text{noiseless}}$, N_{iter}, N_{best}

Calculate ideal return for an episode using ideal parameters of the final surface ψ^* i.e. $u_t = \psi^{*T}\varphi(x,t)$ and $R^* = \rho_{\text{off}} - d_x^*$ where d_x^* is the ideal braking distance

Simulate first episode with zero exploration i.e. $u_t = \psi_0^T \varphi(x,t)$ and record $d_{x,0}$

Record $R_{\text{noiseless},0} = \rho_{\text{off}} - d_{x,0}$

for $i = 0 \ldots N_{iter\text{-}1}$ **do**

Calculate return of the i^{th} episode using $R_i = \rho_{\text{off}} - d_{x,i}$

Store and sort the return in an importance sampling table

$p_{\text{num}} \leftarrow 0$

$p_{\text{dnom}} \leftarrow 0$

for $j = 1 \ldots N_{best}$ **do**

$\varepsilon \leftarrow \psi_{\text{best}} - \psi_i$

$p_{\text{num}} \leftarrow p_{\text{num}} + \varepsilon R_i$

$p_{\text{dnom}} \leftarrow p_{\text{dnom}} + R_i$

end

$\psi_{i+1} \leftarrow \psi_i + p_{\text{num}}/p_{\text{dnom}}$

if $i > 0$ & $i \bmod N_{noiseless} = 0$ **then**

Simulate noiseless episode using ψ_{i+1} and record $R_{\text{noiseless},i/N_{\text{noiseless}}+1} = \rho_{\text{off}} - d_{x,i+1}$

end

if $i < N_{iter} - 1$ **then**

$\psi_{i+1} \leftarrow \psi_{i+1} + \sqrt{\sigma^2}\varepsilon_{i+1}$

end

Simulate $(i+1)^{\text{th}}$ episode and record $d_{x,i+1}$

end

Record $R_{N_{\text{iter}}} = \rho_{\text{off}} - d_{x,N_{\text{iter}}}$

Output: $\psi = \psi_{N_{\text{iter}}}, R = R_{N_{\text{iter}}}$

For the update, the return of an episode $R(\tau_r)$ can also be used instead of the state-action value function $Q(x,u,k)$. In order to reduce the number of episodes in this on-policy scenario, importance sampling can be used as described in the context of reinforcement learning in [22]. Samples with very small importance weights are discarded. In [17], the state x and time k are stored in the trajectory and is used to calculate $\varphi(x,k)$. As per [19], since the parameter vector ψ does not depend on the state and the sum in Equation 2.29 is not over the state, it is not necessary to store the state in the trajectory.

Algorithm 4: Adaptation of ABS control policy parameters by PoWER (Policy Learning by Weighting Exploration with Returns) with adaptive parameter variance

Input: x_0, ρ_{off}, ψ_0, σ_0^2, Initial surface, Final Surface, ψ^*, $N_{\text{noiseless}}$, N_{iter}, N_{best}

Calculate ideal return for an episode using ideal parameters of the final surface ψ^* i.e. $u_t = \psi^{*T}\varphi(x,t)$ and $R^* = \rho_{\text{off}} - d_x^*$ where d_x^* is the ideal braking distance
Simulate first episode with zero exploration i.e. $u_t = \psi_0^T\varphi(x,t)$ and record $d_{x,0}$
Record $R_{\text{noiseless},0} = \rho_{\text{off}} - d_{x,0}$
for $i = 0 \ldots N_{iter} - 1$ **do**
  Calculate return of the i^{th} episode using $R_i = \rho_{\text{off}} - d_{x,i}$
  Store and sort the return in an importance sampling table
  $p_{\text{num}} \leftarrow 0$
  $p_{\text{dnom}} \leftarrow 0$
  for $j = 1 \ldots N_{best}$ **do**
    $\varepsilon \leftarrow \psi_{\text{best}} - \psi_i$
    $p_{\text{num}} \leftarrow p_{\text{num}} + \varepsilon R_i$
    $p_{\text{dnom}} \leftarrow p_{\text{dnom}} + R_i$
  end
  $\psi_{i+1} \leftarrow \psi_i + p_{\text{num}}/p_{\text{dnom}}$
  $v_{\text{num}} \leftarrow 0$
  $v_{\text{dnom}} \leftarrow 0$
  for $j = 1 \ldots 2 \times N_{best}$ **do**
    $\varepsilon \leftarrow \psi_{\text{best}} - \psi_i$
    $v_{\text{num}} \leftarrow v_{\text{num}} + \varepsilon^2 R_i$
    $v_{\text{dnom}} \leftarrow v_{\text{dnom}} + R_i$
  end
  $\sigma_{i+1}^2 \leftarrow \sigma_i^2 + v_{\text{num}}/v_{\text{dnom}}$
  if $i > 0$ & $i \bmod N_{noiseless} = 0$ **then**
    Simulate noiseless episode using ψ_{i+1} and record $R_{\text{noiseless},i/N_{\text{noiseless}}+1} = \rho_{\text{off}} - d_{x,i+1}$
  end
  if $i < N_{iter} - 1$ **then**
    $\psi_{i+1} \leftarrow \psi_{i+1} + \sqrt{\sigma^2}\varepsilon_{i+1}$
  end
  Simulate $(i+1)^{\text{th}}$ episode and record $d_{x,i+1}$
end
Record $R_{N_{\text{iter}}} = \rho_{\text{off}} - d_{x,N_{\text{iter}}}$

Output: $\psi = \psi_{N_{\text{iter}}}, R = R_{N_{\text{iter}}}$

2.8. Summary

In this chapter, the definitions and concepts of Reinforcement Learning used to derive the control policy i.e. Value Iteration using fuzzy sets, methods for initial robustness of the control policy, and methods used for adaptation of the policy parameters i.e. episode based actor-only RL methods were introduced. These form the basis of the results shown in Chapters 4 and 5.

3

ABS model and baseline controller

This chapter first describes the model and parameters of the ABS system i.e. the single corner model and the Pacejka tire model, followed by the the current principles being used for its control. Finally, the performance of a simple proportional controller for ABS is shown.

3.1. Single corner wheel dynamics

The quarter car model has been used for modelling the wheel dynamics of the car. The wheel slip κ is defined by the formula

$$\kappa = \frac{v_x - \omega r_t}{\max\{v_x, \omega r_t\}} \tag{3.1}$$

where ω denotes the angular velocity of the wheel, v_x is the linear velocity of the wheel/chassis and r_t is the effective rolling radius of the wheel. For braking, $v_x - \omega r_t > 0$, hence the wheel slip is given by:

$$\kappa = \frac{v_x - \omega r_t}{v_x} = 1 - \frac{\omega r_t}{v_x} \tag{3.2}$$

The normalized wheel deceleration η_w is given by

$$\eta_w = -\frac{\dot{\omega} r_t}{g} \tag{3.3}$$

where $\dot{\omega}$ is the wheel angular acceleration and g is the acceleration due to gravity.

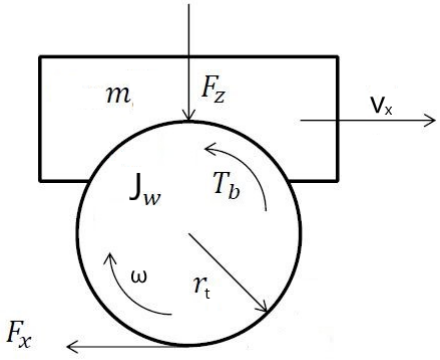


Figure 3.1: The single corner model

The wheel dynamics on a sloped road are described by the following equations:

$$\begin{aligned} J_w\dot{\omega} &= r_t F_x - T_b \\ m\dot{v}_x &= F_g - ma_x = F_g - F_x \end{aligned} \tag{3.4}$$

where T_b is the braking torque on the wheel, F_x is the longitudinal tire-road contact force, J_w and m are the moment of inertia of the wheel and the single-corner mass respectively, F_g is the gravitational force due to the slope of the road, and a_x is the longitudinal acceleration. The parameters of the system are given in Table 3.1.

Table 3.1: ABS system parameters

Parameter	Symbol	Value	Units
Wheel Inertia	J_w	1.2	kgm^2
Tire radius	r_t	0.305	m
Corner car mass	m	450	kg
Gravity acceleration	g	9.81	m/s^2

3.2. Pacejka Tire Model

The tire dynamics can be best modelled by the Pacejka model [23]. It is named the 'magic formula' tire model because there is no particular physical basis for the structure of the equations chosen, but they fit a wide variety of tire constructions and operating conditions.

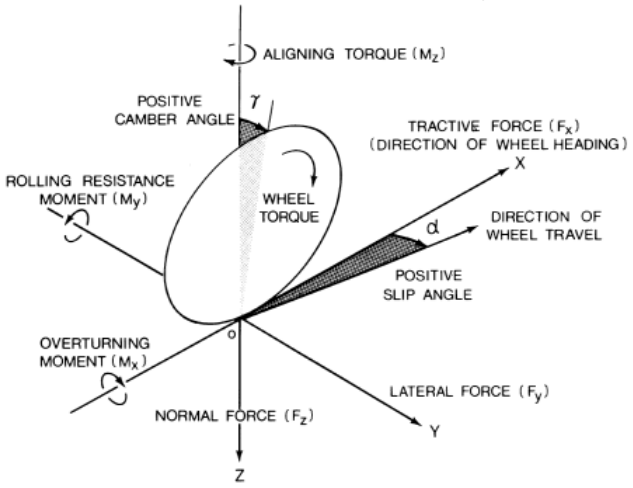


Figure 3.2: The forces and moments on a tire

Each tire is characterized by 10 to 20 coefficients for each important force that it can produce at the contact patch, typically lateral force F_y, longitudinal force F_x, and self-aligning torque M_z, as a best fit between experimental data and the model. These coefficients are then used to generate equations showing how much force is generated for a given vertical load F_z on the tire, camber angle γ_c, lateral slip angle α_s and longitudinal slip κ. The longitudinal force F_x is written as:

$$F_x = F_z \mu_x(\alpha_s, \gamma_c, \kappa) \tag{3.5}$$

where μ_x is the coefficient of friction. For braking in a straight line, the camber angle γ_c and the lateral slip angle α_s are approximately 0. This leads to the magic formula for longitudinal force as a function of vertical load F_z and longitudinal slip κ:

$$F_x(\kappa) = F_z \cdot D \cdot \sin\Big(C \cdot \tan^{-1}[B(1-E)\kappa + E \cdot \tan^{-1}(B\kappa)]\Big) \tag{3.6}$$

where B, C, D and E represent curve fitting constants. The values of these constants are found out by fitting them with available values of F_x from test data in the above equation. The values of the coefficients given in Table 3.2 are taken from [24]. The following assumptions have been made for the quarter car model and tire model:

1. For simplicity, the road is assumed to be flat i.e $F_g = 0$.

2. The four wheels are treated as dynamically decoupled, i.e. the dynamic load transfer phenomena due to pitch motion are neglected.
3. The suspension dynamics are neglected.
4. Since pitch dynamics are neglected, the wheel radius is assumed to be constant.
5. The response of the tire to change in longitudinal slip is assumed to be instantaneous i.e. transient tire behavior is neglected.
6. Delay due to actuator dynamics has not been taken into account.
7. All 4 wheels are assumed to be in contact with the same surface at any instant of time i.e. cases where one or two wheels are on one surface and the other three or two wheels respectively are on another surface have not been considered.

Table 3.2: Magic Formula Coefficients

Surface	B	C	D	E
Dry asphalt	10	1.8	1	0.97
Wet asphalt	12	2.4	0.82	1

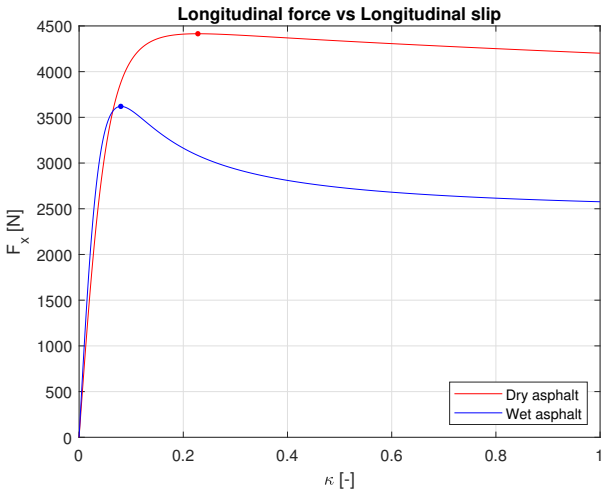


Figure 3.3: Longitudinal braking force vs longitudinal slip for dry and wet asphalt

3.3. Current control methods for ABS

Traditionally, deceleration control has been used for ABS because the wheel deceleration can be easily measured using a wheel encoder [25], [26]. However, if the road surface rapidly changes, on-line estimation of friction characteristics are required [27]. Differentiating Equation 3.2 with respect to time:

$$\dot{\kappa} = -\frac{\dot{\omega} r_t}{v_x} + \frac{\omega r_t \dot{v}_x}{v_x^2} \tag{3.7}$$

In the equilibrium condition, the rate of change of κ will be 0. Using this, the second equation of Equation 3.4 and Equation 3.3:

$$\begin{aligned} \dot{\omega} &= \frac{\omega \dot{v}_x}{v_x} \\ \implies \dot{\omega} &= -\frac{(1-\kappa)F_x}{m r_t} \\ \implies \eta_w &= \frac{F_x(1-\kappa)}{mg} \end{aligned} \tag{3.8}$$

The graph of η_w vs κ for dry and wet asphalt is shown in Figure 3.4. The line for reference value of κ of dry asphalt intersects the curves at unique equilibrium points. This means that a single κ setpoint ensures a stable equilibrium for both dry and wet asphalt. For this reason, the control of wheel slip is easier as one reference slip value provides robust performance, which will thus be sub-optimal. The reference value of κ is found out from Figure 3.3. The slip where the magnitude of the longitudinal force is maximum is the reference slip value $\overline{\kappa}$.

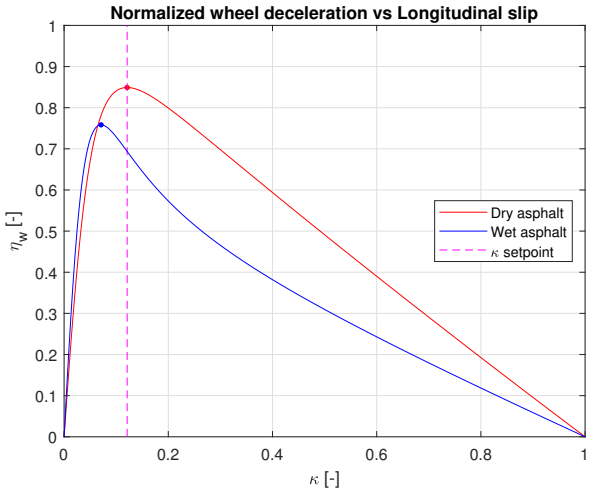


Figure 3.4: η_w vs κ with unique equilibrium points

For deceleration control, adaptation of reference value is required as can be seen from Figure 3.5. The line for reference value of η_w intersects the curve of wet asphalt at only one point but the curve of dry asphalt at two points, resulting in two equilibrium points. As the equilibrium is not unique, adaptation of reference value is required to ensure stable equilibrium.

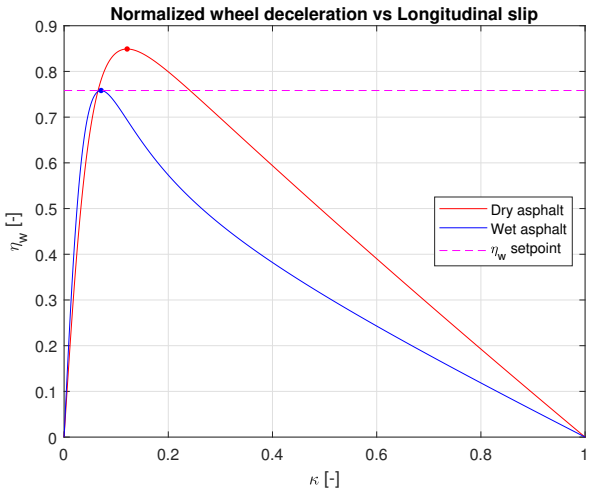


Figure 3.5: η_w vs κ with non-unique equilibrium points

In addition to sub-optimal performance on other surfaces, another drawback of wheel slip control is that wheel slip is difficult to measure because accurate estimation of the linear velocity of the car is difficult, especially at low speeds. A mixed slip controller which controls both the wheel slip and deceleration uses a convex combination of κ and η_w to define a parameter ϵ_w:

$$\epsilon_w = \alpha_w \kappa + (1 - \alpha_w)\eta_w \tag{3.9}$$

where α_w is a constant. For $\alpha_w = 1$, a pure slip controller is obtained and for $\alpha_w = 0$, a pure deceleration controller is obtained. Since the goal is to obtain optimal performance on all surfaces, the setpoint of κ should not be fixed. Hence, if a mixed slip controller is being used, the reference values of the two

variables κ and η_w, and the value of the parameter α_w need to be adapted online according to the surface. After the reference values $\overline{\kappa}$ and $\overline{\eta}_w$ have been found, the reference value of ϵ_w is found out using

$$\overline{\epsilon}_w = \alpha_w \overline{\kappa} + (1 - \alpha_w)\overline{\eta}_w \tag{3.10}$$

The goal is to minimize the error between ϵ_w and $\overline{\epsilon}_w$. The wheel angular acceleration $\dot{\omega}$ and angular speed ω, and the wheel/chassis linear acceleration a_x and linear velocity v_x are measured after each time step. These values are used to calculate κ, η_w and ϵ_w. The controller applies corrective brake pressure according to the the error with respect to the reference value $\overline{\epsilon}_w$.

3.3.1. Performance Criteria

The performance of the ABS controller can be determined on the basis of passenger safety and comfort. For safety purposes, the braking distance of the car should be minimized and thus is used as the primary metric. For the purpose of comfort, variation i.e. standard deviation of the vehicle deceleration is used as the secondary metric. The vehicle deceleration is measured through an accelerometer and thus can be used to calculate the standard deviation.

3.4. Proportional-Integral controller for pure slip control

A P-I controller has to work on the basis of the maximum braking torque possible (not the controller limit) such that the longitudinal braking force is maximized. Using Equation 3.7, Equation 3.4 and Equation 3.2:

$$\begin{aligned} \dot{\kappa} &= (1-\kappa)\left[\frac{T_b - r_t F_x}{J_w \omega} - \frac{F_x}{m v_x}\right] \\ \implies \dot{\kappa} &= \frac{1-\kappa}{J_w \omega}\left[T_b - r_t F_x - \frac{F_x J_w (1-\kappa)}{m r_t}\right] \\ \implies \dot{\kappa} &= \frac{1-\kappa}{J_w \omega}\left[T_b - \zeta(\kappa)\right] \qquad \text{where} \qquad \zeta(\kappa) = F_x\left[r_t + \frac{J_w(1-\kappa)}{m r_t}\right] \end{aligned} \tag{3.11}$$

The graph of $\zeta(\kappa)$ vs κ is shown in Figure 3.6. The points corresponding to the peaks of the curves are the ideal values of the braking torque T_b for the respective surfaces.

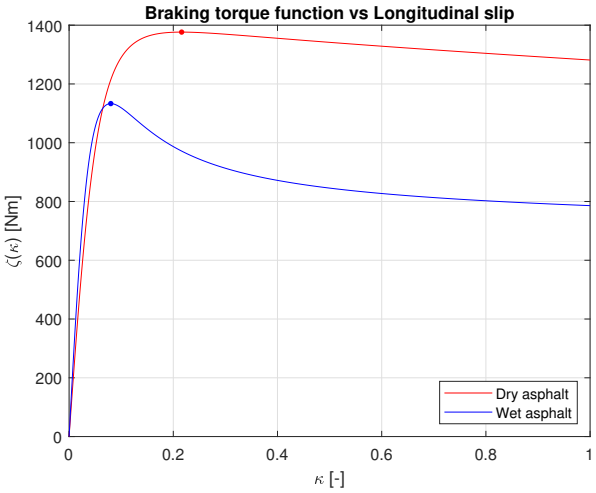
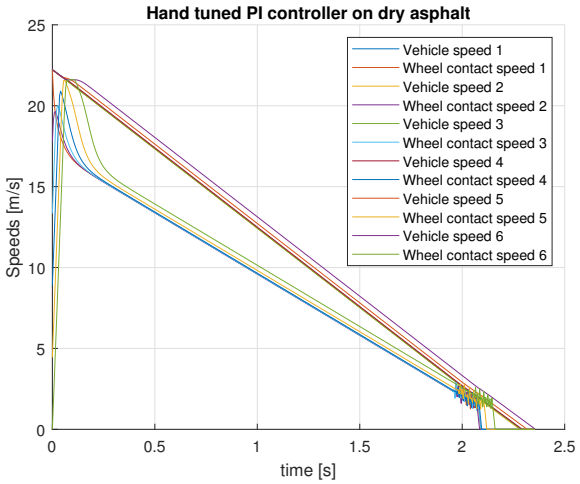


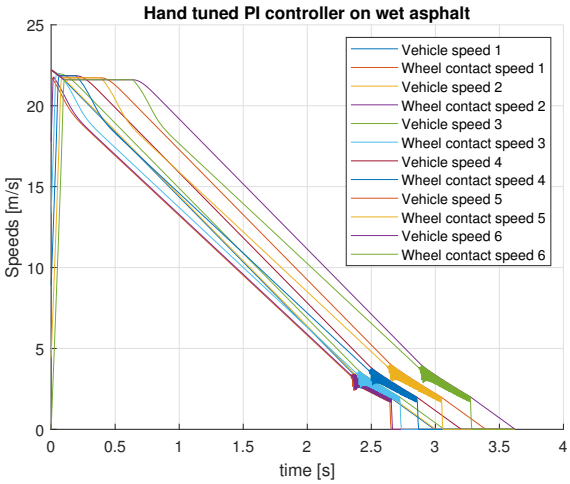
Figure 3.6: Graph of variation of $\zeta(\kappa)$ with κ

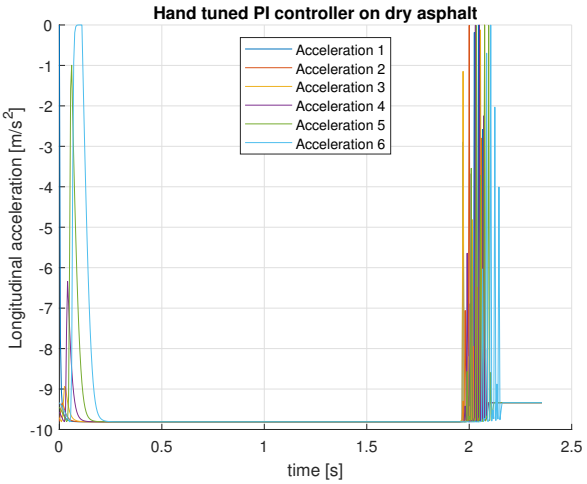
A discrete time proportional-integral controller was implemented for pure wheel slip control for different initial conditions. The initial speed was kept the same (80 km/h) but different initial wheel slips ([0:0.2:1]) were implemented. The slips are referred to as Slip 1, Slip 2, Slip 3, Slip 4, Slip 5, Slip 6 respectively. The control law for the wheel slip at the k^{th} time step is:

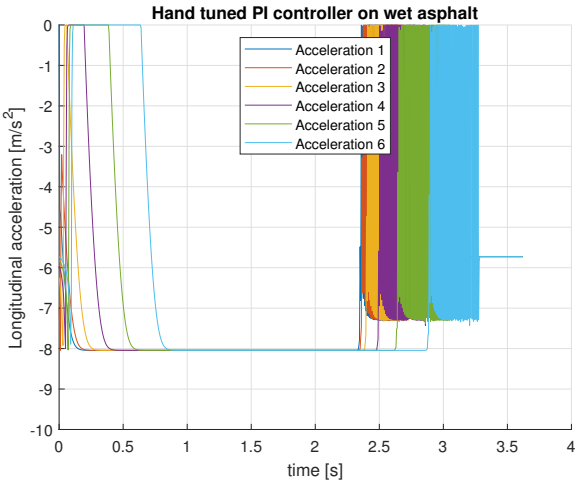
$$T_{b,k} = K_p(\overline{\kappa} - \kappa_k) + K_i \sum_k (\overline{\kappa} - \kappa_k) + T_{b,max} \tag{3.12}$$

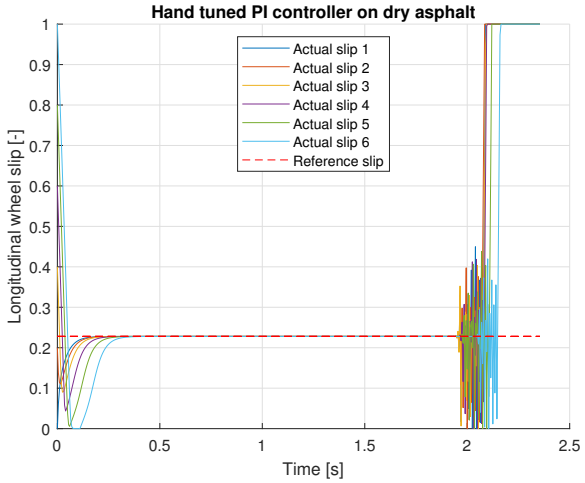
where $T_{b,max}$ is the ideal braking torque found from the peaks of Figure 3.6. The reference value of κ was chosen according to the surface using Figure 3.3. The values of gains taken are $K_p = 10000$ and $K_p = 200000$. The performance on dry and wet asphalt is compared in Figure 3.7.

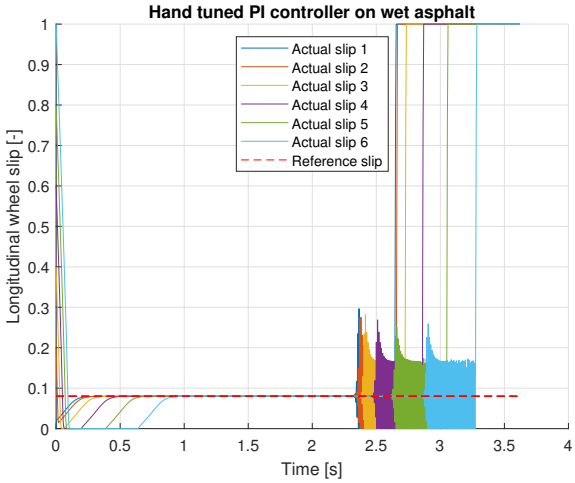












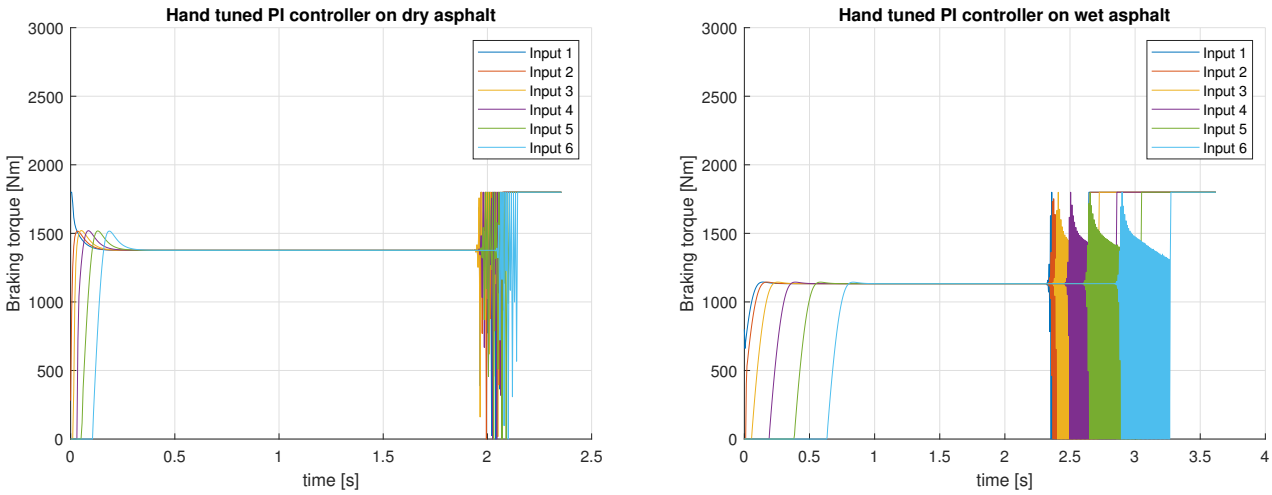


Figure 3.7: Performance comparison on dry and wet asphalt

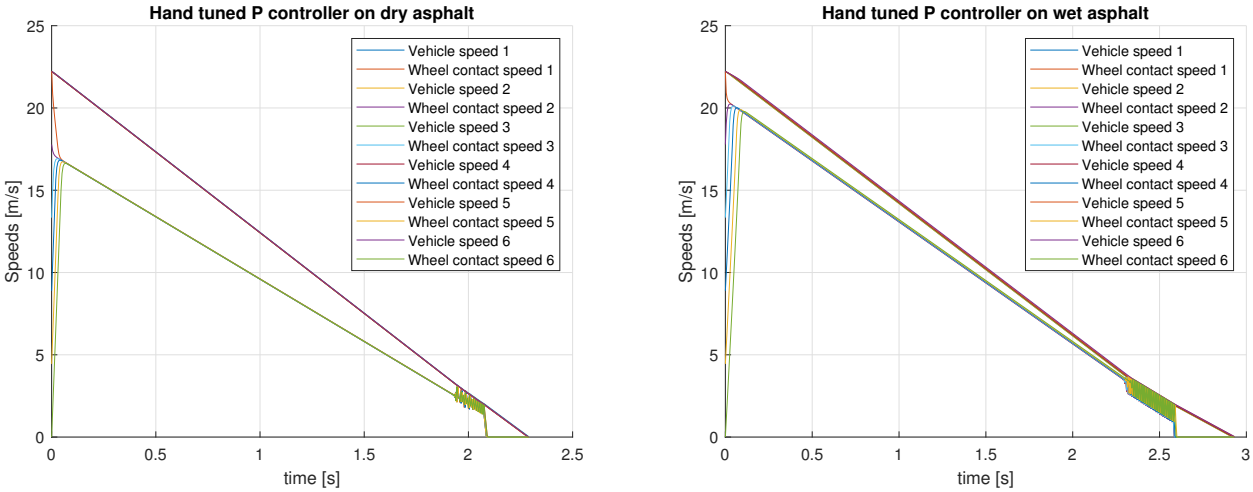
If integral gain is removed to simplify the analysis:

$$T_{b,k} = K_p(\overline{\kappa} - \kappa_k) + T_{b,max} \tag{3.13}$$

The control input without integral gain will offset the setpoint $\overline{\kappa}$ from its ideal value since $T_{b,max}$ is constant for a given surface. Using Equation 3.2:

$$T_{b,k} = -K_p r_t(\frac{\overline{\omega}}{v_x} - \frac{\omega_k}{v_{x,k}}) + T_{b,max} \tag{3.14}$$

Since the state vector at any time step k is $x_k = [v_{x,k}\ \omega_k]^\top$, this control method is equivalent to state feedback control. The value of K_p is taken to be 10000, and the setpoint values of slips for dry and wet asphalt are found to be 2.4 $\overline{\kappa}_{dry}$ and 1.6 $\overline{\kappa}_{wet}$ respectively. The performance on dry and wet asphalt is compared in Figure 3.8.



Among the two controllers, the simple proportional controller is found to perform better. Speeds for lower initial slips converge to zero faster. This difference is marginal on dry asphalt but not small on wet asphalt. Both controllers suffer from control input chattering at speeds just before ABS is switched off. For the P controller, the reason for chattering can be seen from the policy surface. The transition is too steep at the speed of 3 m/s.

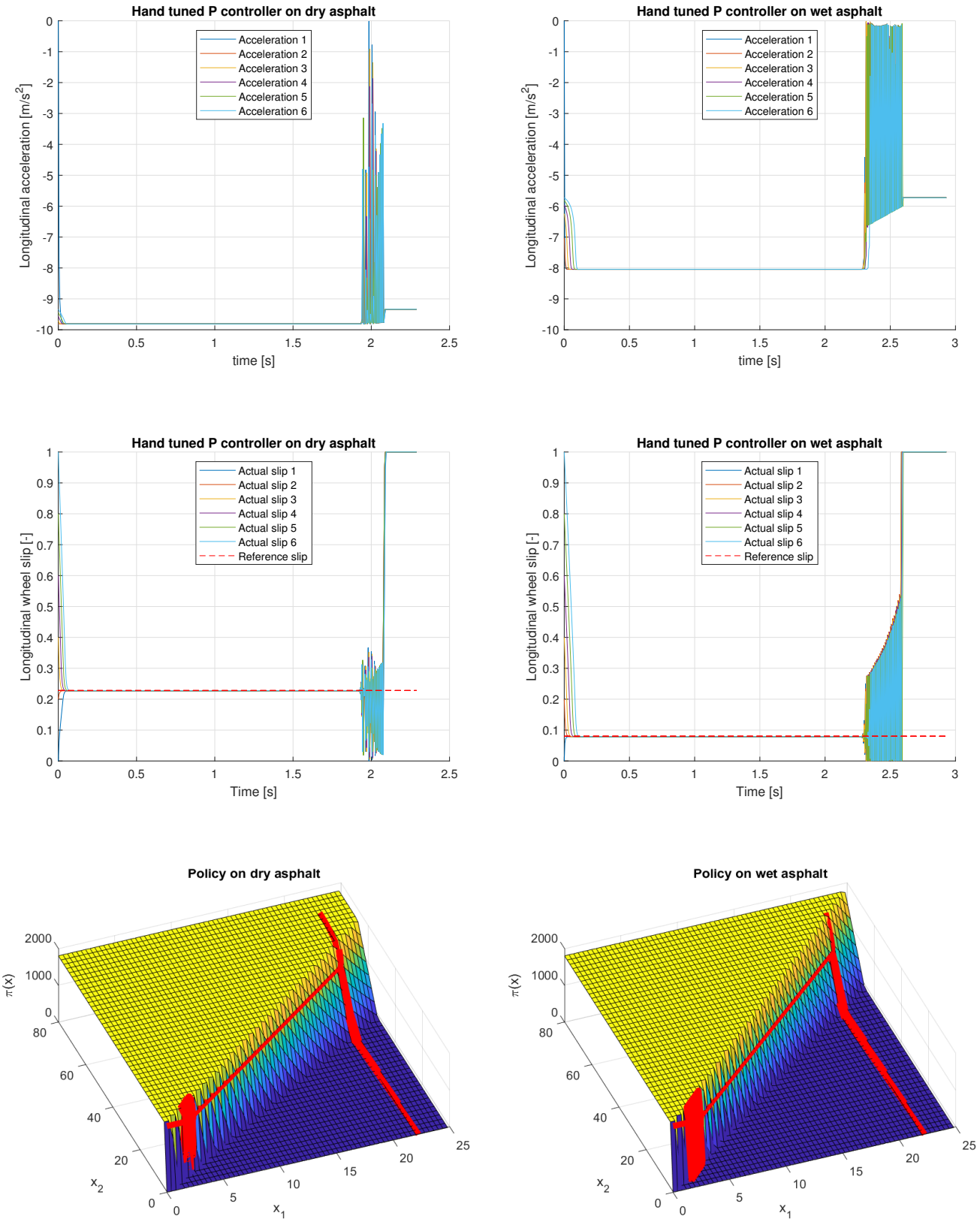


Figure 3.8: Performance comparison on dry and wet asphalt

For the P controller, the braking distances on dry and wet asphalt are given in Table 3.3. When compared with the ideal braking distances from Chapter 4 i.e. 25.31 m for dry on dry and 31.04 m for wet on wet, it can be seen that on dry asphalt, all braking distances except 1 are marginally higher, and on wet asphalt, all braking distances except 1 are higher with the maximum increment being 0.43 m.

Table 3.3: Braking distance [m]

Surface	Slip 1	Slip 2	Slip 3	Slip 4	Slip 5	Slip 6
Dry asphalt	25.36	25.29	25.31	25.32	25.32	25.33
Wet asphalt	31.05	31.03	31.11	31.21	31.35	31.47

For the P controller, the standard deviation in deceleration [m/s^2] on dry and wet asphalt are given in Table 3.4. When compared to ideal values from Chapter 4 i.e. 0.4481 for average on dry and 0.4023 for dry on wet, the comfort levels are much worse for all initial conditions on both dry and wet asphalt.

Table 3.4: Standard deviation in acceleration [m/s^2]

Surface	Slip 1	Slip 2	Slip 3	Slip 4	Slip 5	Slip 6
Dry asphalt	0.8476	0.7964	1.1647	1.1808	1.1035	1.0796
Wet asphalt	1.788	1.7604	1.7602	1.7363	1.7801	1.7051

3.4.1. Proportional control with gain scheduling

The control input for proportional gain is given by Equation 3.14:

$$T_{b,k} = -K_p r_t \left(\frac{\overline{\omega}}{v_x} - \frac{\omega_k}{v_{x,k}}\right) + T_{b,max} \tag{3.15}$$

This equation can be rearranged to give:

$$\begin{aligned} T_{b,k} &= c_1 + c_2 \frac{\omega_k}{v_{x,k}} + T_{b,max} \\ &= \frac{1}{v_{x,k}} (c_3 v_{x,k} + c_2 \omega_k) \end{aligned} \tag{3.16}$$

where c_1, c_2 and c_3 are constants. Thus, the control input is inversely proportional to the linear velocity. This explains the observation that the transition in the policy surfaces in Figure 3.8 becomes more steep as lower velocities are approached. This steep transition leads to control input chattering. To avoid chattering, gain scheduling can be used to cancel the part $\frac{1}{v_{x,k}}$ and the final control input can be written as:

$$T_{b,k} = c_3 v_{x,k} + c_2 \omega_k + c_4 = \begin{bmatrix} c_3 & c_2 & c_4 \end{bmatrix} \begin{bmatrix} v_{x,k} \\ \omega_k \\ 1 \end{bmatrix} \tag{3.17}$$

3.5. Summary

In this chapter, model and parameters of the ABS system were presented along with the mixed slip control method being used currently. The advantages and disadvantages of pure wheel slip control and pure deceleration control were discussed. Results for a hand tuned PI controller were compared with that of a hand tuned proportional controller for pure wheel slip control. Derivation of the control policy of a proportional controller with gain scheduling was also shown.

4

ABS control through offline Reinforcement Learning

As discussed in Chapter 2, there are two policy approximation methods using online interpolation between optimal actions computed offline:

Non-linear interpolation The control action is given by

$$u(x) = p^{\top}\phi(x) \tag{4.1}$$

where $\phi(x)$ are the same basis functions as defined for $V(x)$ and $p = [p_1, \ldots, p_N]^{\top} \in U^N$ are the optimal control actions.

Piecewise linear approximation The policy is approximated as a piece-wise linear function with three parameters and three basis functions:

$$u(x) = \text{sat}(\psi\varphi(x)) = \text{sat}(\begin{bmatrix}\psi_1 & \psi_2 & \psi_3\end{bmatrix}\begin{bmatrix}x_1\\x_2\\1\end{bmatrix}) = \text{sat}(\psi_1 x_1 + \psi_1 x_2 + \psi_3) \tag{4.2}$$

where ψ_1, ψ_2, ψ_3 are the policy parameters and x_1, x_2 are the state variables i.e. chassis linear velocity v_x and wheel angular velocity ω respectively. This policy is similar to Equation 3.17 i.e. the control input for proportional control after gain scheduling.

Table 4.1: Parameters of each policy

Parameter	Dry asphalt policy	Wet asphalt policy	Average policy	Max-Min policy
ψ_1	-556.5	-577.7	-568.3	-577.2
ψ_2	218.9	192.9	196.9	192.7
ψ_3	1347.7	1017.4	1192.3	1016

The parameters of wet asphalt policy and max-min policy are almost equal. This is because wet asphalt is the worst case among dry asphalt and wet asphalt i.e. the RHS of the Bellman equation will be lower for wet asphalt as the peak longitudinal force available on wet asphalt is lower than that on dry asphalt. Hence, the performance of max-min policy is not shown.

4.1. Fuzzy-V iteration and system parameters

The parameters of the fuzzy value iteration algorithm are listed in Table 4.2. The number of membership functions for each state variable is chosen large (41) in order to get a dense coverage of the state space domain of interest. The controller is trained for a maximum possible initial speed of 25 m/s. The maximum input braking torque is taken as 1800 Nm. The discount factor $\gamma = 0.999$ is selected close to one, so that virtually too much discounting takes place towards the end of a typical closed-loop transient

lasting about $3/0.005 = 600$ samples ($\gamma^{600} \approx 0.55$).

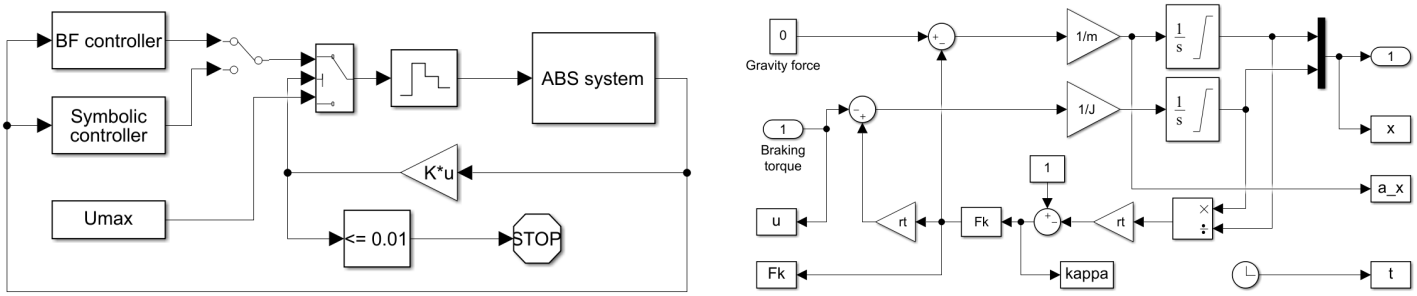
Table 4.2: Fuzzy Value iteration parameters

Parameter	Symbol	Value	Units
State domain	$\mathcal{X}$	$[25] \times [25/r_t]$	m/s × rad/s
Number of BF	N_f	$1681 = 41 \times 41$	–
Discount factor	γ	0.999	–
Convergence threshold	ϵ_f	0.001	–
Sampling period	T_s	0.005	s

The discrete-time model (2.1) is obtained by numerically integrating the continuous-time dynamics (3.4) using the fourth-order Runge-Kutta method with the sampling period of $T_s = 0.005$ s. The state vector consists of the car linear velocity and the wheel angular velocity i.e. $x_k = [v_{x,k}\ \omega_k]^\top$, and the reward function is defined as:

$$r_{k+1} = \rho(x_k, u_k, x_{k+1}) = -\left|x_r - x_k\right|^\top Q \tag{4.3}$$

where $x_r = [0\ 0]^\top$ is the reference state, and $Q = \mathrm{diag}(T_s, 0)$ is a weighting matrix, specifying that instantaneous reward is the negative of the distance travelled in one sampling period and thus the total braking distance must be minimized. The initial speed of the car for calculation of the parameters of the control policy is taken as 80 km/h. The threshold speed for switching off ABS control is chosen as 2 m/s. The Simulink model used for simulation is given in Figure 4.1.



(a) The controller structure

(b) The ABS system

Figure 4.1: The Simulink model

4.2. Results for non-linear interpolation

Results are shown for only one initial speed i.e. 80 km/h for comparison with the results of piecewise linear interpolation, since it is the policy that will be used for adaptation.

4.2.1. Initial speed = 80 km/h

The braking distances on dry and wet asphalt are given in Table 4.3. As expected, the average policy performs better on dry and wet asphalt than wet and dry asphalt policies respectively. On dry asphalt, the average policy is 5.31% worse while the wet policy is 14.96% worse than the ideal. On wet asphalt, the average policy is 6.56% worse while the dry policy is 19.94% worse than the ideal.

Table 4.3: Braking distance [m]

Surface	Dry asphalt policy	Wet asphalt policy	Average policy
Dry asphalt	25.4	29.2	26.36
Wet asphalt	37.29	31.1	33.14

The standard deviation in deceleration [m/s^2] for each policy on dry and wet asphalt are given in Table 4.4. In terms of comfort of the passenger, dry on dry performs better than wet on dry but average

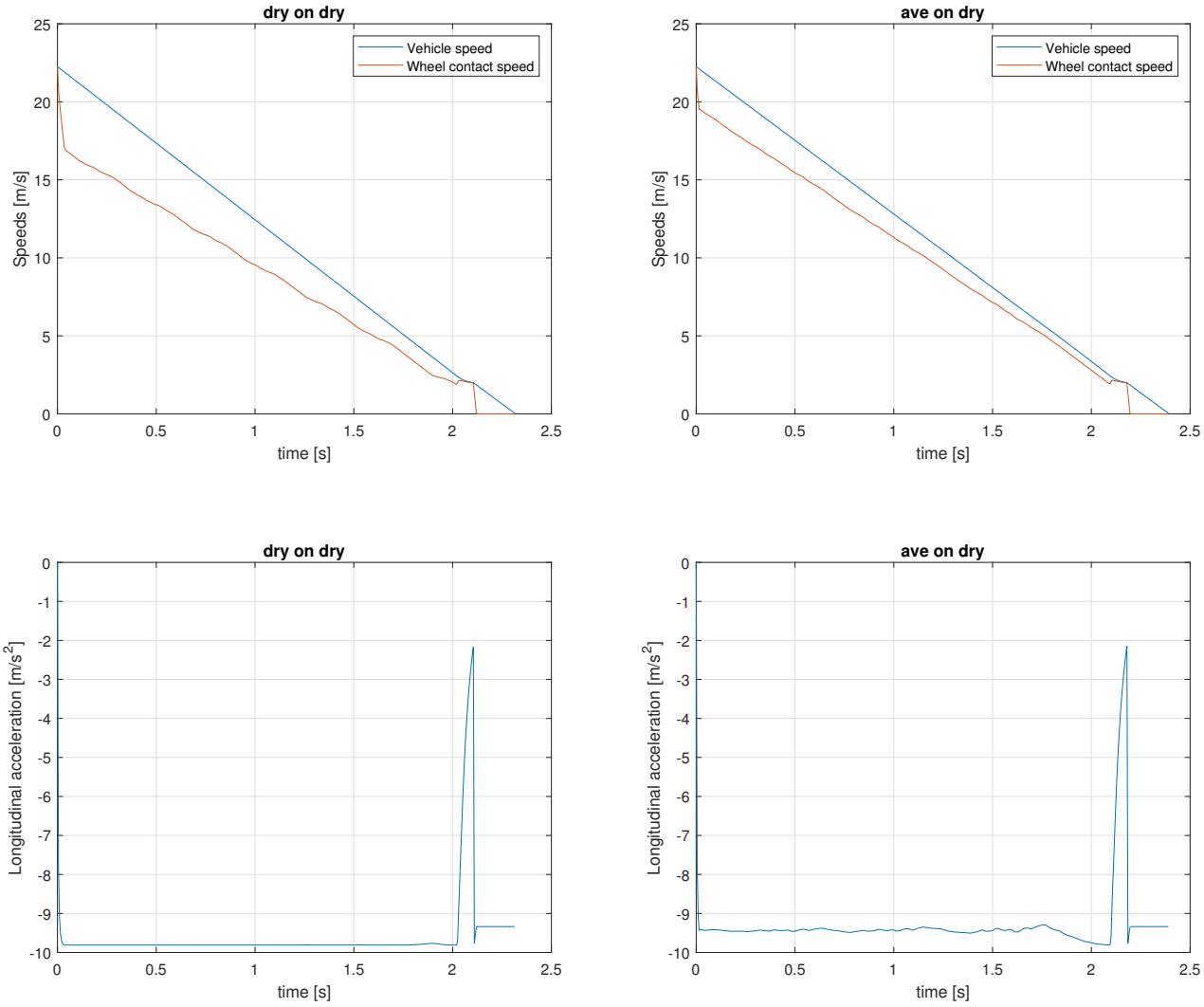
on dry performs better than both dry on dry and wet on dry. Average on wet does not give the best performance but since it is better than wet on wet, it is a better choice as it performs well on both dry and wet asphalt.

Table 4.4: Standard deviation in deceleration [m/s^2]

Surface	Dry asphalt policy	Wet asphalt policy	Average policy
Dry asphalt	1.0877	1.3573	1.0281
Wet asphalt	0.5733	1.2882	0.8539

Performance on dry asphalt

As seen from the Figure 4.2, the performance of the average policy is good when compared to the ideal case i.e. dry on dry. The time taken to stop is marginally higher, the deceleration is slightly lower with minor variations, the wheel slip is mostly constant with minor variations, and the control policy is more smooth i.e. less chattering. The most noticeable lacking part is that the attained wheel slip is almost half of the ideal value.



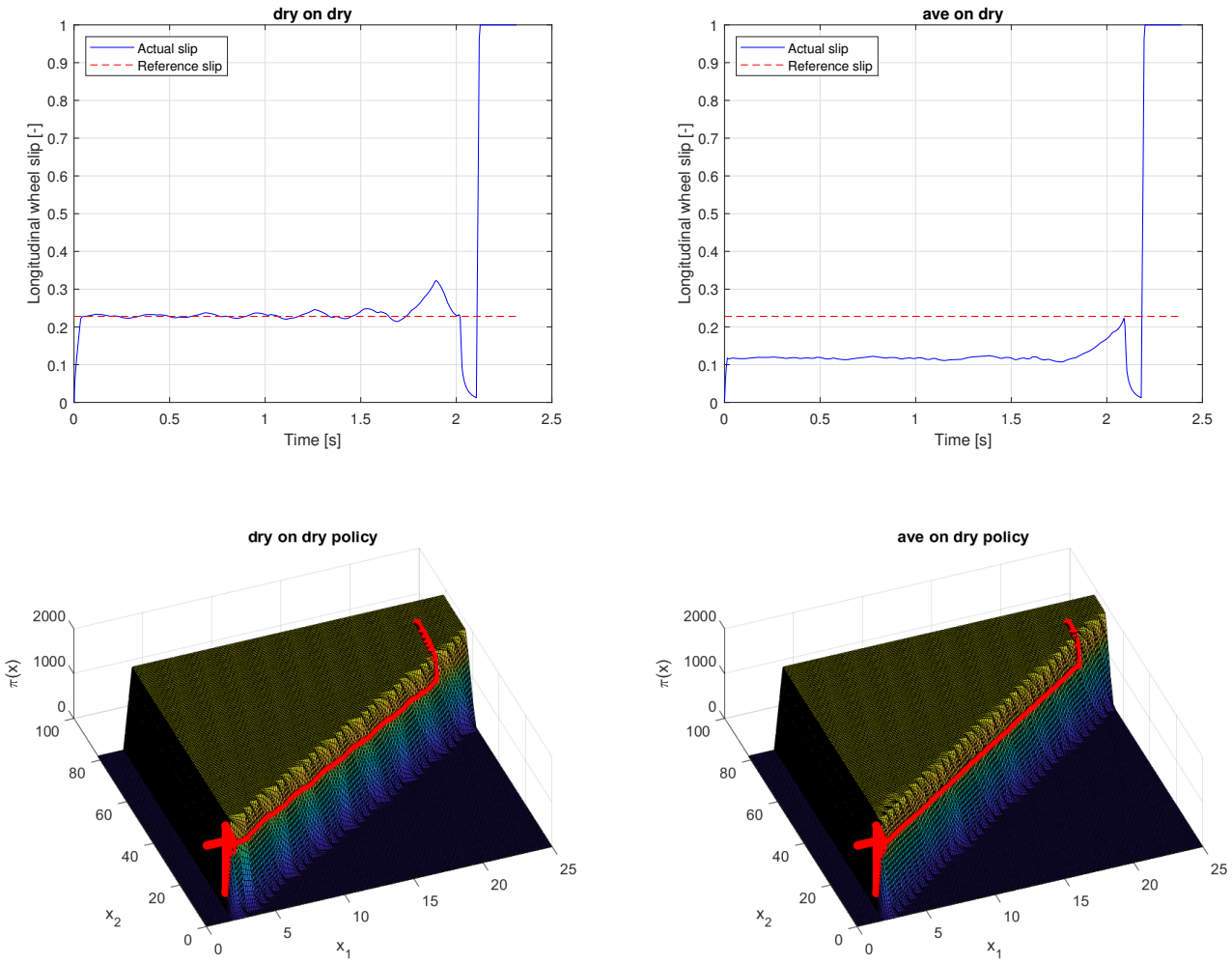
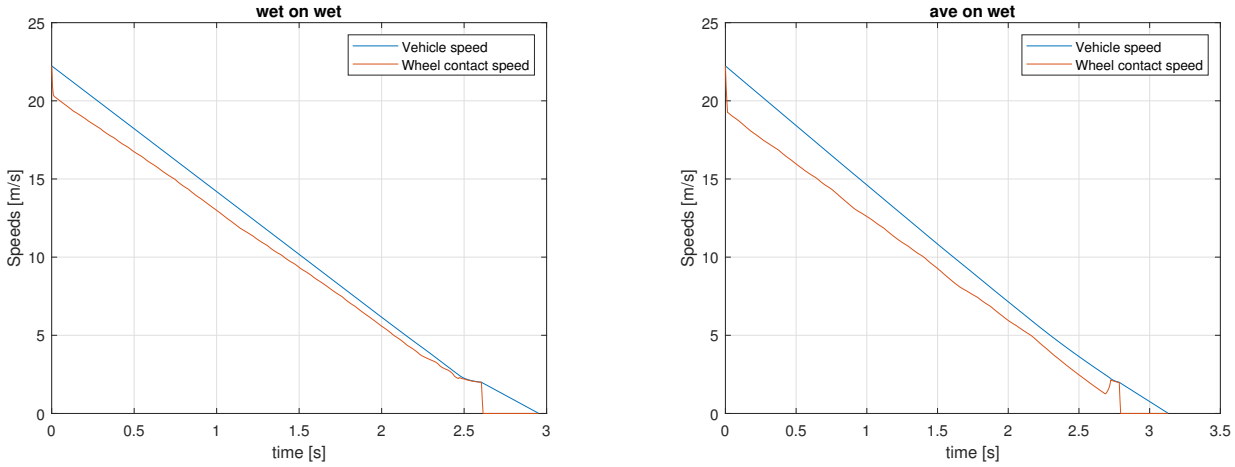


Figure 4.2: Performance comparison on dry asphalt

Performance on wet asphalt

As seen from the Figure 4.3, the performance of the average policy is worse as compared to the ideal level i.e. wet to wet. The time taken to stop is marginally higher, but the variation in contact velocity is increased, the deceleration is slightly lower with more variation, the wheel slip is varying more and is not close to the ideal value, and the control policy is less smooth i.e. increased chattering.



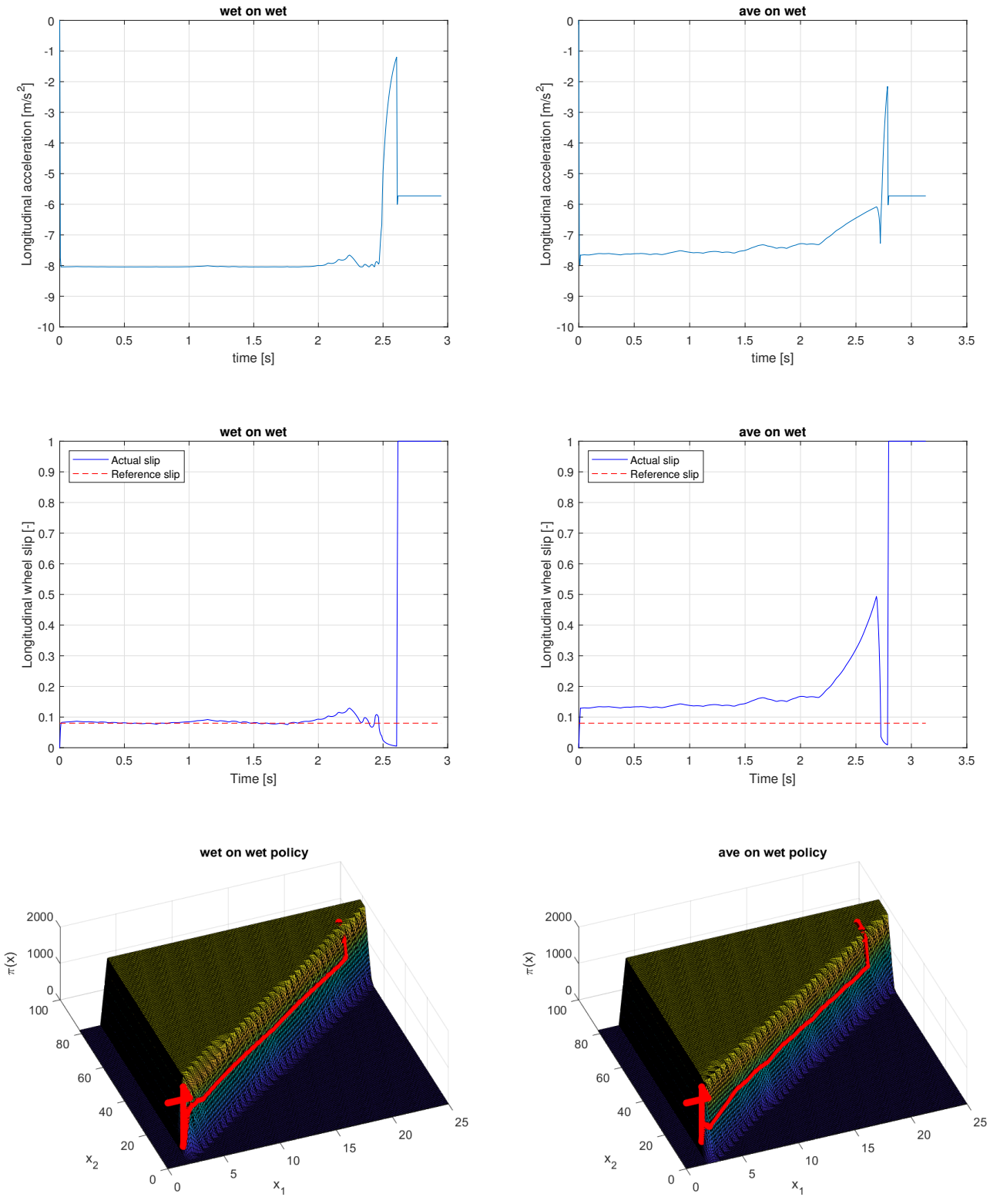


Figure 4.3: Performance comparison on wet asphalt

4.3. Results for piecewise linear approximated policy

Results for the linear piecewise approximated policy are shown and compared with that of the non linear policy for the initial speed of 80 km/h. Results for the the linear policy using same parameters for 60 km/h are also shown.

4.3.1. Initial speed = 80 km/h

The braking distance [m] for each policy on dry and wet asphalt are given in Table 4.5. The performance pattern is similar to that of the non-linear interpolation. The average policy performs better on dry and wet asphalt than wet and dry asphalt policies respectively. On dry asphalt, the average policy is 5.69% worse while the wet policy is 19.16% worse than the ideal. On wet asphalt, the average policy is 5.48% worse while the wet policy is 18.36% worse than the ideal. However, on comparing the braking distances with that of linear interpolation in Table 4.5, it is seen that on dry asphalt, the average policy performs exactly the same, while the dry policy performs marginally better and the wet policy performs worse. On wet asphalt, all the three policies perform marginally better. Thus, linear interpolation is the overall better performer in terms of braking distance, since braking on wet asphalt is more critical.

Table 4.5: Braking distance [m]

Surface	Dry asphalt policy	Wet asphalt policy	Average policy
Dry asphalt	25.31	30.16	26.75
Wet asphalt	37.27	31.04	32.75

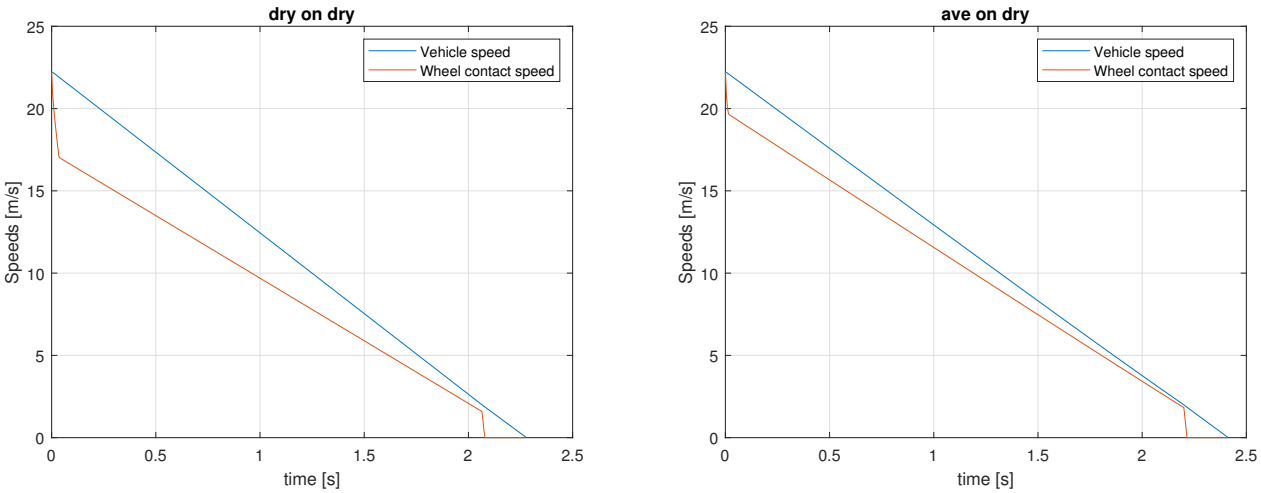
The standard deviation in deceleration [m/s^2] for each policy on dry and wet asphalt are given in Table 4.6. The pattern is similar to that of non-linear interpolation. Average on dry performs better than both dry on dry and wet on dry. Average on wet does not give the best performance but is better than wet on wet. On comparing these results with that of non-linear interpolation in Table 4.6, it is seen that on both surfaces, the comfort level has improved significantly for all policies.

Table 4.6: Standard deviation in deceleration [m/s^2]

Surface	Dry asphalt policy	Wet asphalt policy	Average policy
Dry asphalt	0.4869	0.5474	0.4481
Wet asphalt	0.4023	0.7967	0.6797

Performance on dry asphalt

As seen from the Figure 4.4, the performance of the average policy is good when compared to the ideal case i.e. dry on dry. The time taken to stop is marginally higher, the deceleration is slightly lower but mostly constant, the wheel slip is mostly constant, and the control policy is smooth i.e. no chattering. The only lacking part is that the wheel slip is almost half of the ideal value. On comparing the performance of non-linear interpolation in Figure 4.2 with that of linear interpolation in Figure 4.4 for policies of respective surfaces i.e. dry on dry, it is seen that for linear interpolation, the rate of change of wheel contact velocity, vehicle deceleration and wheel slip is more smooth, and chattering in the control input is lower. Also, the average policy reduces variations in wheel angular speed, wheel slip, vehicle deceleration and control input to a large extent.



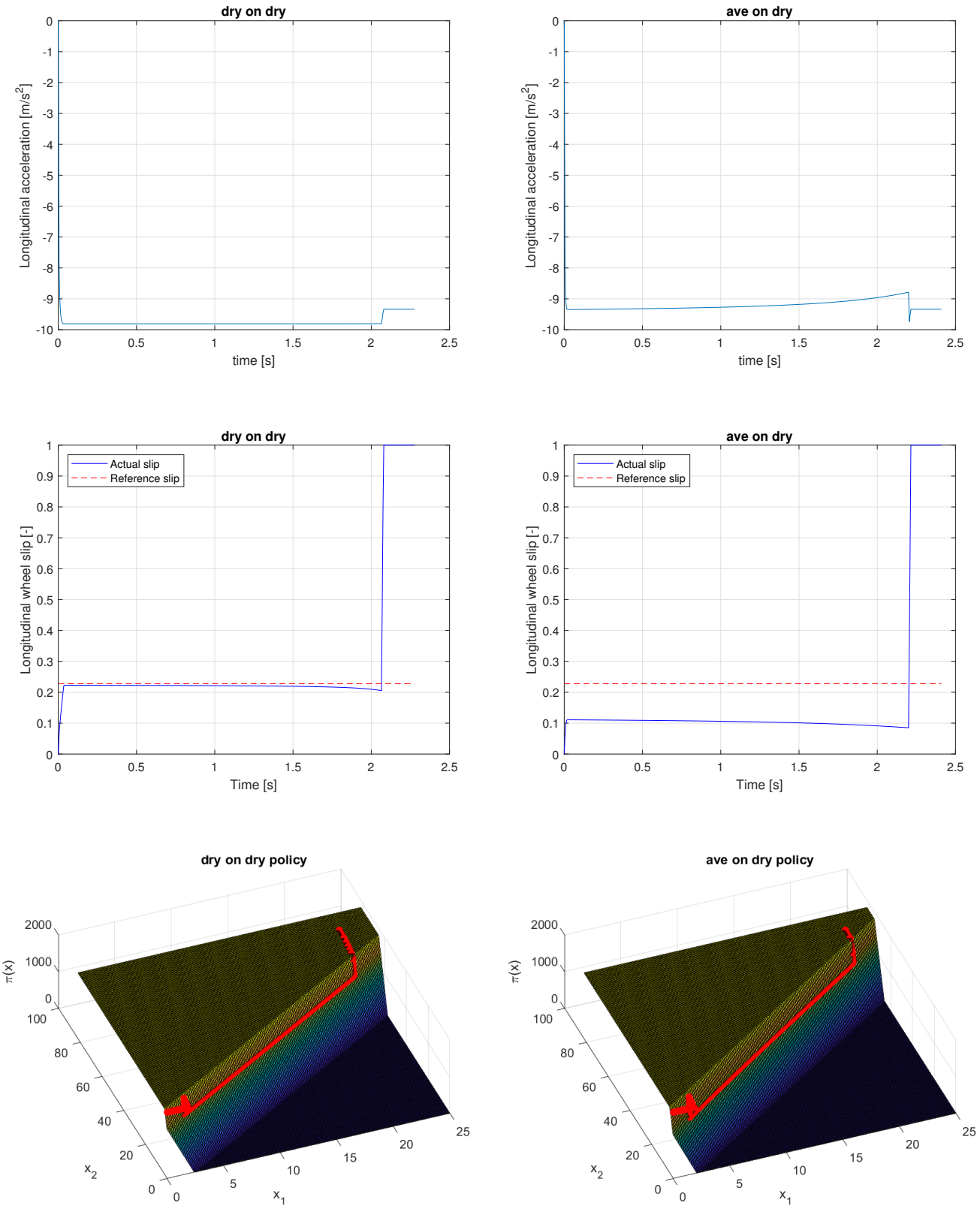
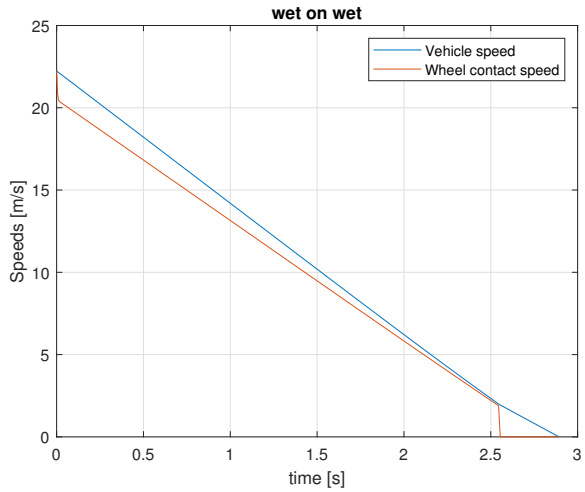


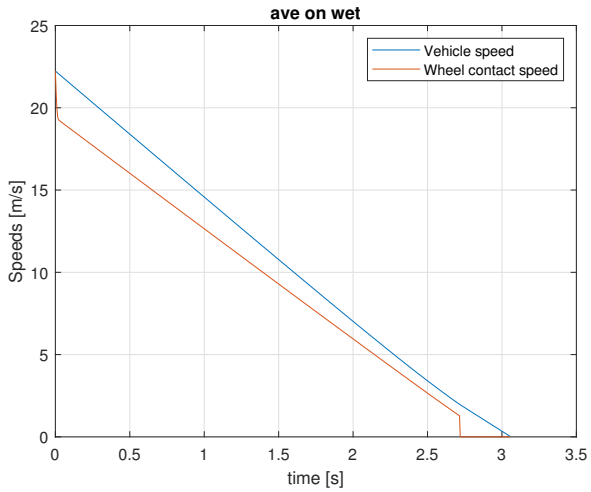
Figure 4.4: Performance comparison on dry asphalt

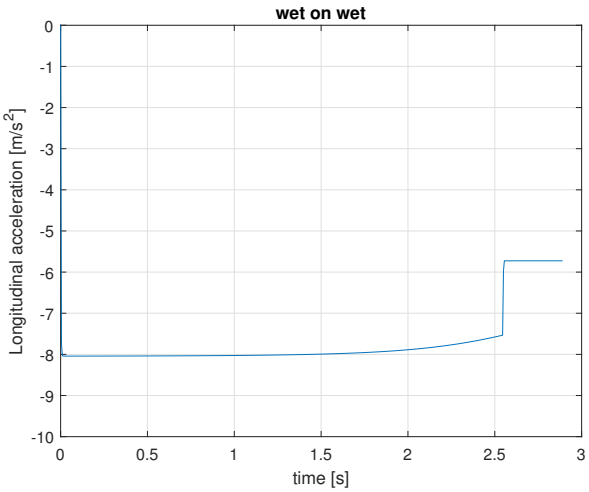
Performance on wet asphalt

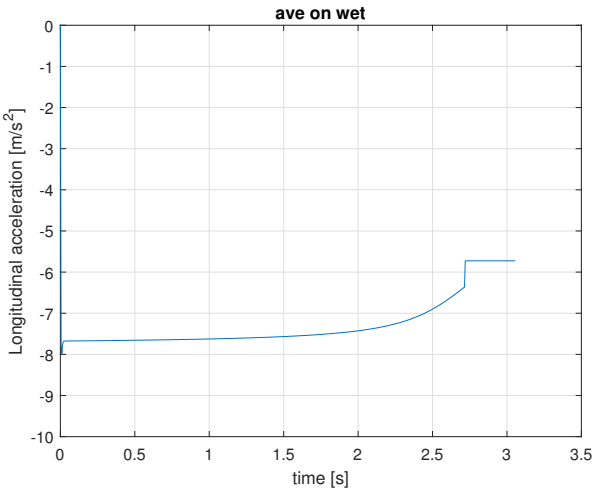
As seen from the Figure 4.5, the performance of the average policy is close to the ideal level i.e. wet to wet. The time taken to stop is marginally higher, the deceleration is slightly lower but mostly constant,

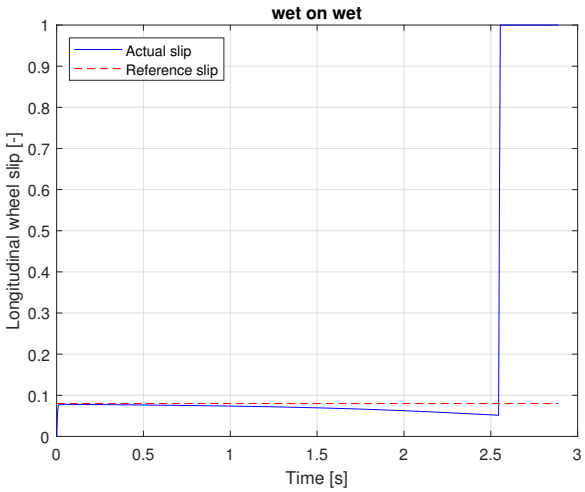
the wheel slip is mostly constant and close to the ideal value, and the control policy is smooth i.e. no chattering. On comparing the performance of non-linear interpolation in Figure 4.3 with that of linear interpolation in Figure 4.5 for wet on wet, it is seen that for linear interpolation, the rate of change of wheel contact velocity, vehicle deceleration and wheel slip is more smooth, and chattering in the control input is much lower. Also, the average policy is able to reduce variations in wheel angular speed, vehicle deceleration, wheel slip and control input to a large extent.

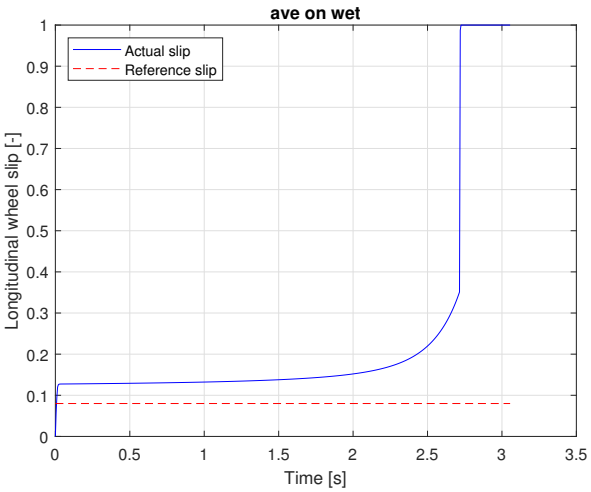












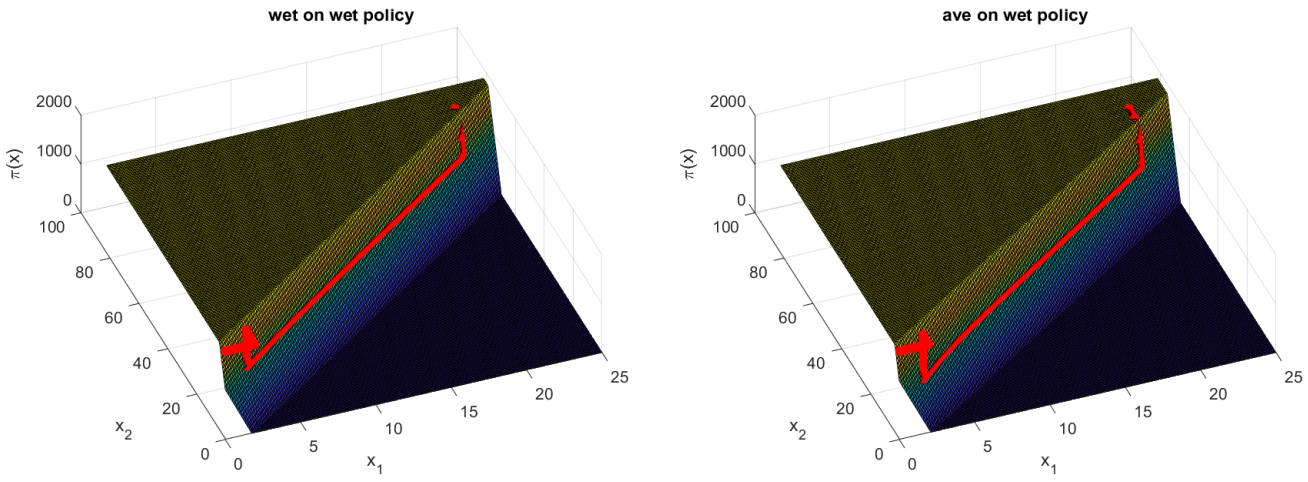


Figure 4.5: Performance comparison on wet asphalt

4.3.2. Initial speed = 60 km/h

To test the robustness of the control policy, it was tested for an initial speed of 60 km/h using the same parameters as derived for 80 km/h. The braking distances on dry and wet asphalt are given in Table 4.7. The policies are found to work well on both surfaces, but not as good as for 80 km/h. On dry asphalt, the average policy is 6.24% worse while the wet policy is 20.7% worse than the ideal. On wet asphalt, the average policy is 6.09% worse while the wet policy is 20.67% worse than the ideal.

Table 4.7: Braking distance [m]

Surface	Dry asphalt policy	Wet asphalt policy	Average policy
Dry asphalt	14.25	17.2	15.14
Wet asphalt	21.19	17.56	18.63

The standard deviation in deceleration [m/s^2] for each policy on dry and wet asphalt are given in Table 4.8. The comfort level follows a similar pattern to that of 80 km/h. Average on dry performs better than both dry on dry and wet on dry. Average on wet performs better than wet on wet but worse than dry on wet. However, the magnitudes of the standard deviation have increased for all the cases as compared to those of 80 km/h. This is strange since the initial speed has reduced. This result again shows that the policy parameters for 80 km/h do not work as well for 60 km/h.

Table 4.8: Standard deviation in deceleration [m/s^2]

Surface	Dry asphalt policy	Wet asphalt policy	Average policy
Dry asphalt	0.5531	0.6173	0.5047
Wet asphalt	0.4297	0.879	0.7335

The average policies on dry and wet asphalt are compared with the ideal values in Figure 4.6 and Figure 4.7 respectively. Although the performance for 60 km/h is not as good as for 80 km/h, the control inputs are free from chattering. The analysis in this section shows that policy parameters derived for higher speeds can be used for lower speeds with marginal decrease in performance.

Policy comparison on dry asphalt

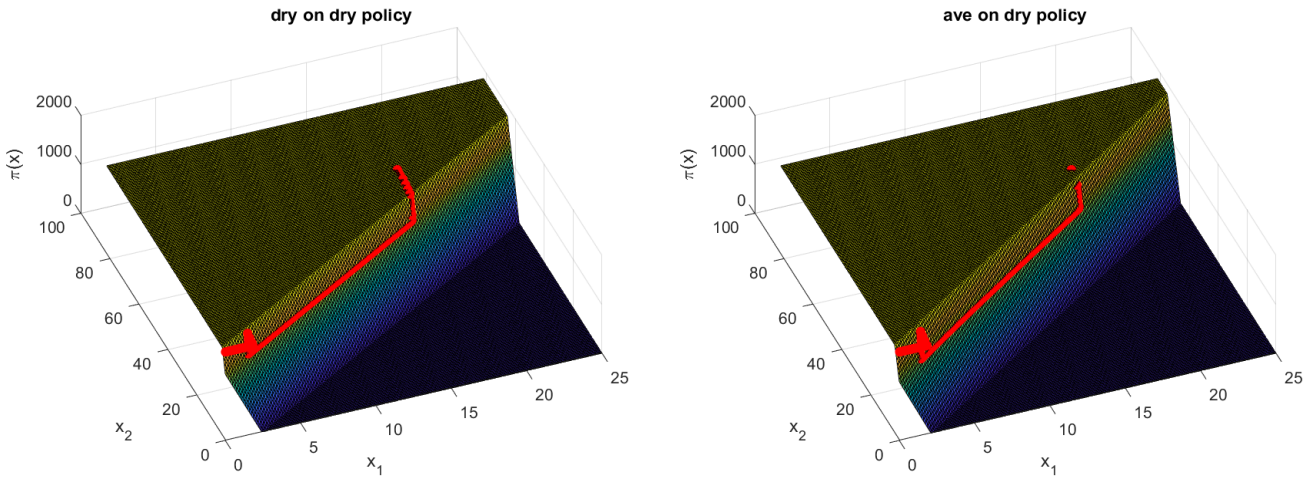


Figure 4.6: Policy comparison on dry asphalt

Policy comparison on wet asphalt

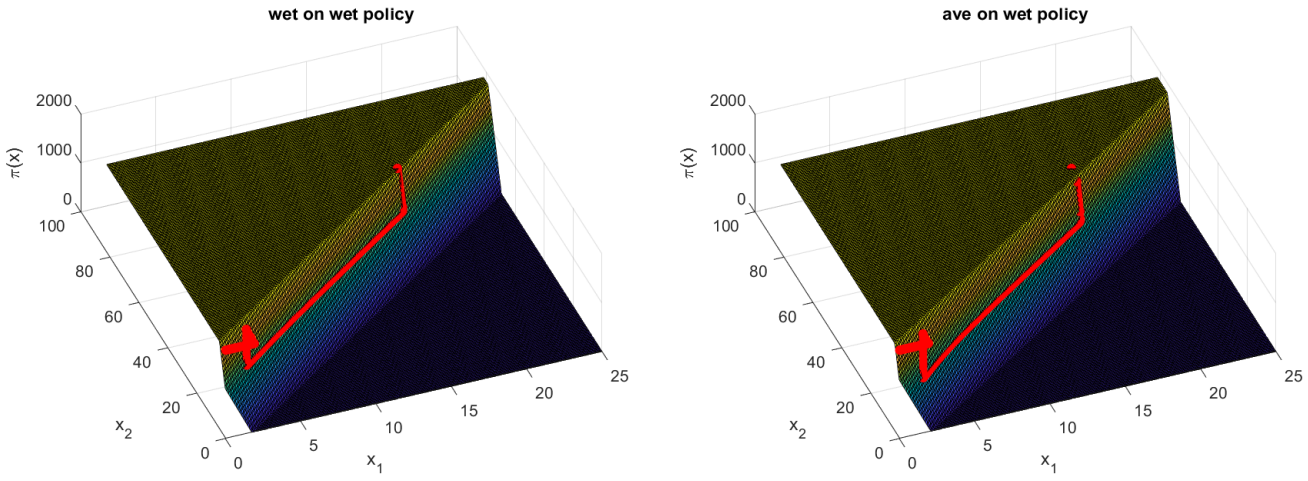


Figure 4.7: Policy comparison on wet asphalt

4.4. Summary

In this chapter, the results for the two interpolation methods for deriving the policy from optimal actions computed offline were shown. The parameters of each linear policy and Fuzzy-V iteration were also shown and discussed. The piecewise linear approximation is found to perform better than non-linear interpolation in terms of smoothness of control policy, deceleration, wheel slip, and wheel linear and contact velocities, as well as in terms of braking distance and comfort. The same policy can also be used for lower speeds without a significant drop in performance. Thus, the policies obtained through piecewise linear approximation will be used for adaptation in Chapter 5. The piecewise linear policy is found to be similar to that of a proportional controller with gain scheduling.

5

Policy and PI controller adaptation results

There are three possible variations of PoWER:

1. The parameter variance can be kept constant or can be adapted for each episode.
2. The random exploration during each episode can be kept constant or can be varied at each time step of the episode.
3. Equal or different random exploration can be used for each parameter.
4. Each basis function or only one of the basis functions can be assumed to be active at each time step.

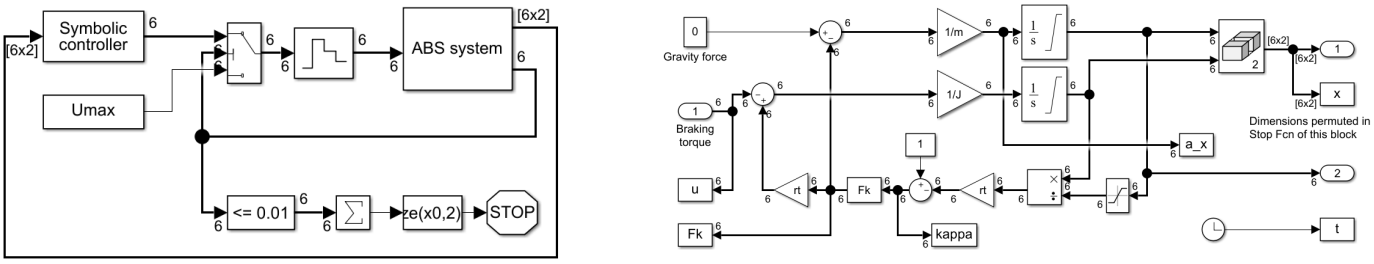
Constant and different random exploration for each parameter, and the assumption of activation of only one basis function at a time step has given better results for adaptive parameter variance than time varying exploration, constant parameter variance and the assumption that all basis functions are active simultaneously. The results have been shown for 4 adaptations:

1. Wet asphalt policy to dry asphalt (called wet to dry)
2. Average policy to dry asphalt (called ave to dry)
3. Dry asphalt policy to wet asphalt (called dry to wet)
4. Average policy to wet asphalt (called ave to wet)

For each adaptation, an optimal initial variance vector was found out by trial and error. Optimality here is defined in terms of the final return, not the speed of convergence. For each adaptation, the result for this optimal initial variance (named as optimal adaptive parameter variance in graphs) has been compared with that of an non-optimal initial variance vector (named as adaptive parameter variance in graphs) found out by averaging the four initial variance vectors. This is done because a single initial variance vector must be used for robust performance. The value of the initial robust variance vector obtained is $\sigma^2 = [25\ 37\ 317]^\mathsf{T}$.

5.1. Algorithm parameters

The Simulink model used is shown in Figure 5.1. Multiple wheel slip initial conditions have been taken into account i.e. $\kappa_0 = [0\ 0.2\ 0.4\ 0.6\ 0.8\ 1]$ and are referred to as Slip 1, Slip 2, Slip 3, Slip 4, Slip 5, Slip 6 respectively. The initial chassis speed is kept the same i.e. 80 km/h for all wheel slips. The return of the j^{th} episode is calculated using $R_j = \rho_{\text{off}} - d_{x,j}$ where ρ_{off} is the reward offset and $d_{x,j}$ is the average braking distance for all initial conditions for that episode. The reward offset is taken to be 200 to ensure that the return is positive as PoWER requires the episode return to be positive [17]. The number of iterations are taken to be 300 as these were sufficient to show return and parameter convergence while not taking too much time (approx 60 seconds on a dual core computer). After every 10 iterations, a noiseless trial i.e. an episode without random exploration but with parameters updated through importance sampling is conducted to test the true performance.



(a) The controller structure

(b) The ABS system

Figure 5.1: The Simulink model

5.2. Piecewise linear policy adaptation results

The braking distances on dry and wet asphalt are given in Table 5.1. On dry asphalt, the braking distances of ave to dry are slightly lesser than those of wet to dry for both optimal and robust variances, for each initial condition. For wet to dry, braking distances of optimal variances are slightly lower than that of robust variances for all initial conditions. For ave to dry, braking distances for both variances are same. On wet asphalt, braking distances of ave to wet for both optimal and adaptive variances are lesser than those of robust variances of dry to wet, but are mostly same as those of optimal variances of dry to wet, for all initial conditions. The braking distances of optimal variances are lower than robust variances for all initial conditions. When compared with the ideal braking distances from Chapter 4 i.e. 25.31 m for dry on dry and 31.04 m for wet on wet, the adaptations to dry asphalt are successful with all initial conditions except 1 achieving a lower braking distance, and adaptations to wet asphalt achieving a lower braking distance for 3 out of 6 initial conditions with the maximum increment being 0.34 m.

Table 5.1: Braking distance [m]

Case	Method	Slip 1	Slip 2	Slip 3	Slip 4	Slip 5	Slip 6
Wet to Dry	Optimal variance	25.32	25.25	25.25	25.26	25.26	25.28
	Robust variance	25.33	25.26	25.26	25.26	25.27	25.28
Ave to Dry	Optimal variance	25.31	25.24	25.24	25.25	25.25	25.26
	Robust variance	25.31	25.24	25.24	25.24	25.25	25.26
Dry to Wet	Optimal variance	30.88	30.86	30.94	31.05	31.18	31.31
	Robust variance	30.96	30.94	31.01	31.12	31.25	31.38
Ave to Wet	Optimal variance	30.88	30.86	30.95	31.06	31.18	31.31
	Robust variance	30.88	30.87	30.94	31.05	31.18	31.31

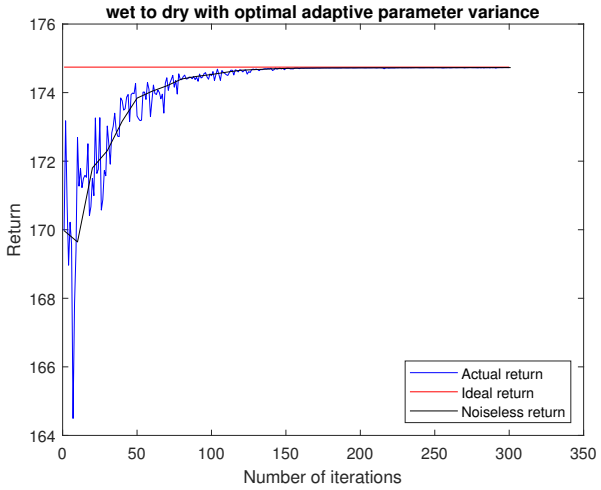
The standard deviation in deceleration [m/s^2] for each adaptation are given in Table 5.2. On dry asphalt, the comfort performance of optimal variances is better than that of robust variances for each respective initial condition, and the difference in performance is lesser for ave to dry and wet to dry. Also, the deviation values are much less for Slip 2 - Slip 6 as compared to Slip 1. The same pattern is seen on wet asphalt. When compared to ideal values from Chapter 4 i.e. 0.4481 for average on dry and 0.4023 for dry on wet, it is seen that the adaptations to dry asphalt perform well, but adaptations to wet asphalt do not.

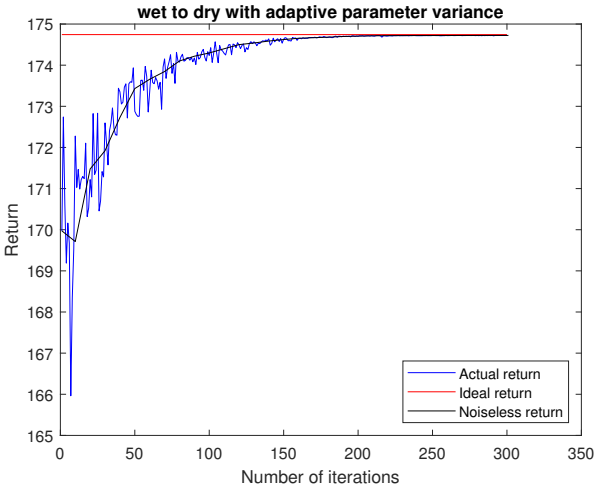
Table 5.2: Standard deviation in acceleration [m/s^2]

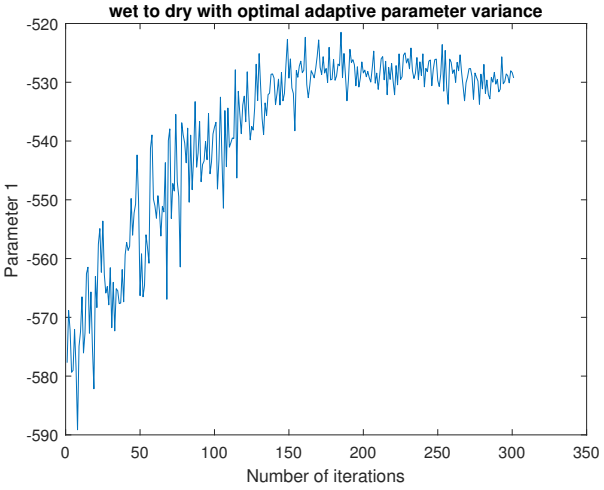
Case	Method	Slip 1	Slip 2	Slip 3	Slip 4	Slip 5	Slip 6
Wet to Dry	Optimal variance	0.4871	0.1409	0.1408	0.1411	0.1424	0.1451
	Robust variance	0.4891	0.1492	0.1491	0.1493	0.1504	0.1527
Ave to Dry	Optimal variance	0.4862	0.134	0.134	0.1345	0.1361	0.1392
	Robust variance	0.4864	0.134	0.134	0.1345	0.1361	0.1393
Dry to Wet	Optimal variance	0.8303	0.7656	0.7725	0.7808	0.7923	0.8016
	Robust variance	0.8365	0.774	0.7762	0.7836	0.794	0.8061
Ave to Wet	Optimal variance	0.8302	0.7655	0.7724	0.7806	0.7922	0.8014
	Robust variance	0.8307	0.7705	0.7729	0.7812	0.7927	0.8020

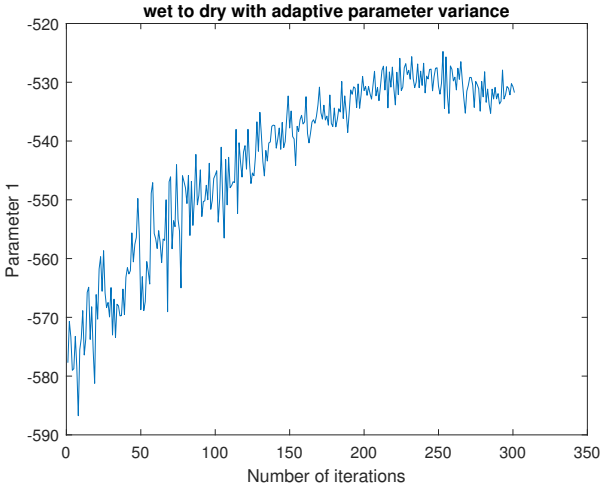
5.2.1. Adaptation of wet asphalt policy to dry asphalt

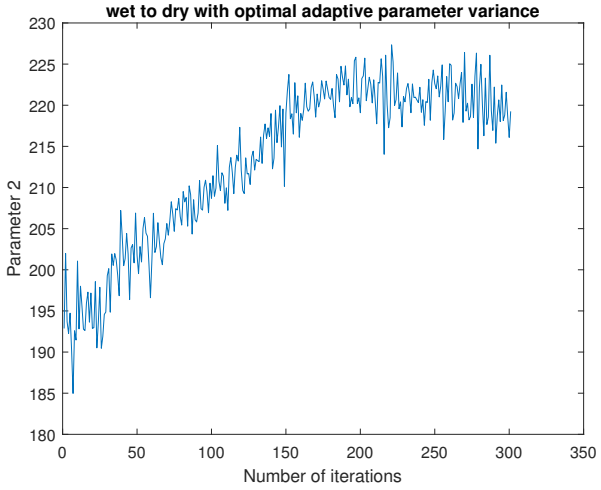
The result comparison of optimal and robust parameter variance for wet to dry is shown in Figure 5.2. The initial optimal variance vector is taken as $\sigma^2 = [40\ 60\ 500]^\mathsf{T}$. The return converges faster for optimal variance as it is greater than than robust variance. The final return for both variances is the same.

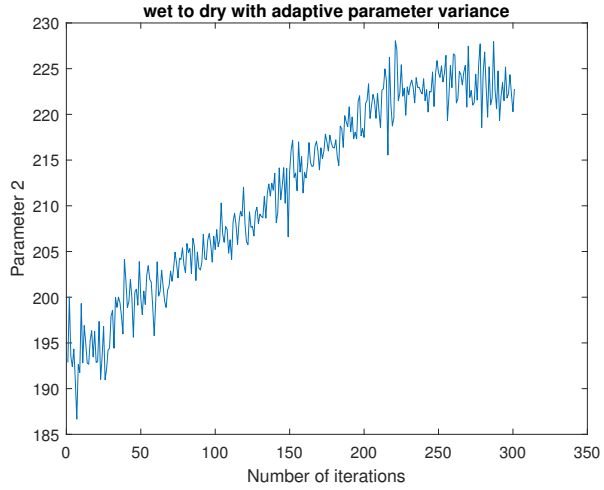


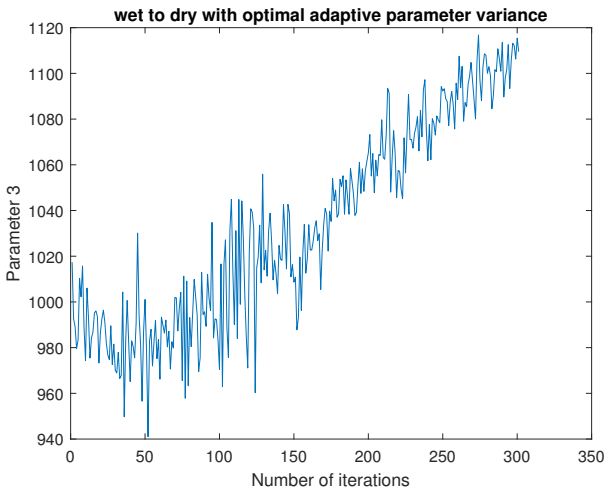


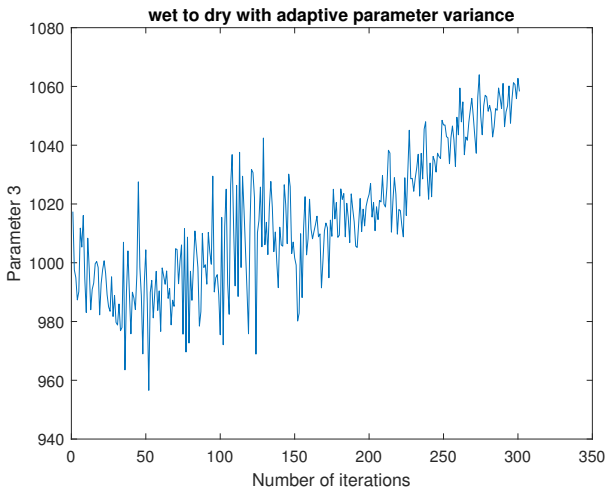




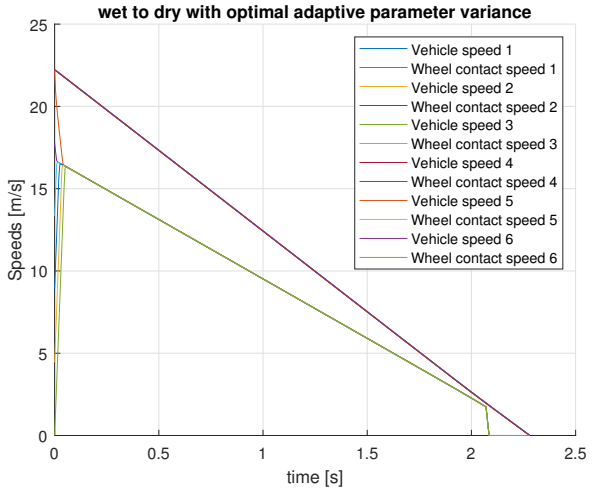


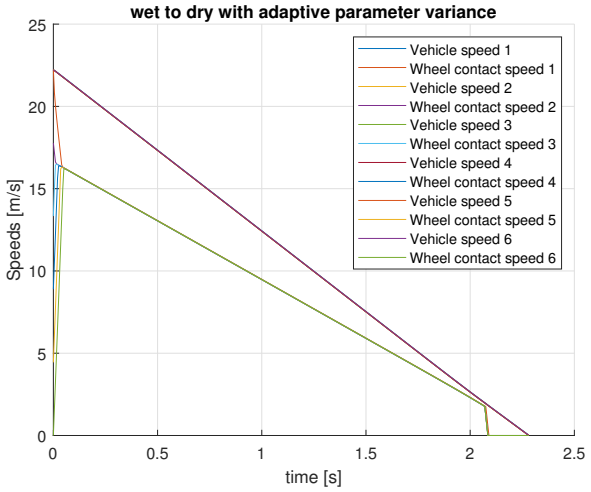


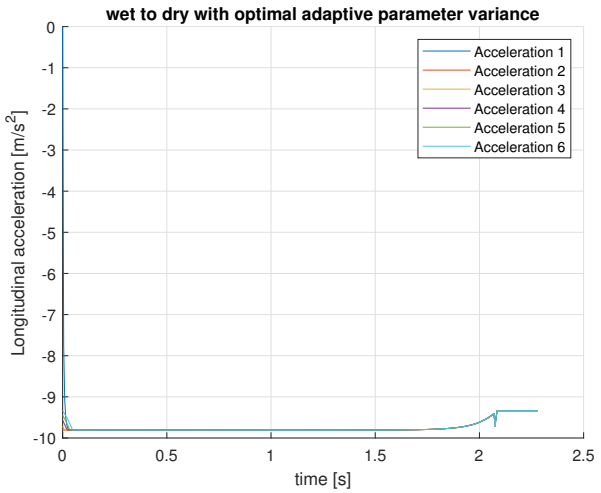


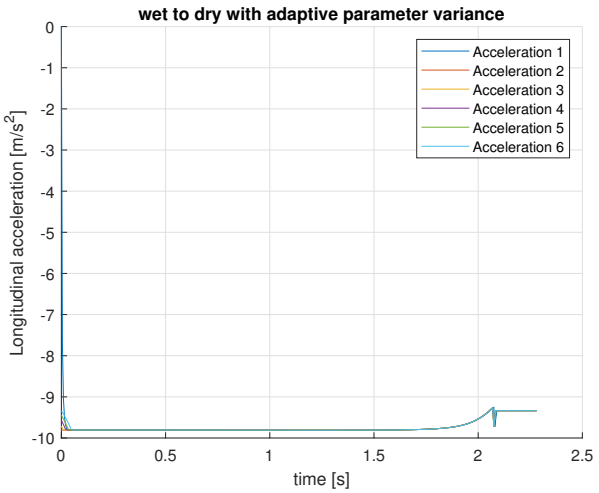


The parameters show convergent behaviour for both variances and converge to similar values except for parameter 3. The optimal and robust variances perform the same in terms of convergence of speeds, deceleration and wheel slip. The control input is free from chattering for both.









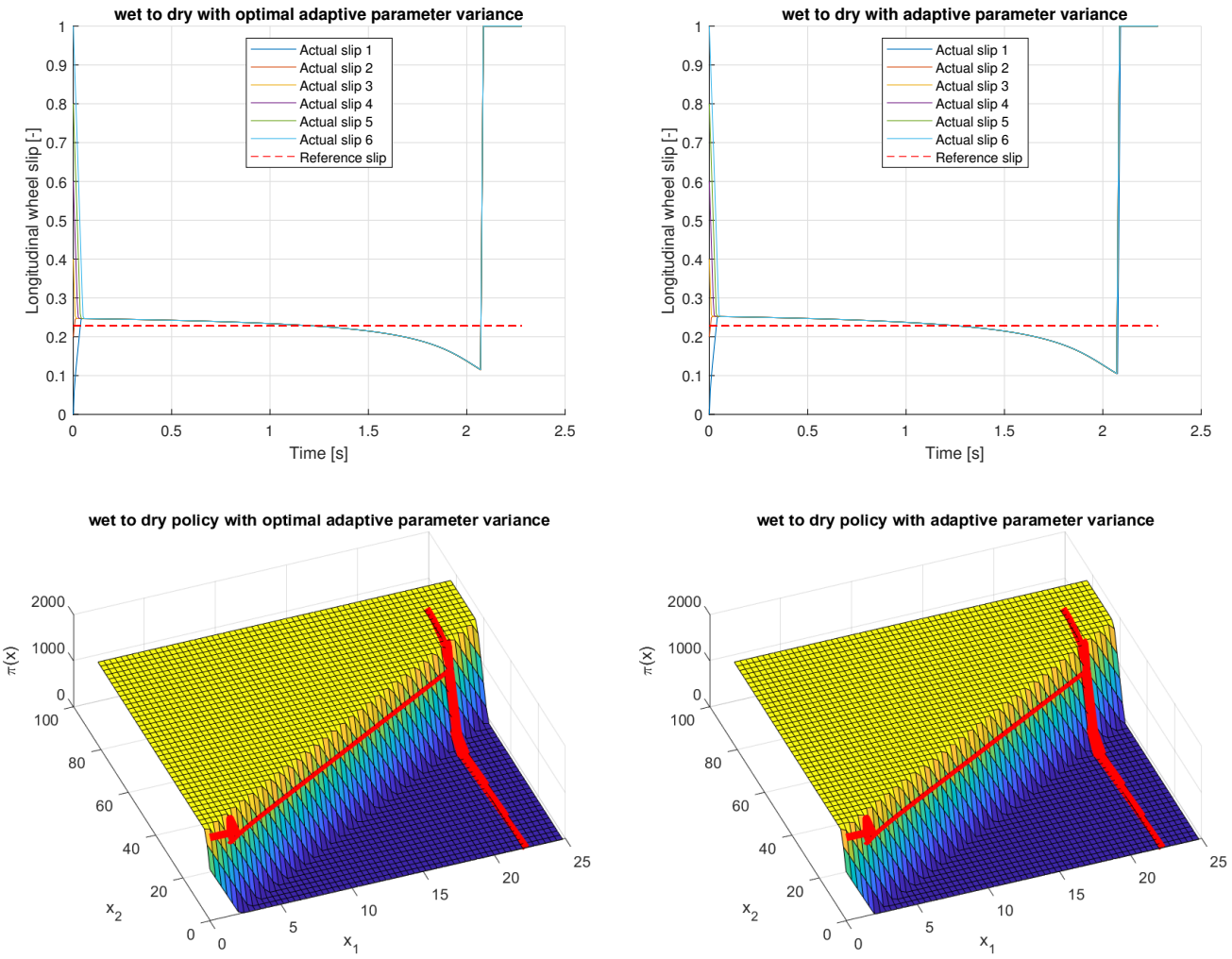
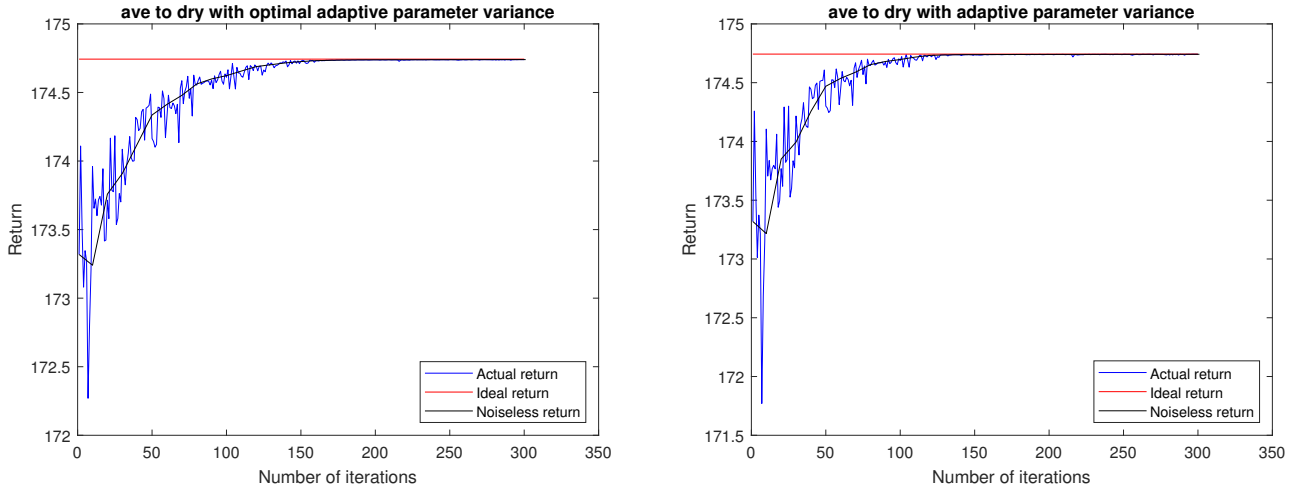


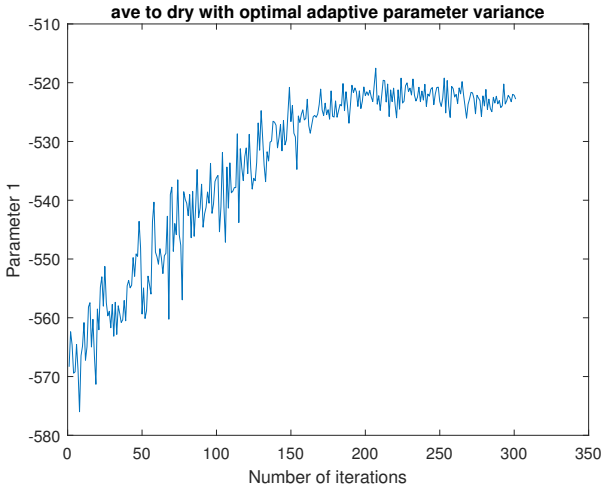
Figure 5.2: Performance comparison for wet to dry

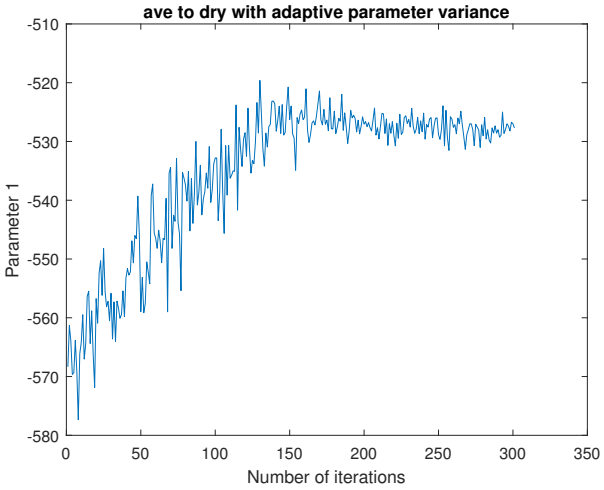
5.2.2. Adaptation of average policy to dry asphalt

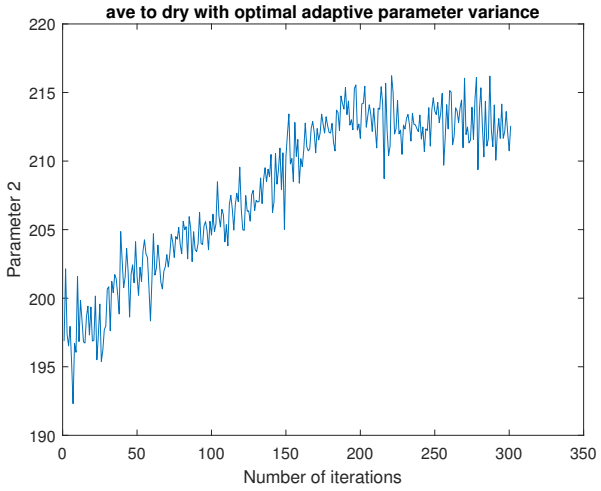
The result comparison of optimal and robust parameter variance for ave to dry is shown in Figure 5.3. The initial optimal variance vector is taken as $\sigma^2 = [18\ 20\ 200]^\mathrm{T}$. The return converges faster for robust variance as it is greater than than optimal variance. The final return for both variances is the same.

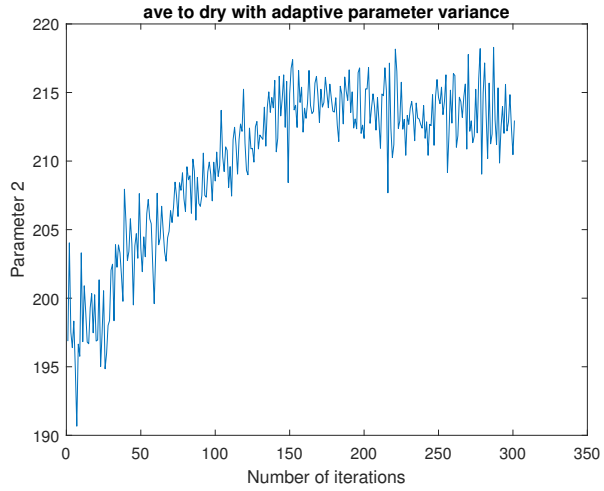


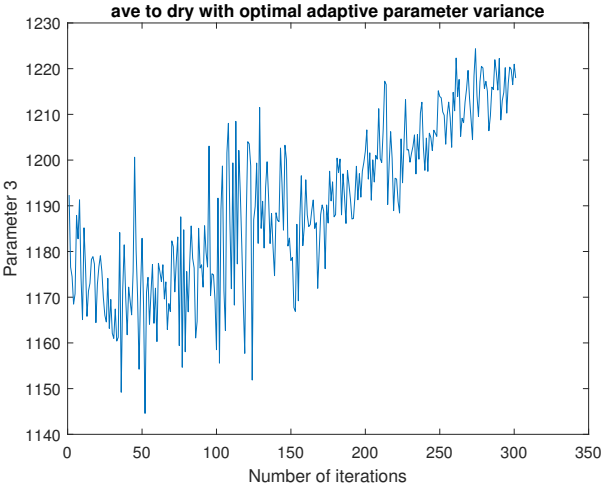
The parameters show convergent behaviour for both variances and converge to similar values except for parameter 3.

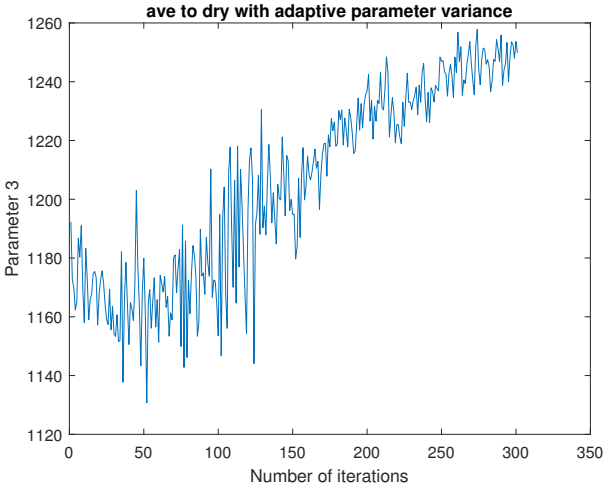




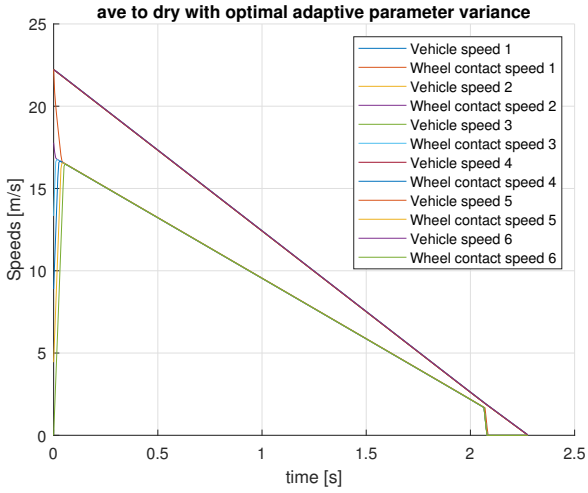


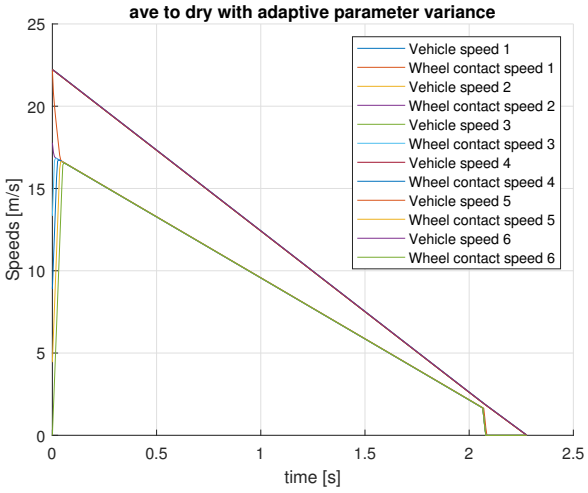


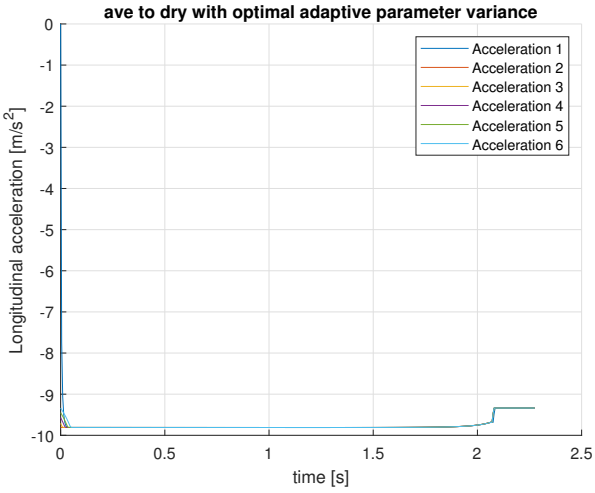


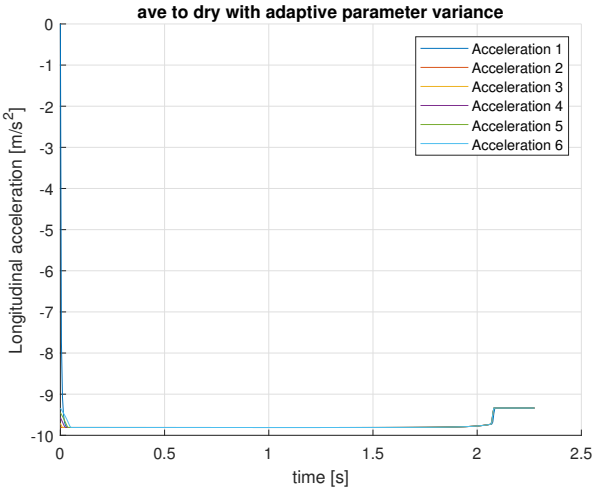


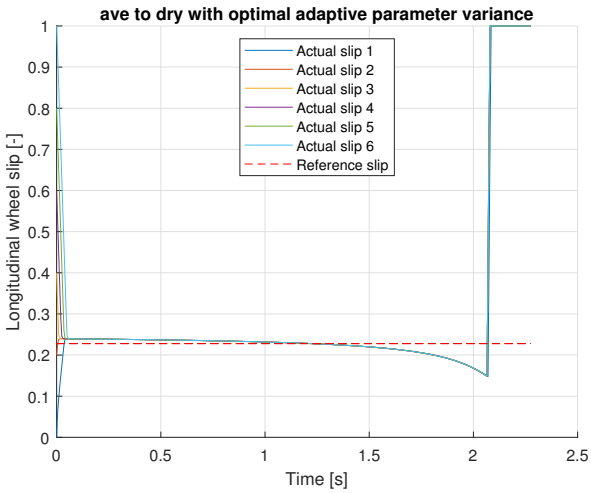
The optimal and robust variances perform the same in terms of convergence of speeds, deceleration and wheel slip. The control input is free from chattering for both.

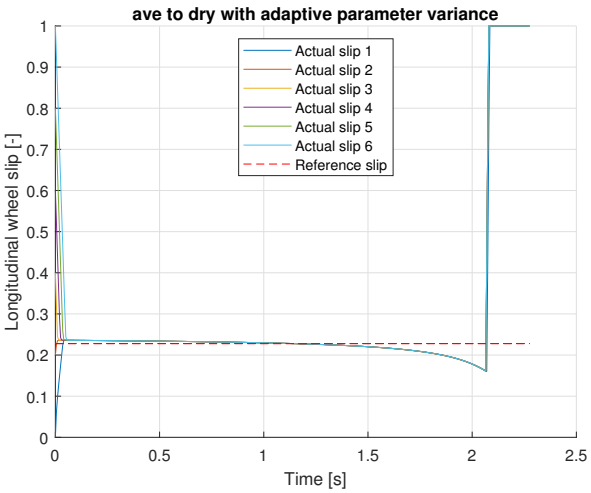












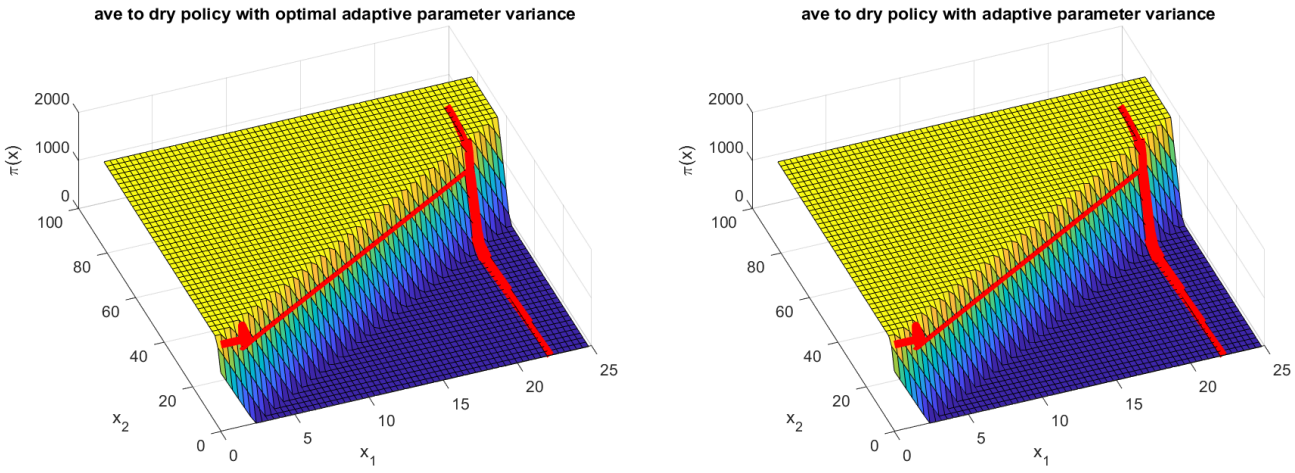
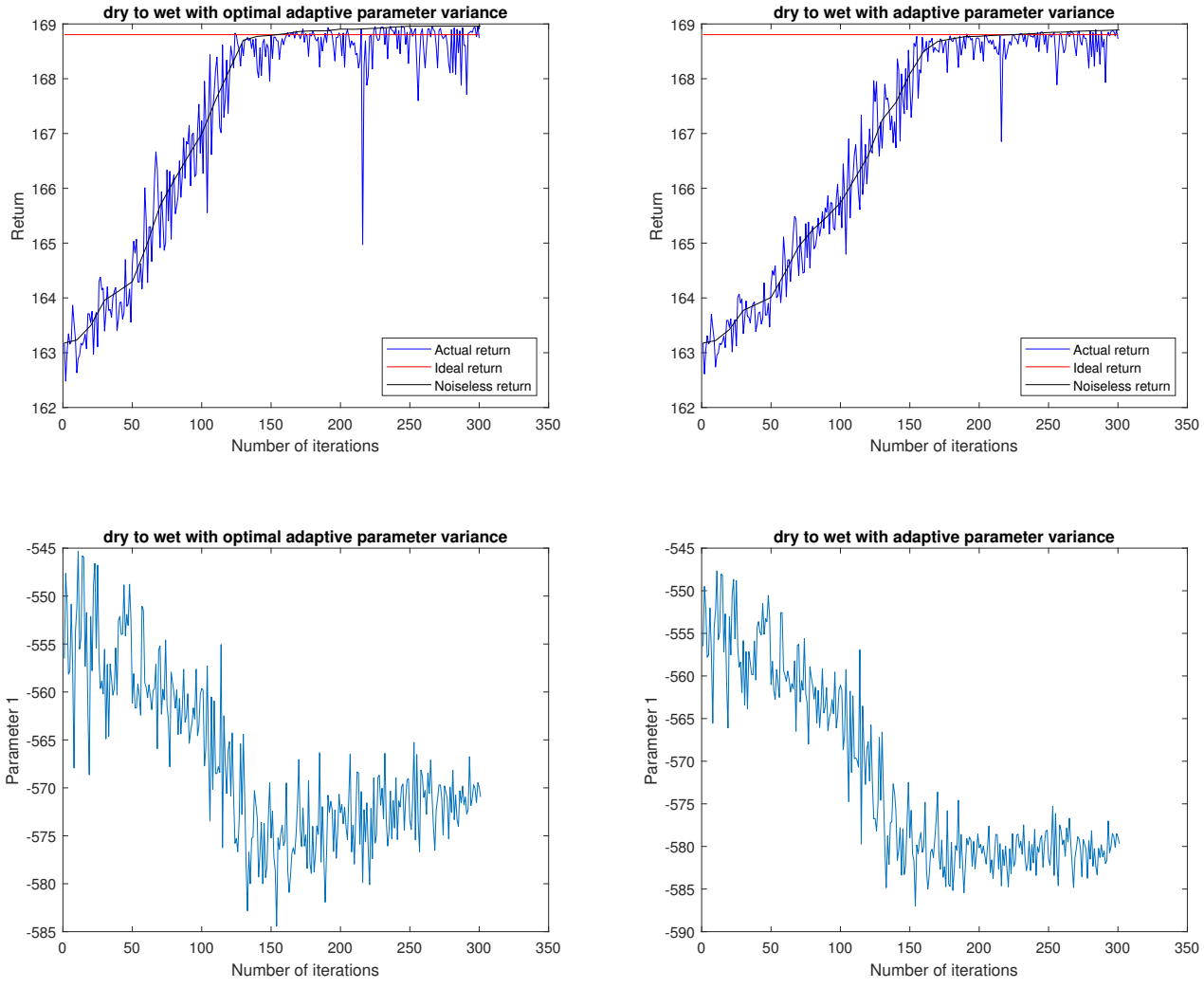
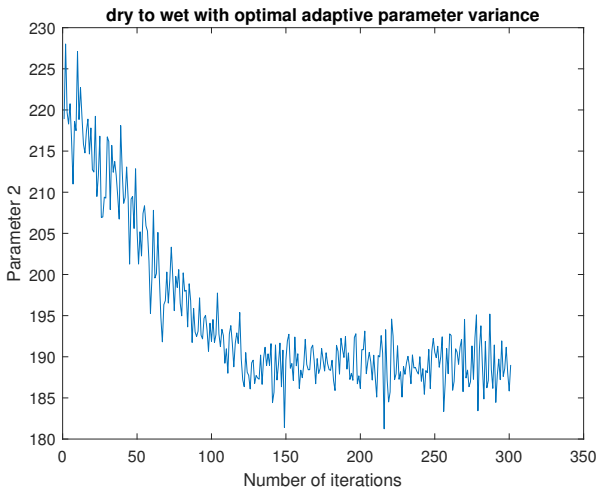


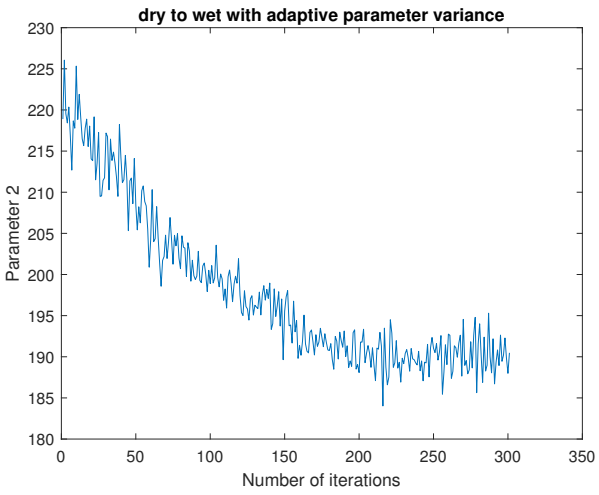
Figure 5.3: Performance comparison for ave to dry

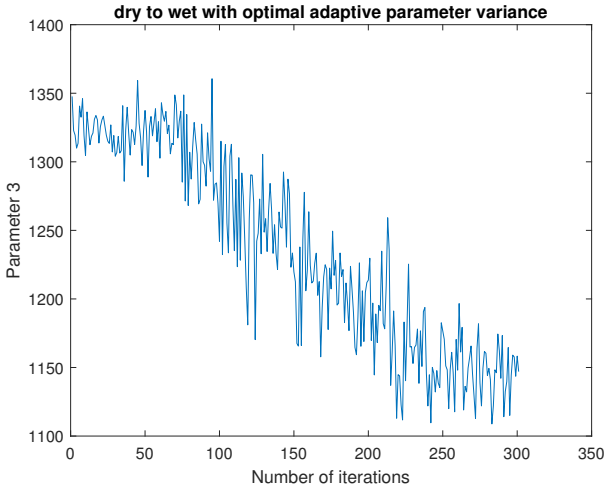
5.2.3. Adaptation of dry asphalt policy to wet asphalt

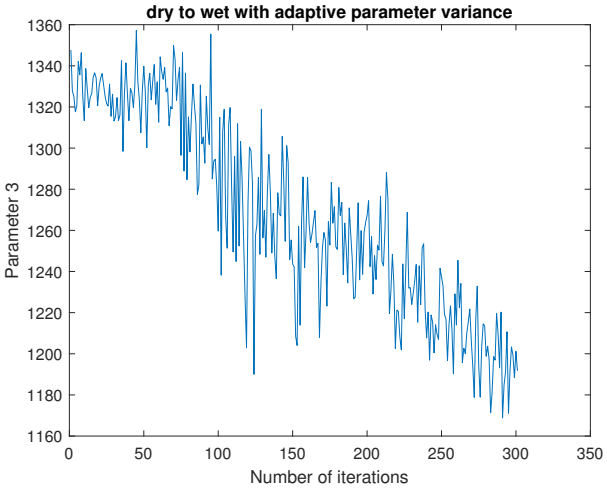
The result comparison of optimal and robust parameter variance for dry to wet is shown in Figure 5.4. The initial optimal variance vector is taken as $\sigma^2 = [40\ 60\ 500]^\mathsf{T}$. The return converges faster for optimal variance as it is greater than than robust variance. It also converges to a slightly higher value.



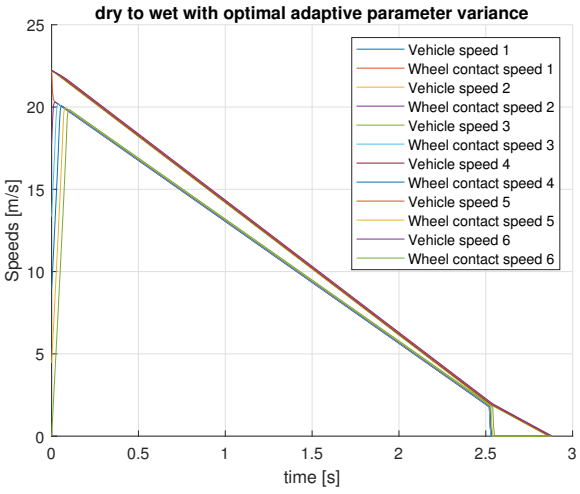


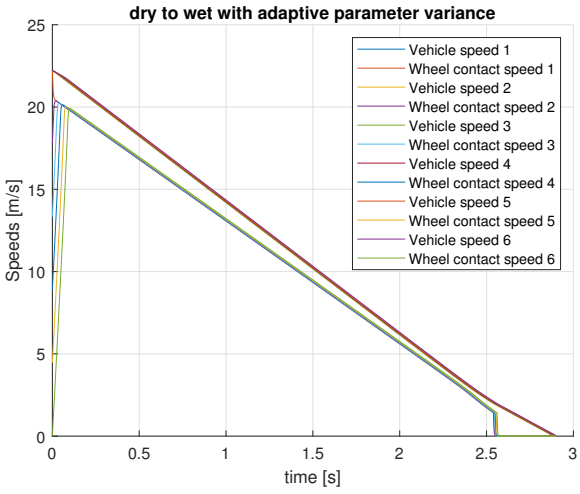






The parameters show convergent behaviour for both variances and converge to different values as the final return is different. The optimal variance performs marginally better than robust variance in terms of convergence of speeds, deceleration and wheel slip. The control input is free from chattering for both.





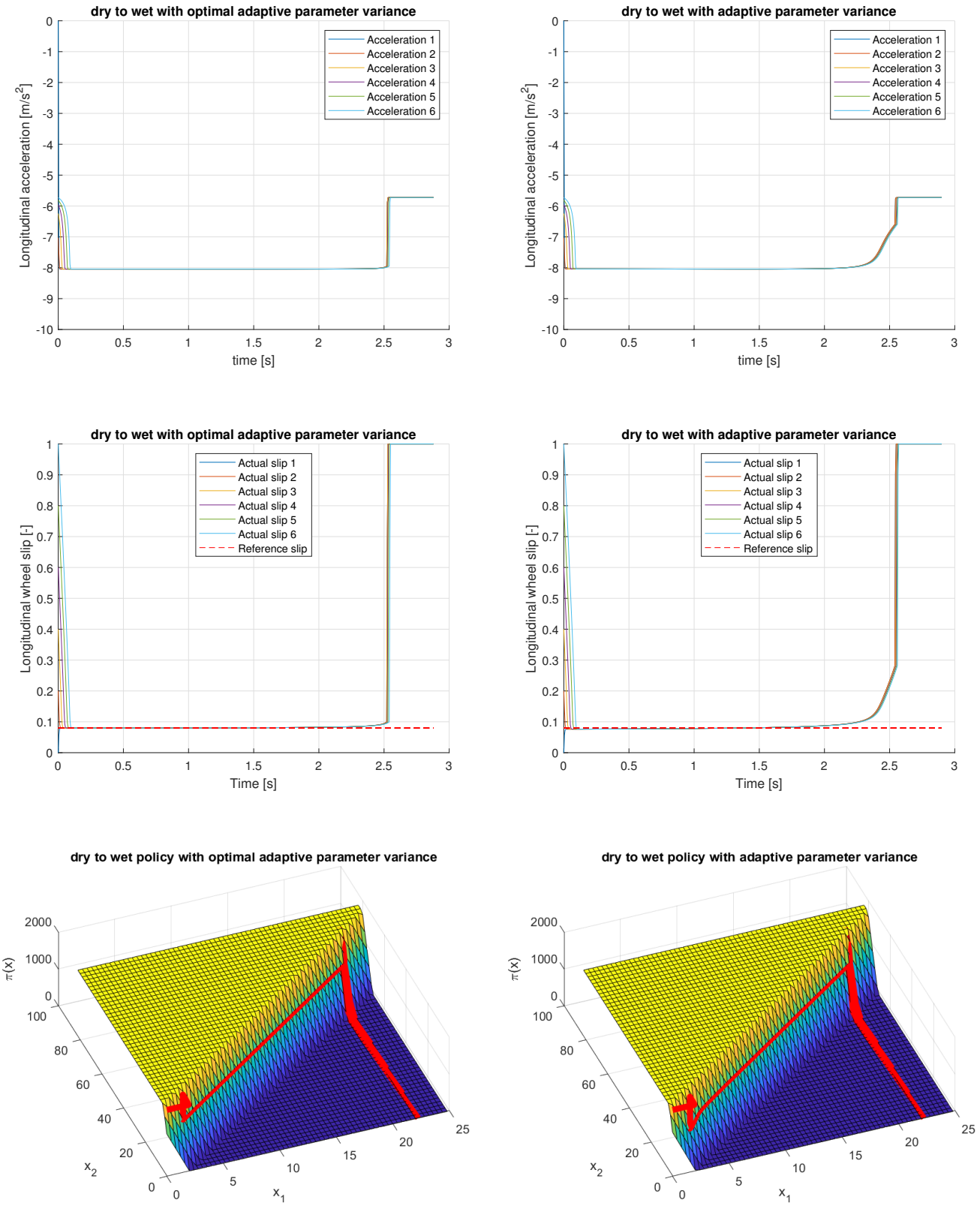
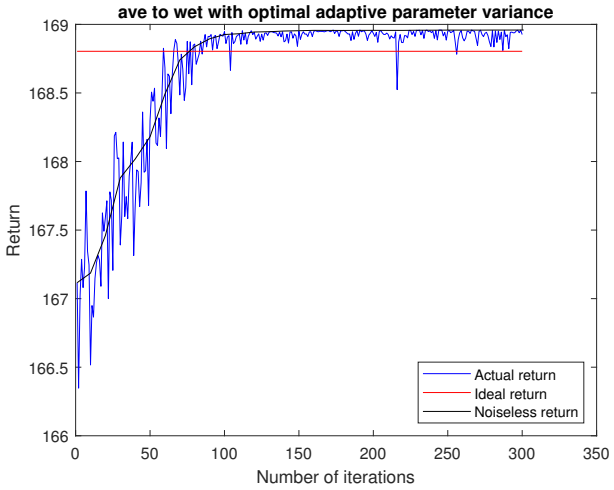


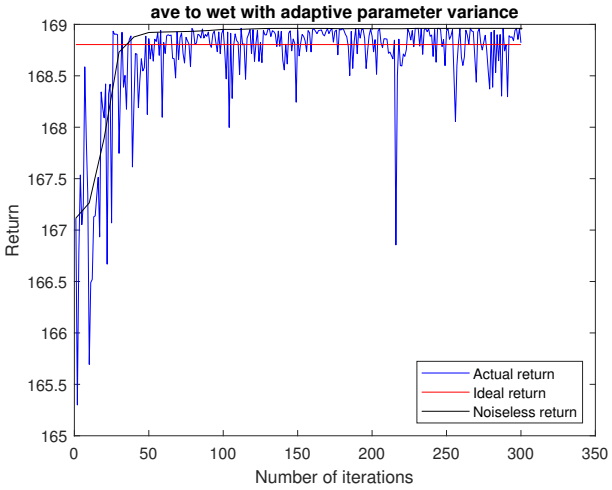
Figure 5.4: Performance comparison for dry to wet

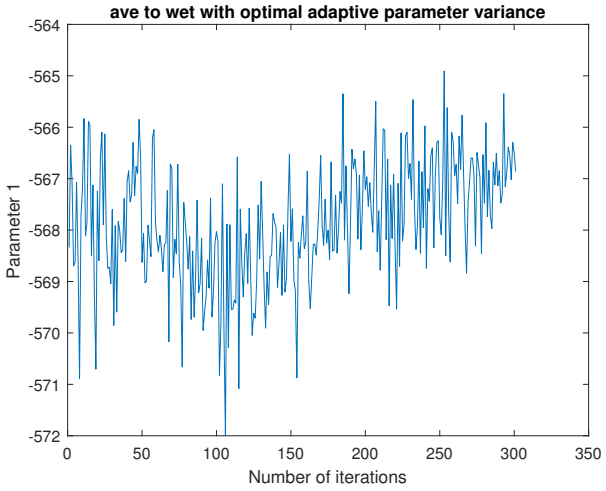
5.2.4. Adaptation of average policy to wet asphalt

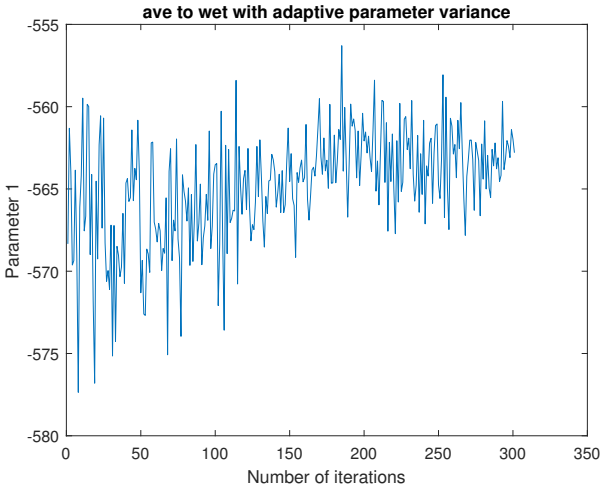
The result comparison of optimal and robust parameter variance for ave to wet is shown in Figure 5.5. The initial optimal variance vector is taken as $\sigma^2 = [2\ 7\ 68]^\mathrm{T}$. The return converges faster for robust

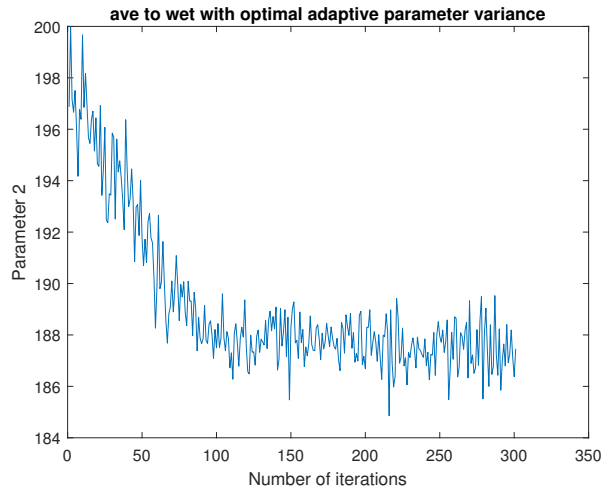
variance as it is greater than than optimal variance. The final returns are nearly equal. Random drops in the return are much less sharper for optimal variance.

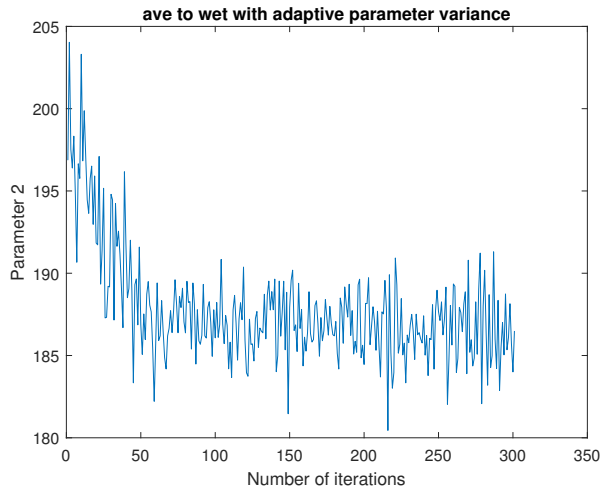


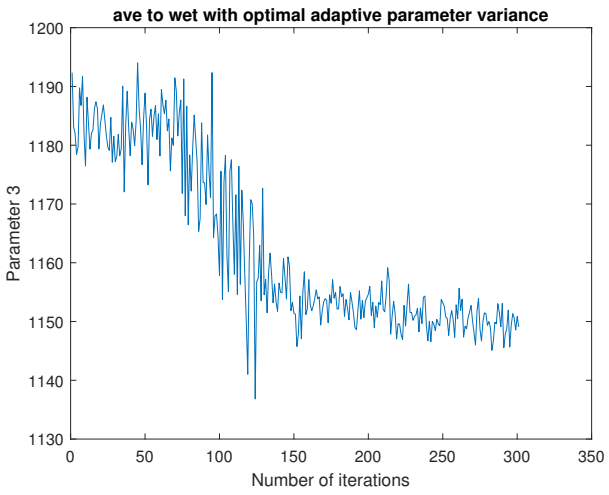


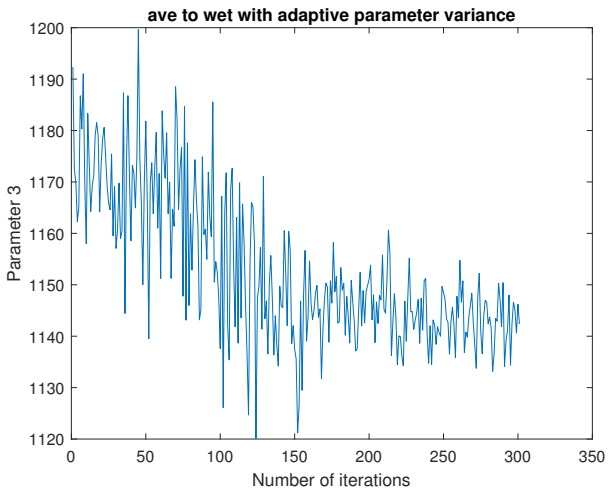




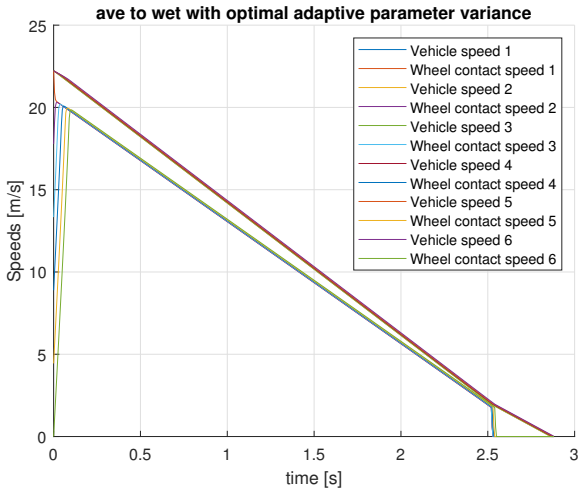


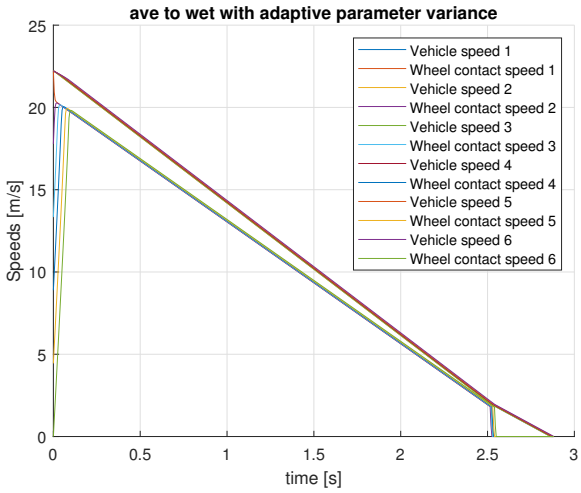


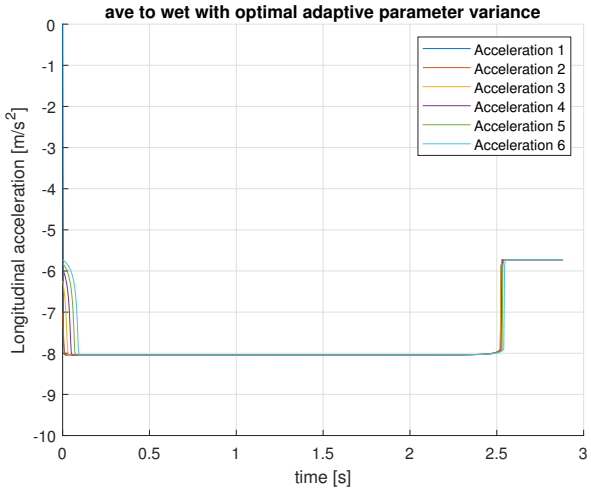


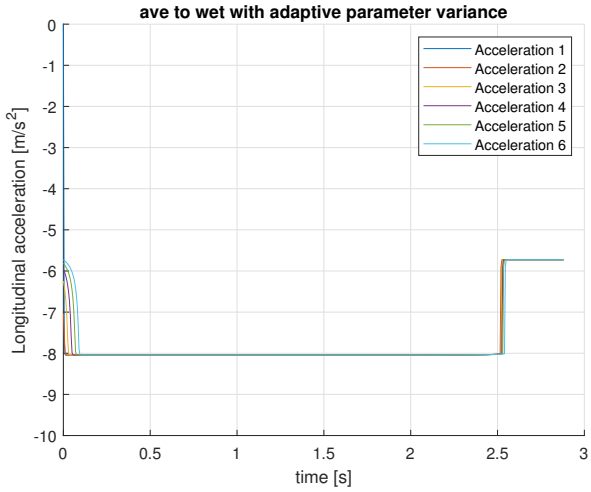


The parameters show convergent behaviour for both variances and converge to similar values. The optimal and robust variances perform almost the same in terms of convergence of speeds, deceleration and wheel slip. The control input is free from chattering for both.









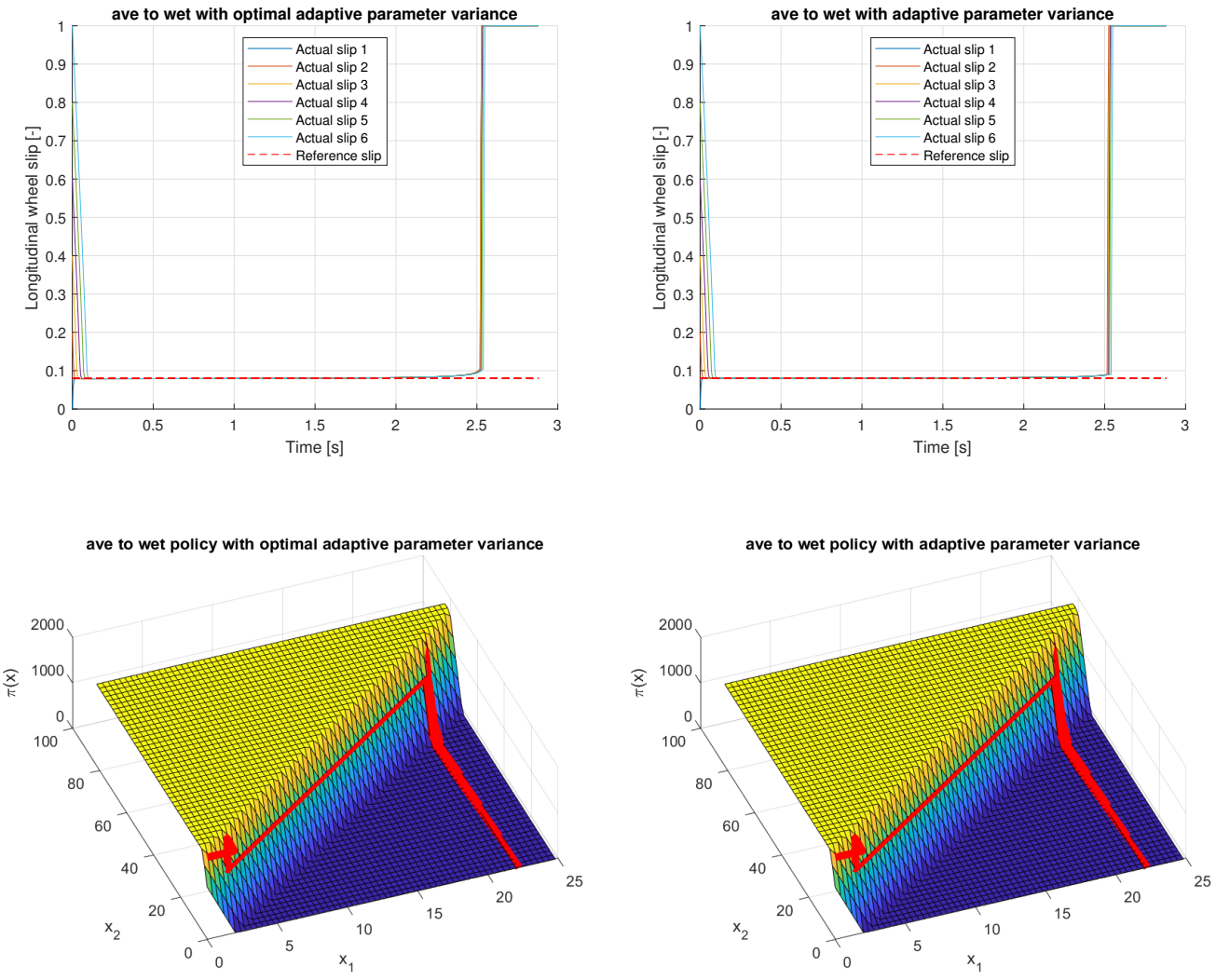


Figure 5.5: Performance comparison for ave to wet

5.3. Adaptive Proportional-Integral (PI) control

A discrete time P-I and P controller for pure wheel slip control on dry and wet asphalt was implemented with the following cases:

1. Fixed κ setpoint, adaptive proportional gain K_p and integral gain K_i
2. Adaptive κ setpoint, proportional gain K_p and integral gain K_i
3. Adaptive κ setpoint, fixed proportional gain K_p and integral gain K_i
4. Adaptive κ setpoint, fixed proportional gain K_p
5. Adaptive κ setpoint and proportional gain K_p

PoWER was used for adaptation in each case. The Simulink model used is shown in Figure 5.6. The ABS system is the same as in the previous case. Multiple wheel slip initial conditions have been taken into account i.e. $\kappa_0 = [0\ 0.2\ 0.4\ 0.6\ 0.8\ 1]$. The initial chassis speed is kept the same i.e. 80 km/h for all wheel slips. The return of the j^{th} episode is calculated using $R_j = \rho_{\text{off}} - d_{x,j}$ where ρ_{off} is the reward offset and $d_{x,j}$ is the average braking distance for all initial conditions for that episode. The reward offset is taken to be 200 to ensure that the return is positive as PoWER requires the episode return to be positive [17]. The number of iterations are taken to be 300 as these were sufficient to show return

and parameter convergence while not taking too much time. After every 10 iterations, a noiseless trial i.e. an episode without random exploration but with parameters updated through importance sampling is conducted to test the true performance. Adaptive parameter variance was used for all adaptations.

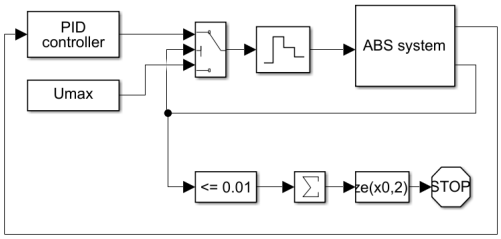
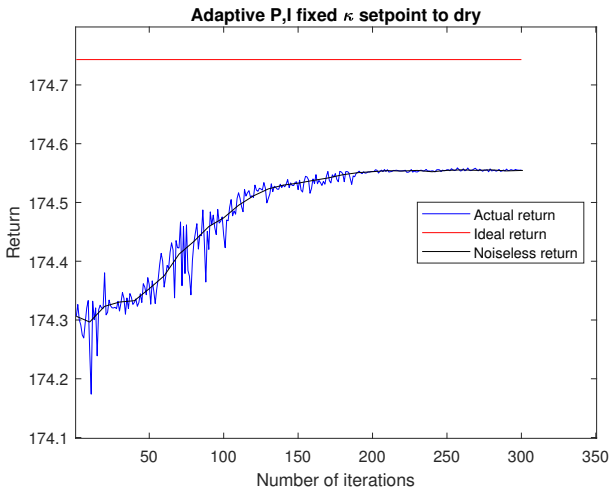
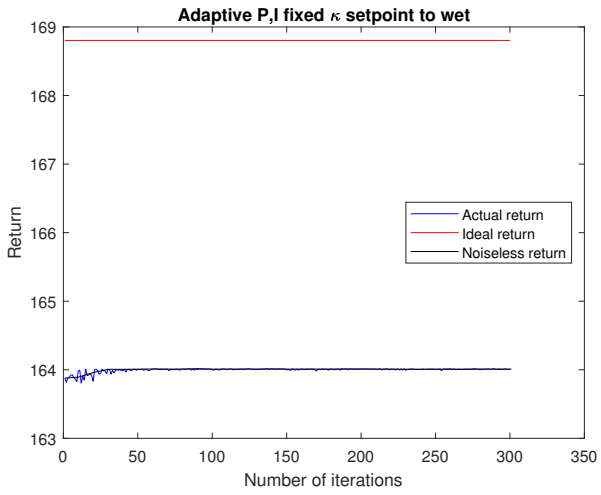


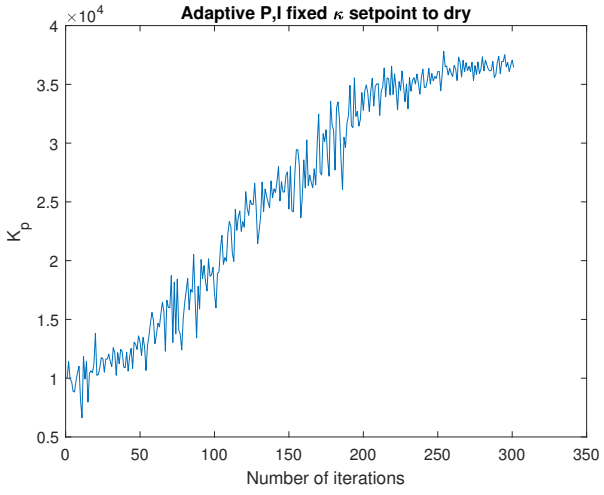
Figure 5.6: The Simulink model

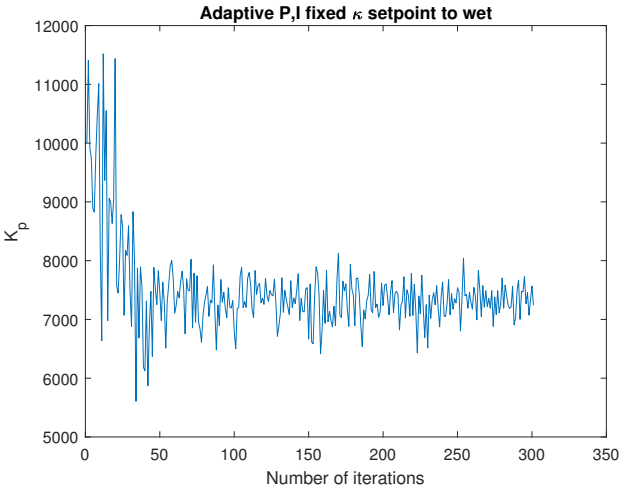
5.3.1. Fixed slip setpoint, adaptive proportional gain and integral gain

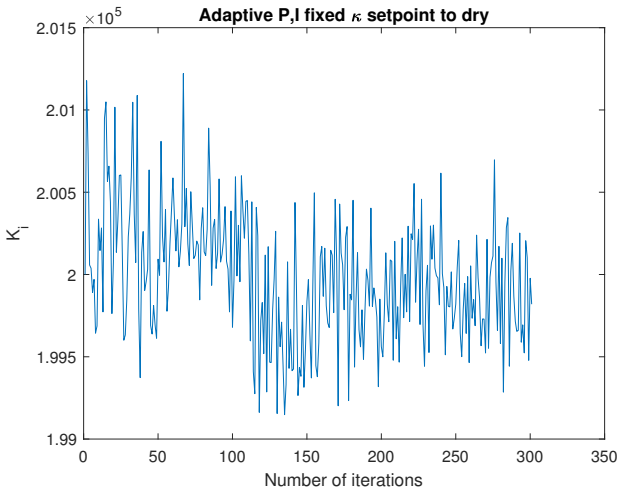
The initial values of the gains and variance were taken as $K_p = 10000$ and $K_i = 200000$, $\sigma^2 = [1000000\ 1000000]^T$ respectively. The reason for using large values is that it gives faster convergence. The slip setpoint was found for each surface from Figure 3.3. The result comparison for dry and wet asphalt is shown in Figure 5.7. The return converges faster for wet asphalt but does not converge to the ideal values for both surfaces. The parameters also show convergence.

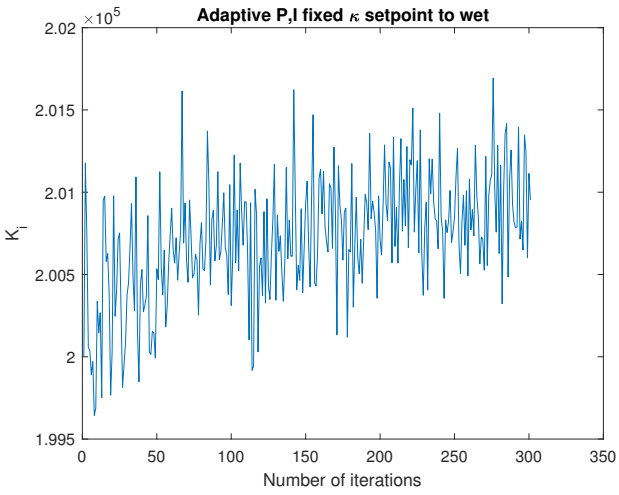




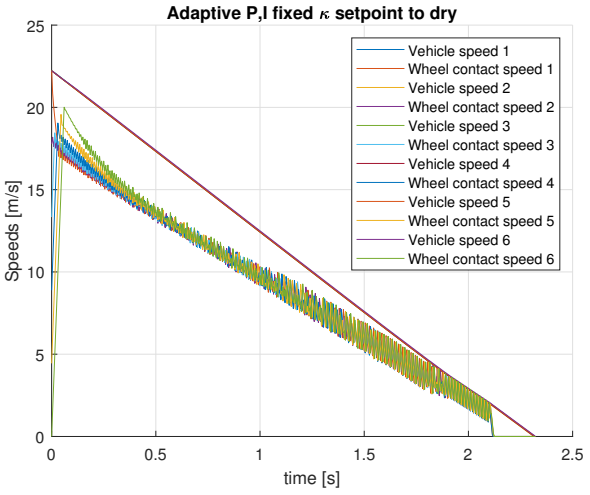


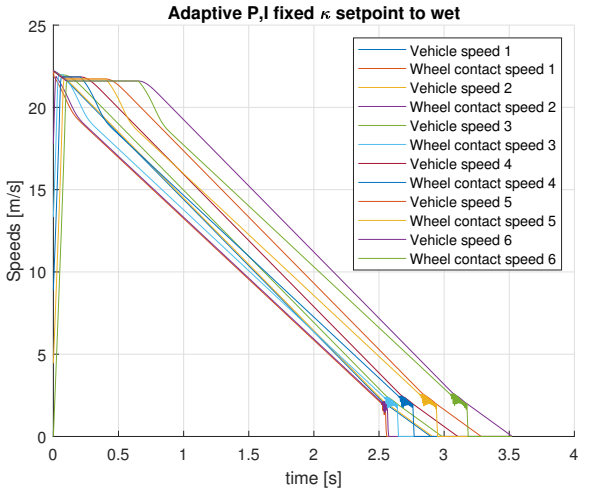


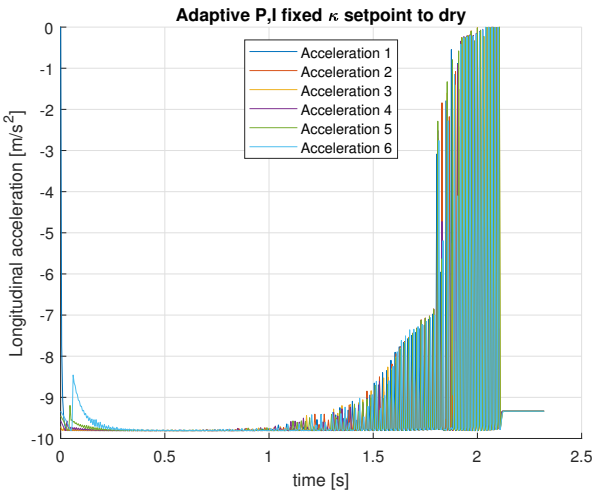


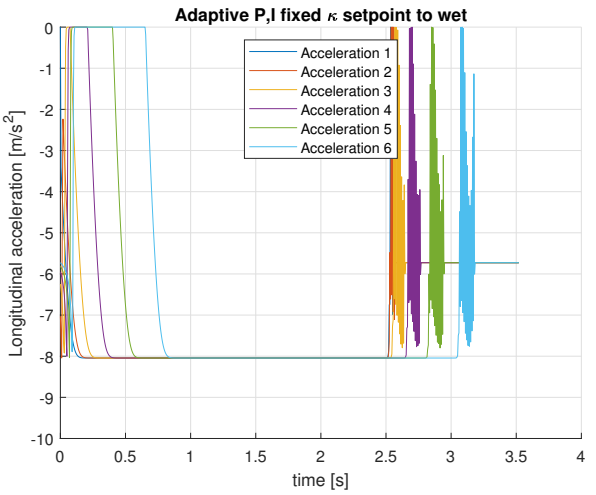


For dry asphalt, the speeds converge to zero in the same time for all the initial conditions but there are a lot of oscillations in the angular wheel velocity, linear deceleration and wheel slip for lower initial slips. These oscillations increase at lower speeds. This behaviour is due to significant chattering in the control input as seen from the graph of braking torque.









In contrast, for wet asphalt, the speeds converge to zero in different times and there are minor oscillations in the wheel angular velocity just before ABS is switched off. The deceleration is mostly constant as the ideal slip value is attained and both oscillate just before ABS is switched off. The ideal slip is attained after longer times for lower wheel slips. The smooth behaviour is due to smooth control input, except before ABS is about to be switched off. This PI controller works better on wet asphalt than on dry asphalt in terms of stability of control input.

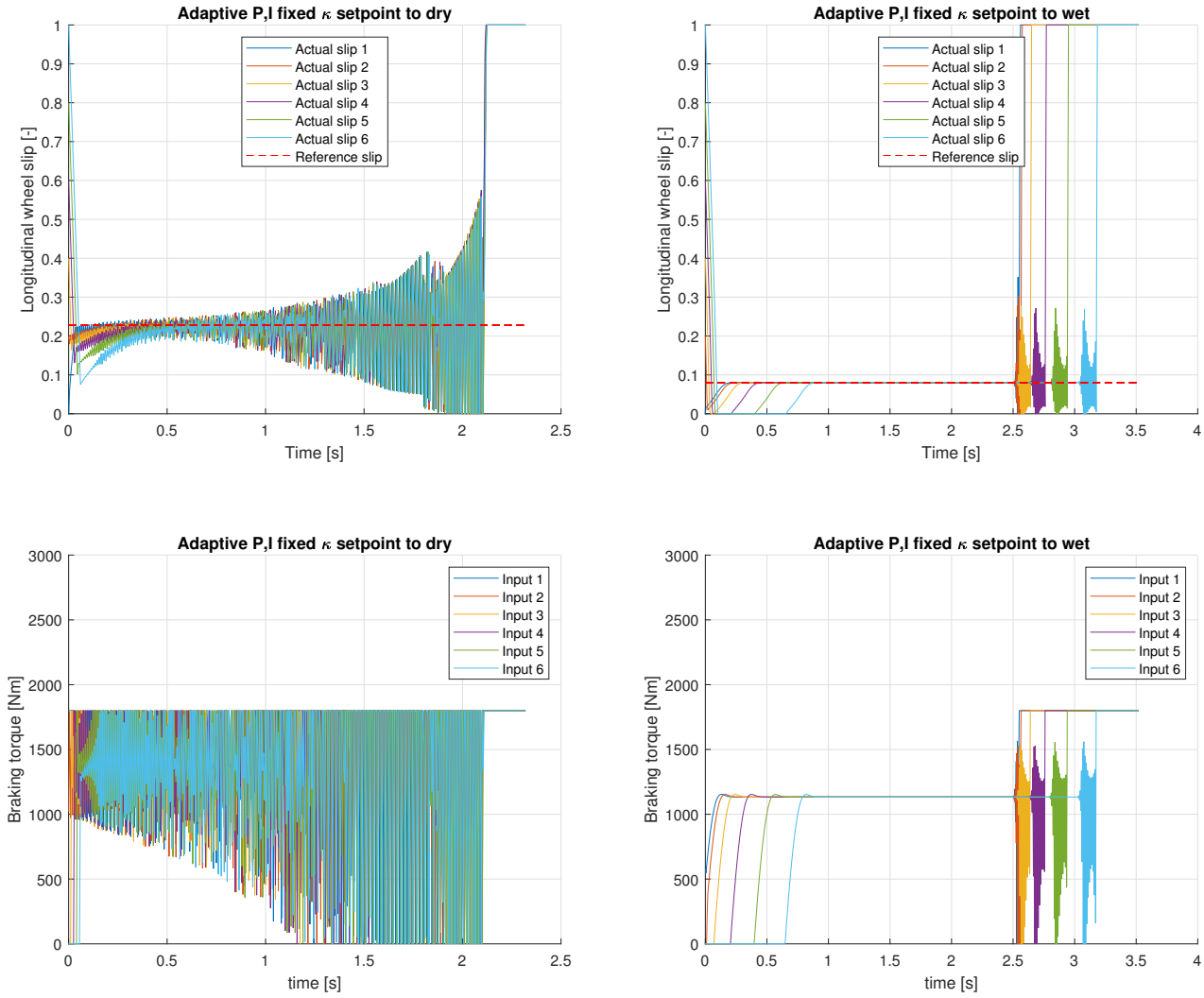


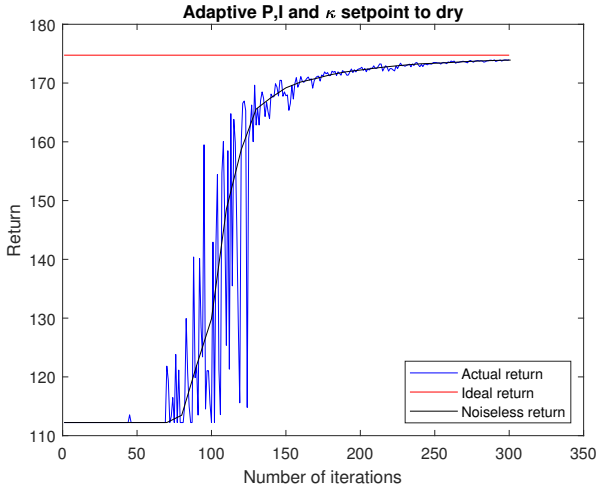
Figure 5.7: Performance comparison on dry and wet asphalt

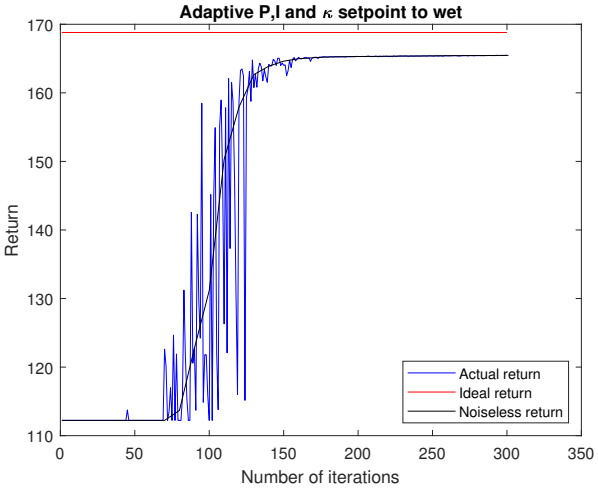
5.3.2. Adaptive slip setpoint, proportional gain and integral gain

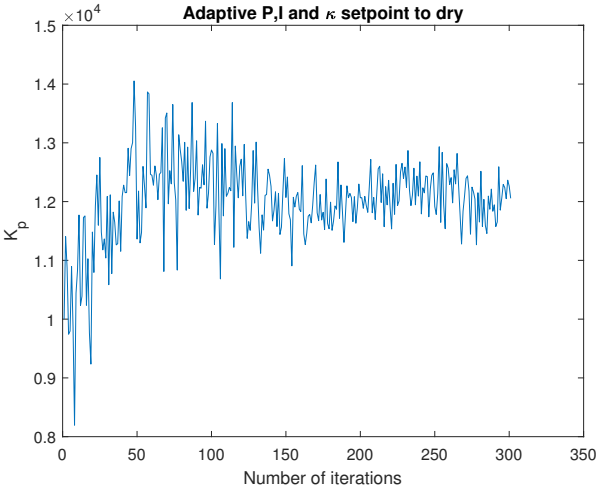
The initial values of the gains, setpoint and variance were taken as $K_p = 10000$, $K_i = 200000$, $\overline{\kappa} = 0$ and $\sigma^2 = [1000000\ 1000000\ 0.0001]^\top$ respectively. The result comparison for dry and wet asphalt is shown in Figure 5.8. The return almost converges to the ideal value for dry asphalt but the same does not happen for wet asphalt. More iterations are needed for both surfaces for convergence. All parameters except K_p for wet asphalt and slip setpoint for dry asphalt show convergence.

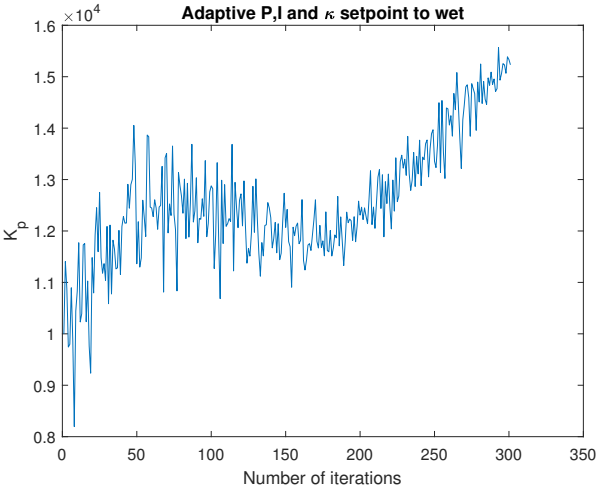
For dry asphalt, the speeds converge to zero in almost the same time for all the initial conditions but there are a some oscillations in the angular wheel velocity and wheel slip for lower initial slips at lower speeds (approx 4 m/s). This behaviour is due to significant chattering in the control input at lower speeds. However, the linear deceleration is mostly constant till 1.8 seconds, which is better than that of the previous PI controller. Oscillatory behaviour is also seen for wet asphalt, in contrast to that of the previous PI controller. The speeds converge to zero in different times and there are larger oscillations in the wheel angular velocity, deceleration and wheel slip for lower initial slips at lower speeds (approx 6

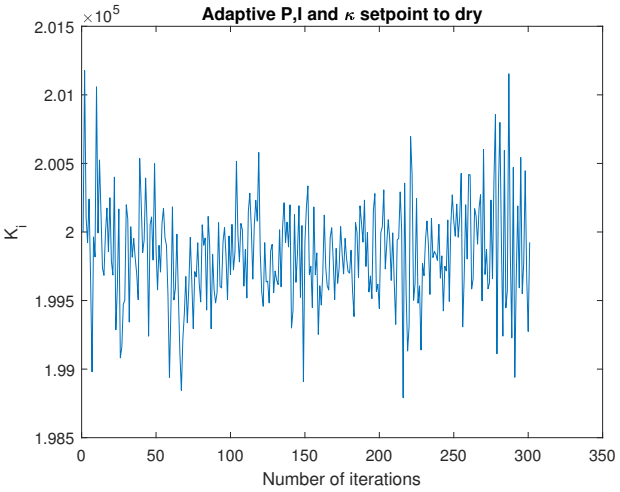
m/s). These oscillations are caused due to chattering in the control input. Compared to the previous PI controller, this PI controller works better on dry asphalt but worse on wet asphalt in terms of stability of control input.

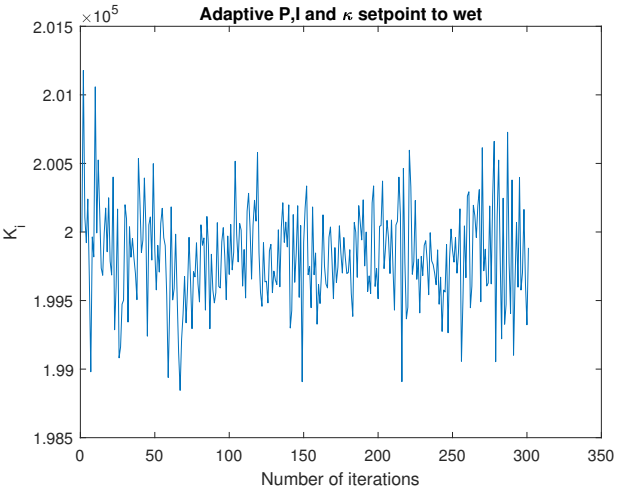


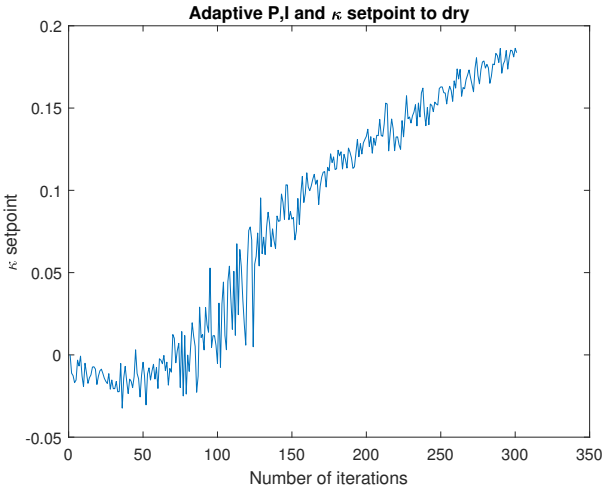


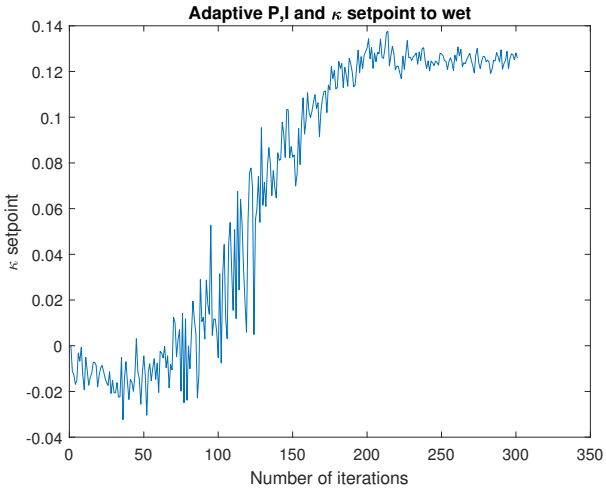


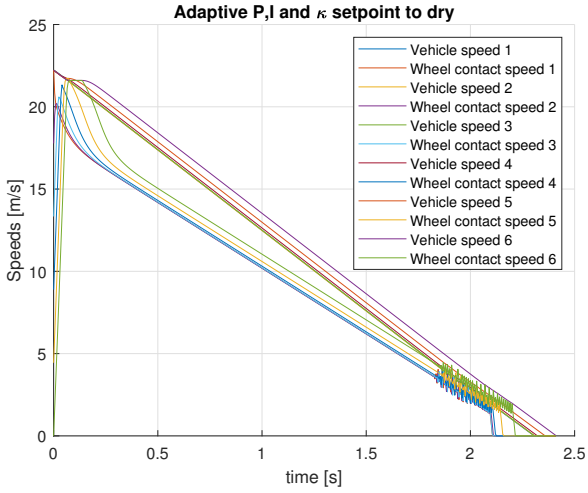


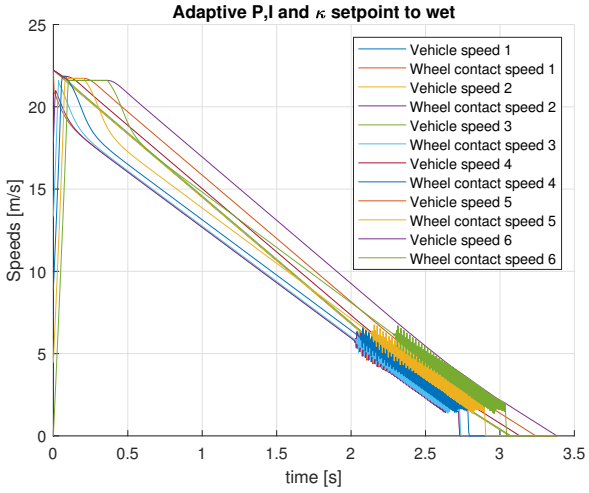


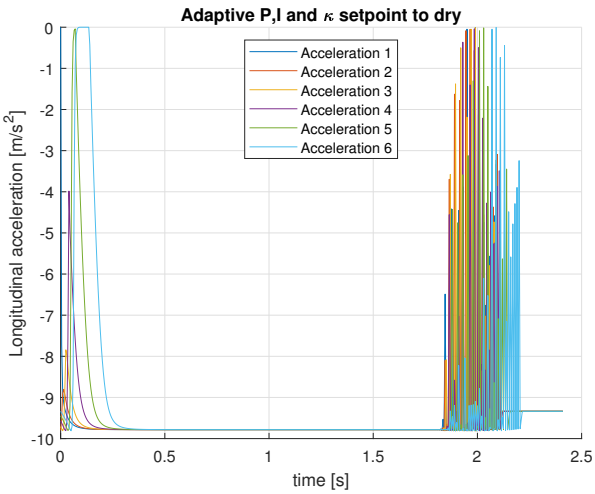


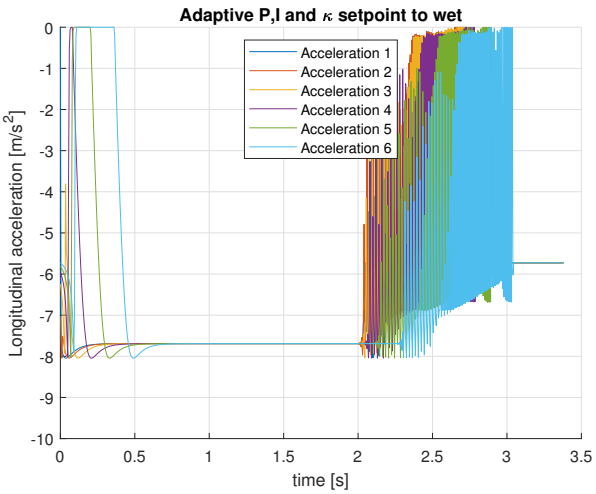

Adaptive P,I and κ setpoint to dry
κ setpoint
Number of iterations


Adaptive P,I and κ setpoint to wet
κ setpoint
Number of iterations


Adaptive P,I and κ setpoint to dry
Speeds [m/s]
time [s]
Vehicle speed 1
Wheel contact speed 1
Vehicle speed 2
Wheel contact speed 2
Vehicle speed 3
Wheel contact speed 3
Vehicle speed 4
Wheel contact speed 4
Vehicle speed 5
Wheel contact speed 5
Vehicle speed 6
Wheel contact speed 6


Adaptive P,I and κ setpoint to wet
Speeds [m/s]
time [s]
Vehicle speed 1
Wheel contact speed 1
Vehicle speed 2
Wheel contact speed 2
Vehicle speed 3
Wheel contact speed 3
Vehicle speed 4
Wheel contact speed 4
Vehicle speed 5
Wheel contact speed 5
Vehicle speed 6
Wheel contact speed 6


Adaptive P,I and κ setpoint to dry
Longitudinal acceleration [m/s2]
time [s]
Acceleration 1
Acceleration 2
Acceleration 3
Acceleration 4
Acceleration 5
Acceleration 6


Adaptive P,I and κ setpoint to wet
Longitudinal acceleration [m/s2]
time [s]
Acceleration 1
Acceleration 2
Acceleration 3
Acceleration 4
Acceleration 5
Acceleration 6

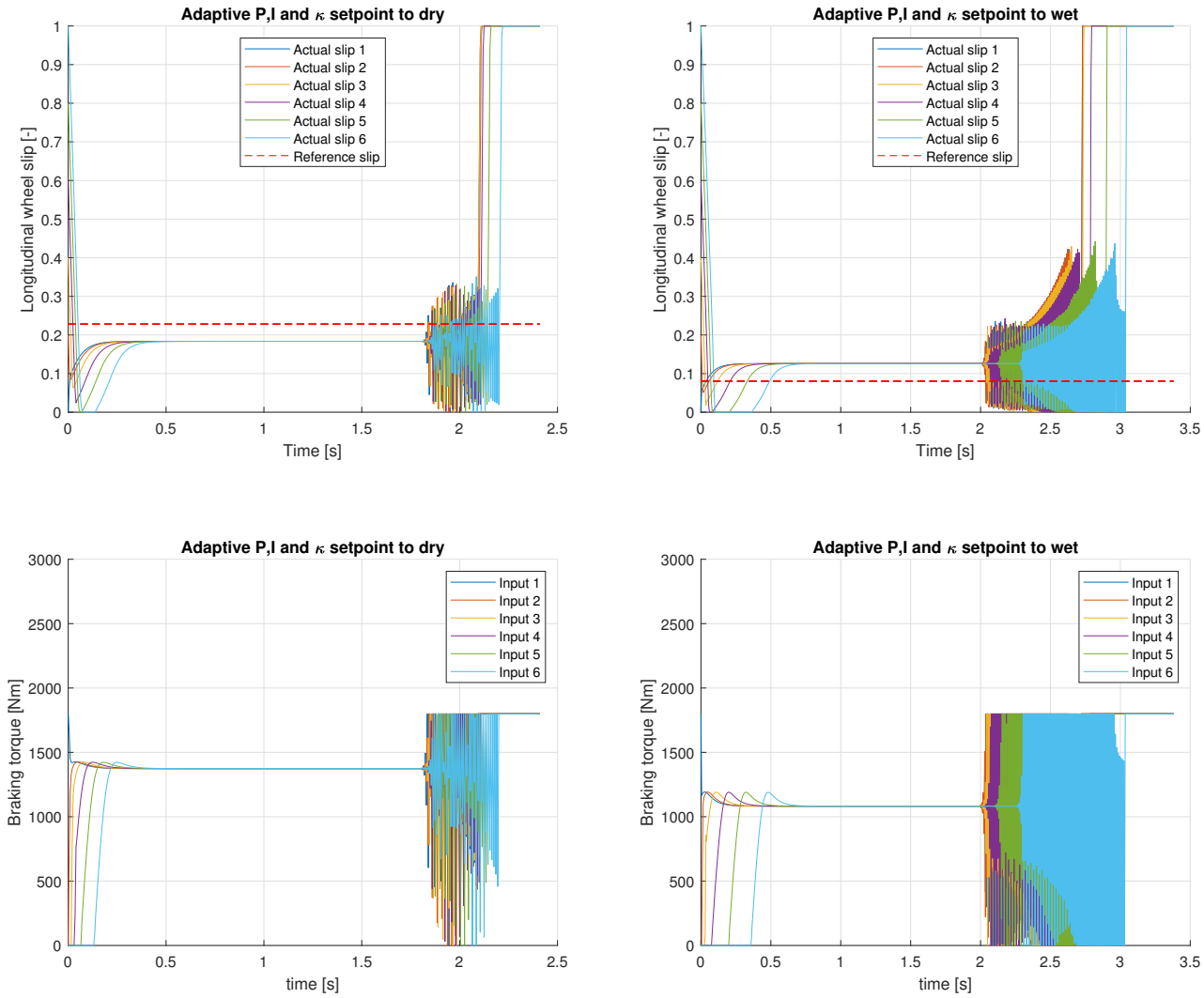
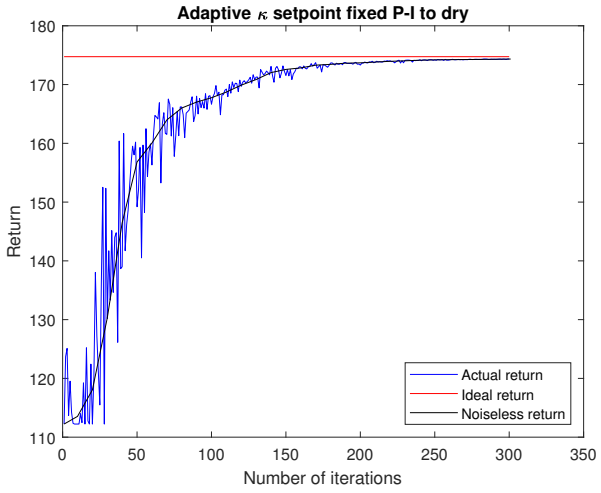


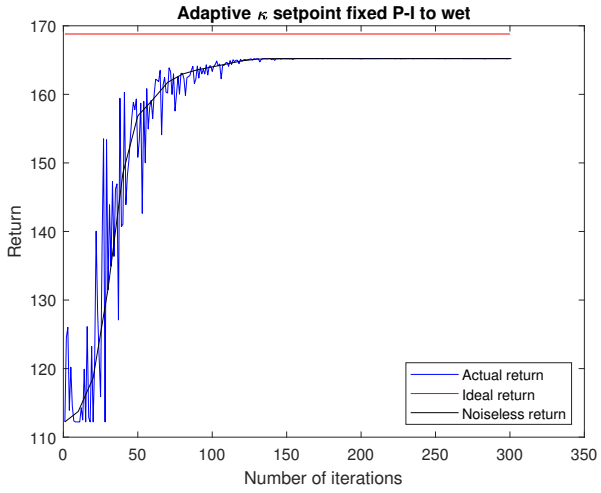
Figure 5.8: Performance comparison on dry and wet asphalt

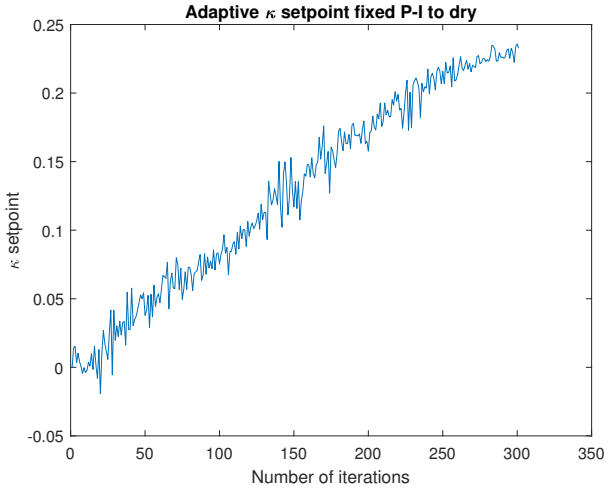
5.3.3. Adaptive slip setpoint, fixed proportional gain and integral gain

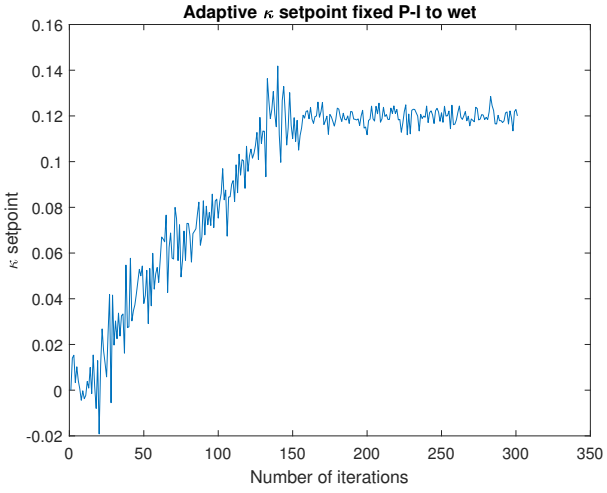
The initial values of the setpoint and variance were taken as $\overline{\kappa} = 0$ and $\sigma^2 = 0.0001$ respectively. The fixed values of the gains were taken as $K_p = 10000$ and $K_i = 200000$. The result comparison for dry and wet asphalt is shown in Figure 5.9. The return converges to the ideal value for dry asphalt but only converges to a sub-optimal value for wet asphalt.

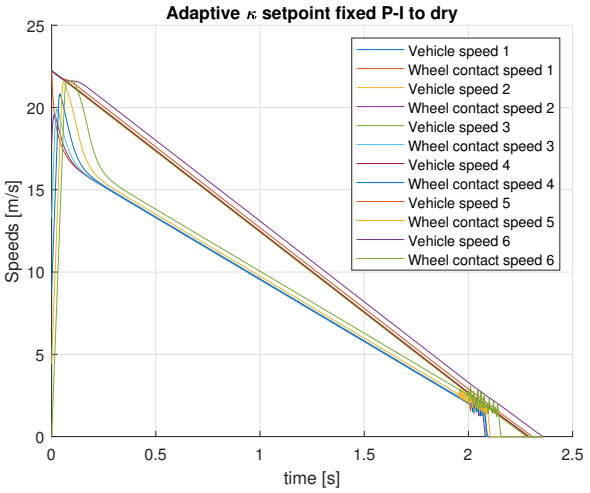
For dry asphalt, the speeds converge to zero in almost the same time for all the initial conditions but there are minor oscillations in the angular wheel velocity, linear deceleration, and wheel slip for lower initial slips just before ABS is switched off. This behaviour is due to minor chattering in the control input at lower speeds (approx 3 m/s). Oscillatory behaviour is also seen for wet asphalt at low speeds, and is much lesser than that of the previous PI controller. The speeds converge to zero in different times and there are larger oscillations in the wheel angular velocity, deceleration and wheel slip for lower initial slips. Chattering in the control input occurs at approx 4 m/s. In terms of stability of control input, this controller works better than the previous 2 PI controllers on dry asphalt, and better than the previous controller but not as good as the first PI controller on wet asphalt.

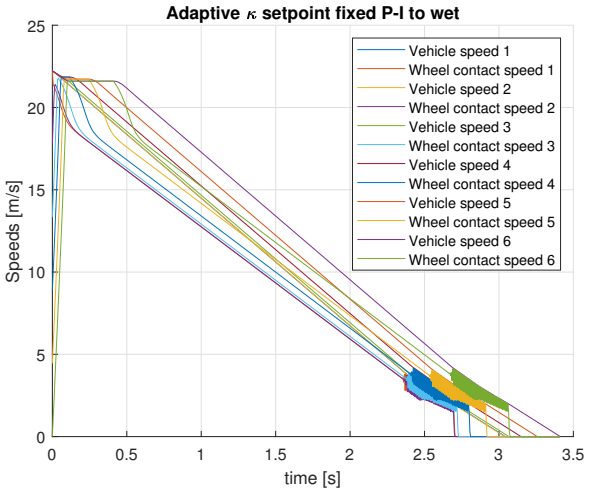

Adaptive κ setpoint fixed P-I to dry
Return
Number of iterations
Actual return
Ideal return
Noiseless return


Adaptive κ setpoint fixed P-I to wet
Return
Number of iterations
Actual return
Ideal return
Noiseless return


Adaptive κ setpoint fixed P-I to dry
κ setpoint
Number of iterations


Adaptive κ setpoint fixed P-I to wet
κ setpoint
Number of iterations


Adaptive κ setpoint fixed P-I to dry
Speeds [m/s]
time [s]
Vehicle speed 1
Wheel contact speed 1
Vehicle speed 2
Wheel contact speed 2
Vehicle speed 3
Wheel contact speed 3
Vehicle speed 4
Wheel contact speed 4
Vehicle speed 5
Wheel contact speed 5
Vehicle speed 6
Wheel contact speed 6


Adaptive κ setpoint fixed P-I to wet
Speeds [m/s]
time [s]
Vehicle speed 1
Wheel contact speed 1
Vehicle speed 2
Wheel contact speed 2
Vehicle speed 3
Wheel contact speed 3
Vehicle speed 4
Wheel contact speed 4
Vehicle speed 5
Wheel contact speed 5
Vehicle speed 6
Wheel contact speed 6

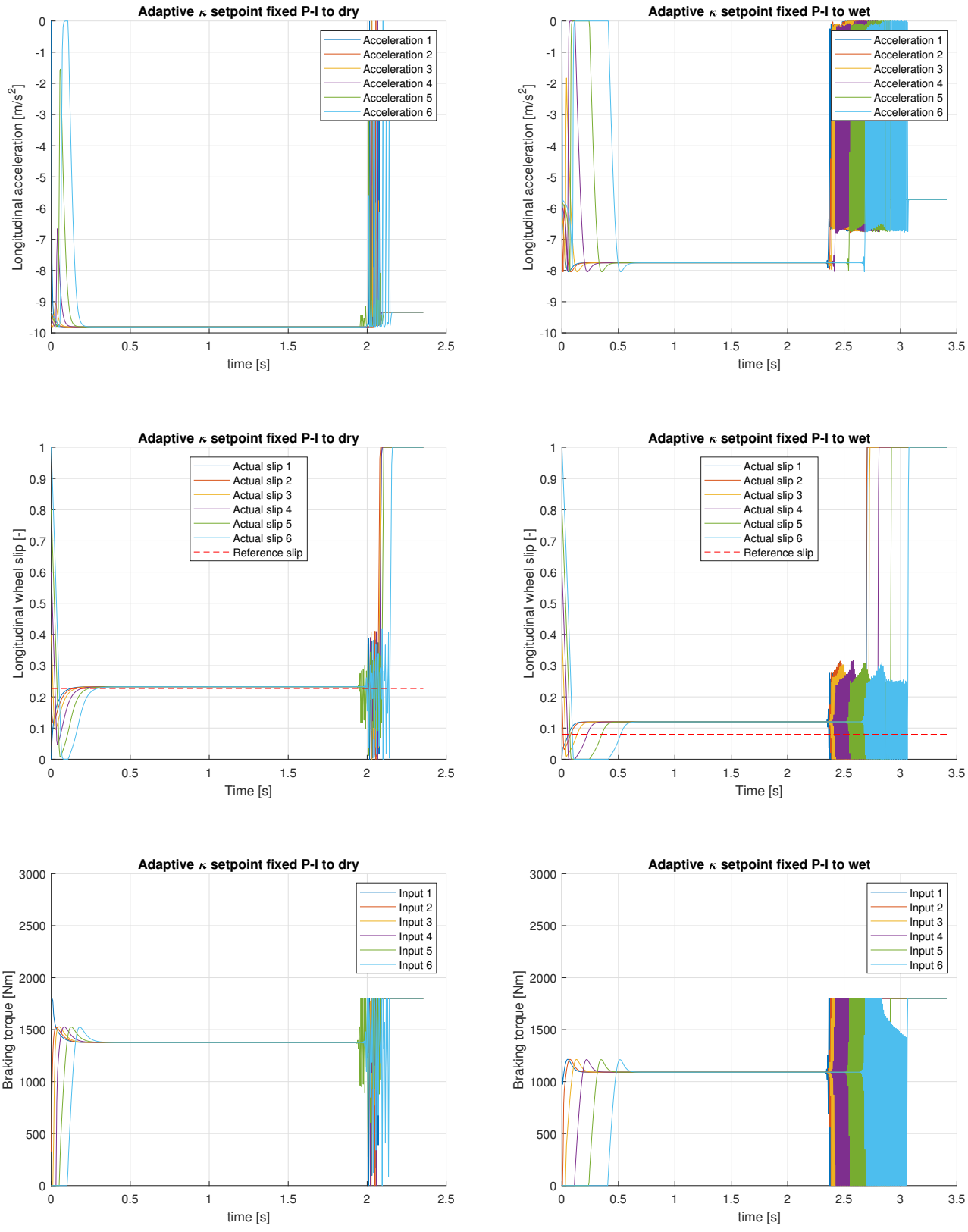
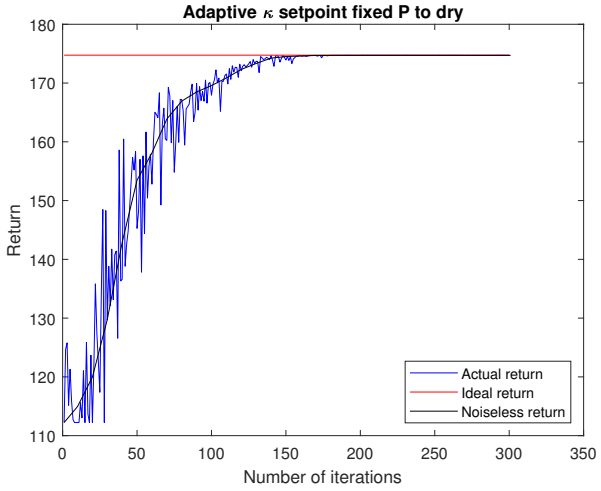


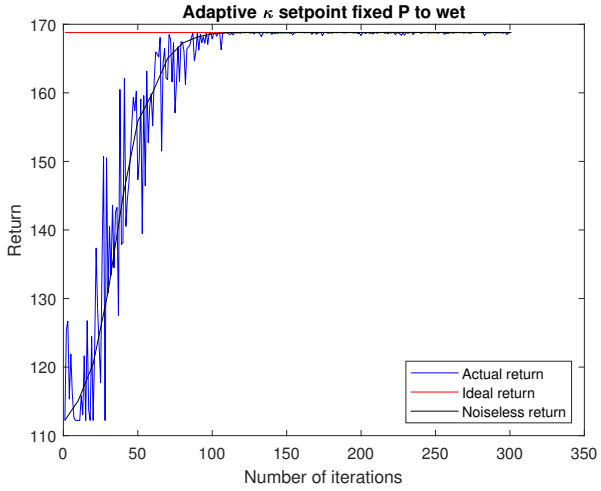
Figure 5.9: Performance comparison on dry and wet asphalt

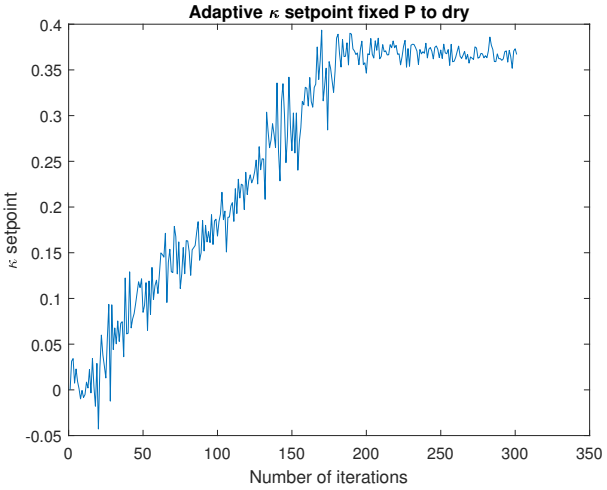
5.3.4. Adaptive slip setpoint, fixed proportional gain

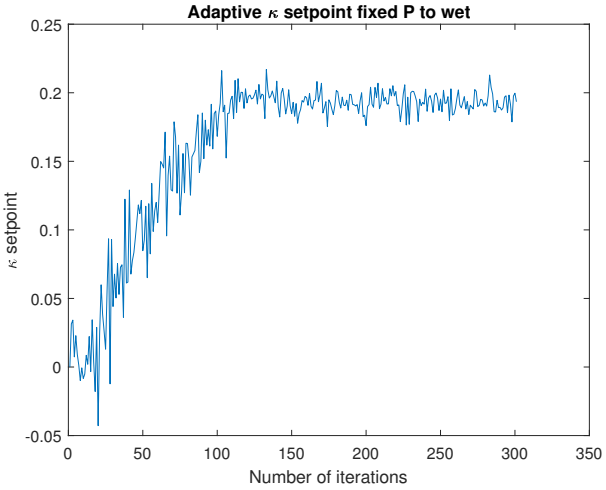
The initial values of the setpoint and variance were taken as $\overline{\kappa} = 0$ and $\sigma^2 = 0.0005$ respectively. The fixed value of the gain was taken as $K_p = 10000$. The result comparison for dry and wet asphalt is shown

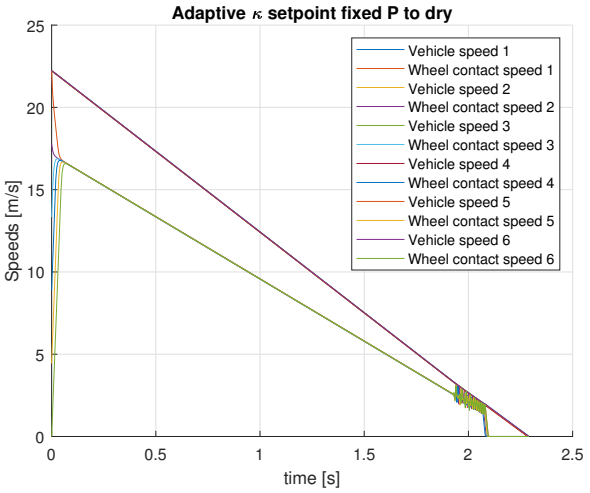
in Figure 5.10. The return converges to the ideal value faster for wet asphalt than for dry asphalt. The slip setpoint converges to higher values than those found by the previous controller since integral gain is absent here.

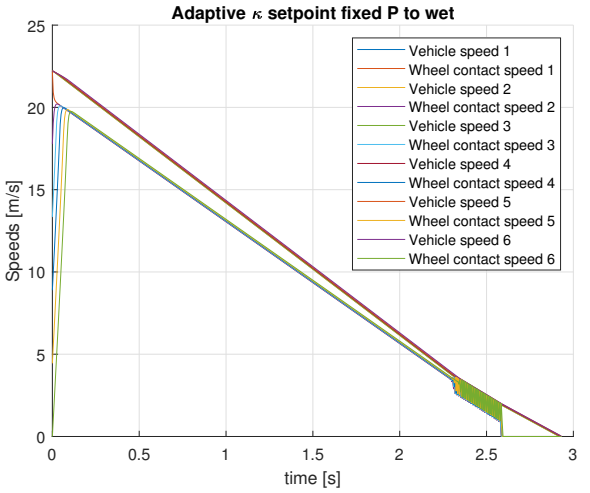












For dry asphalt, the speeds converge to zero in the same time for all the initial conditions with minor oscillations in the angular wheel velocity and and wheel slip for lower initial slips just before ABS is

switched off. These oscillations are higher for linear deceleration since it is proportional to the rate of change of angular speed. This behaviour is due to minor chattering in the control input at very low speeds (approx 2.5 m/s). On wet asphalt, the speeds converge to zero in the same time and there are larger oscillations in the wheel angular velocity, deceleration and wheel slip for lower initial slips at speeds just before and after ABS is switched off, but are much lesser than that of the previous PI controller. Chattering in the control input occurs at approx 3 m/s. In terms of braking distance, this controller performs better than the last 3 PI controllers on both dry and wet asphalt.

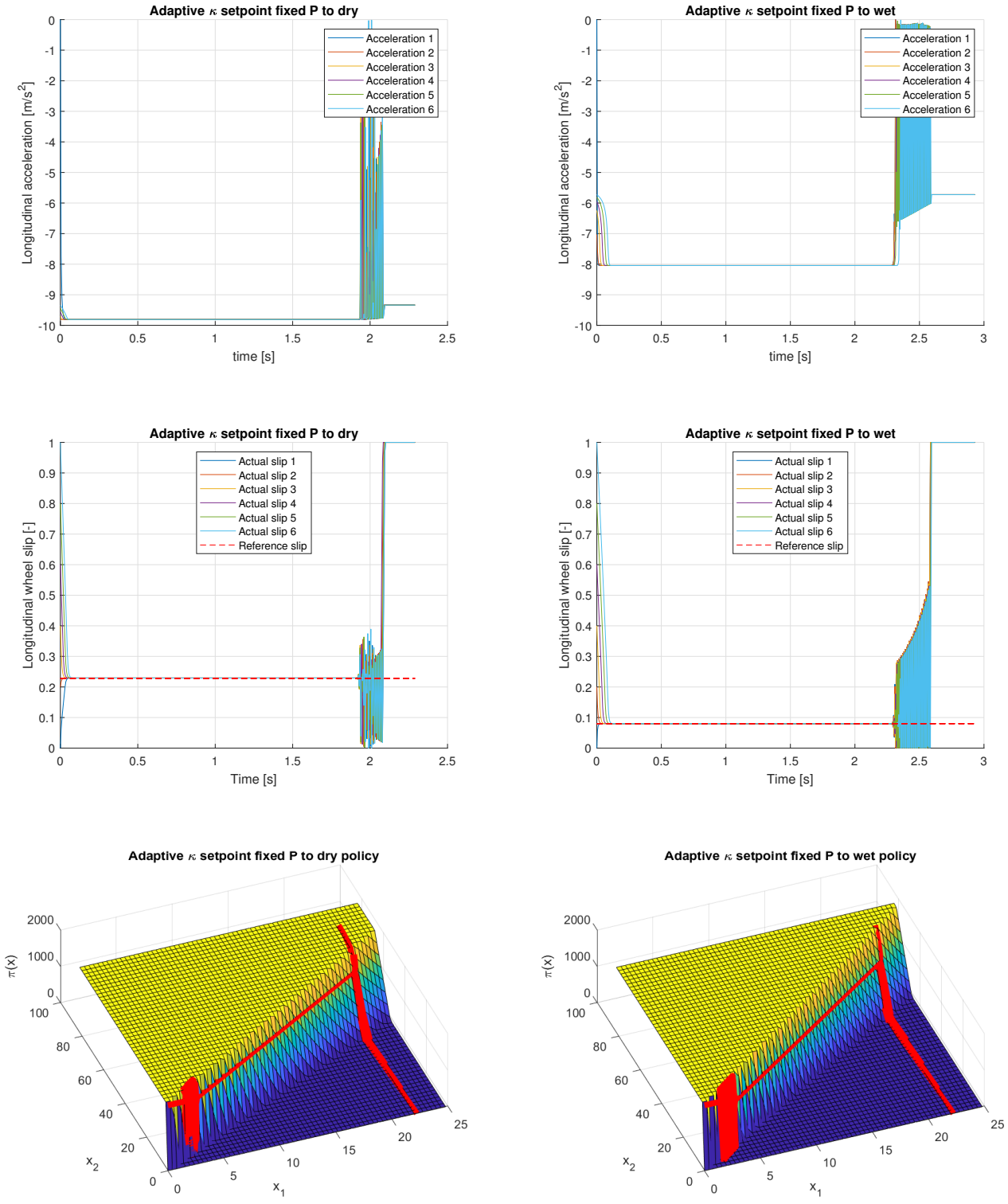
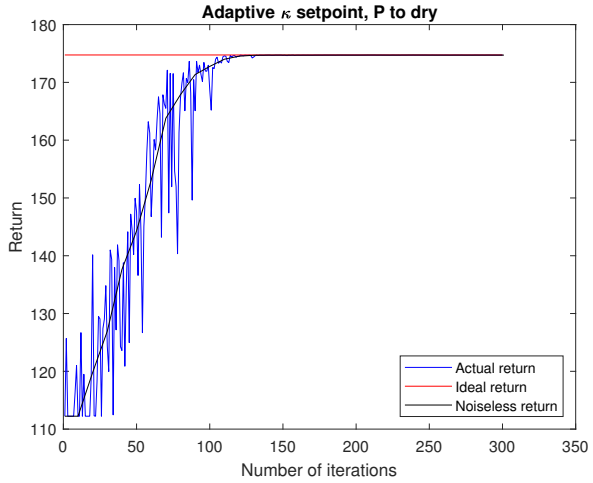
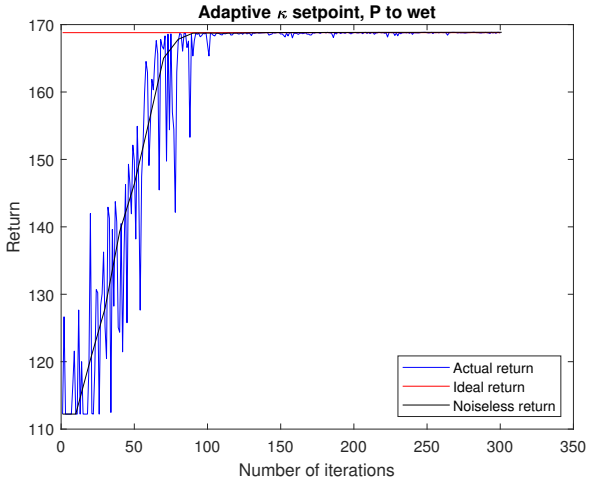


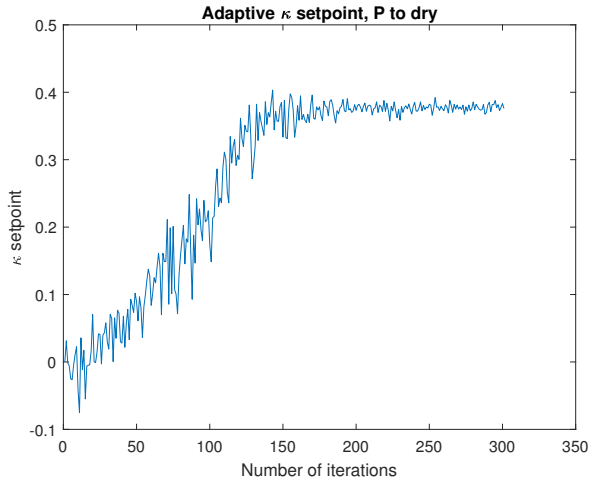
Figure 5.10: Performance comparison on dry and wet asphalt

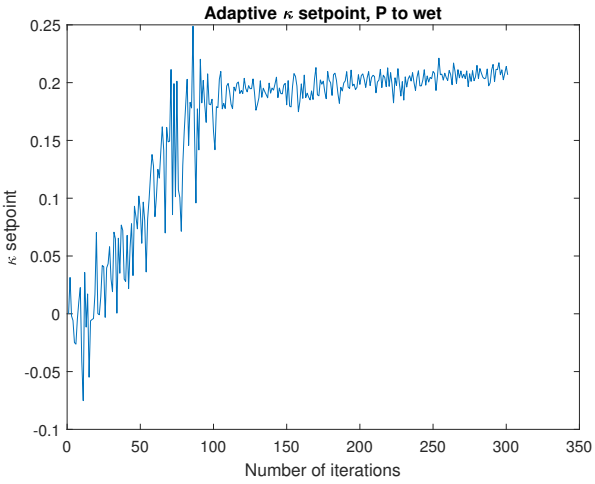
5.3.5. Adaptive slip setpoint and proportional gain

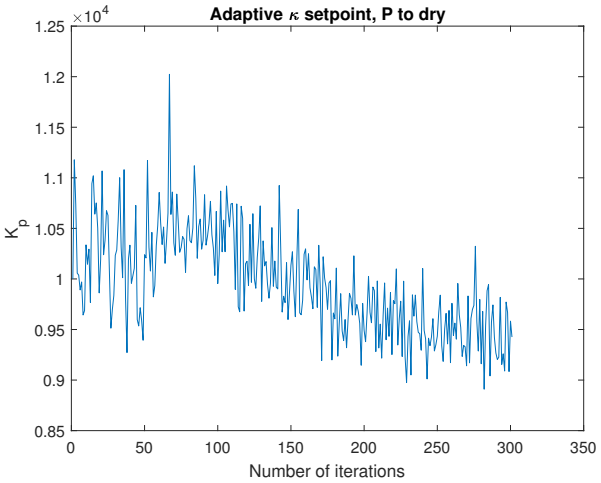
The initial values of the setpoint, proportional gain and variance were taken as $[\overline{\kappa}\ K_p] = [0\ 10000]$ and $\sigma^2 = [0.0005\ 1000000]^\top$ respectively. The result comparison for dry and wet asphalt is shown in Figure 5.11. The return converges to the ideal value faster for wet asphalt than for dry asphalt. The slip setpoint converges to higher values than those found by the previous PI controllers since integral gain is absent. The proportional gains converge to lower values for both surfaces.

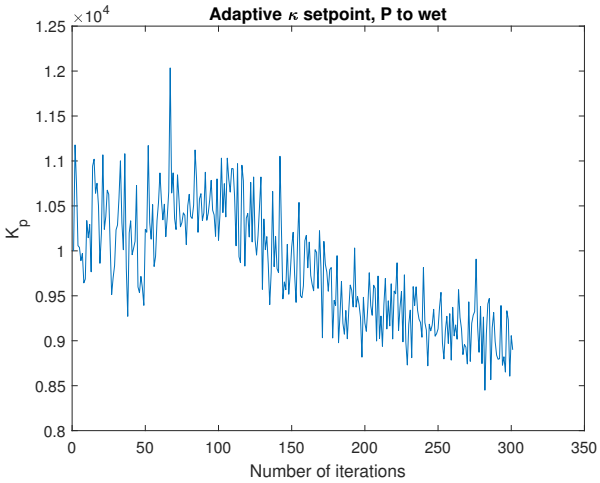


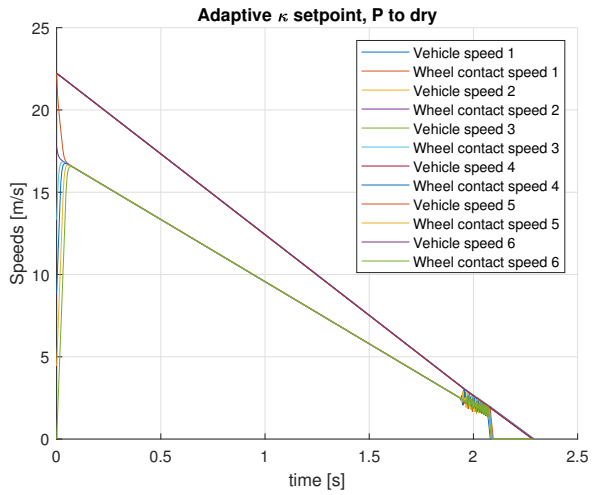


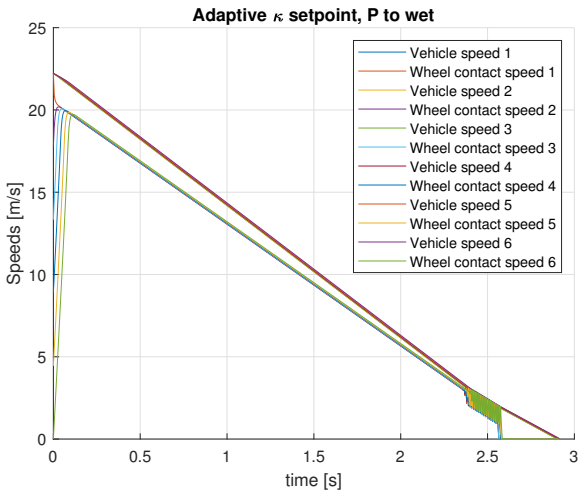


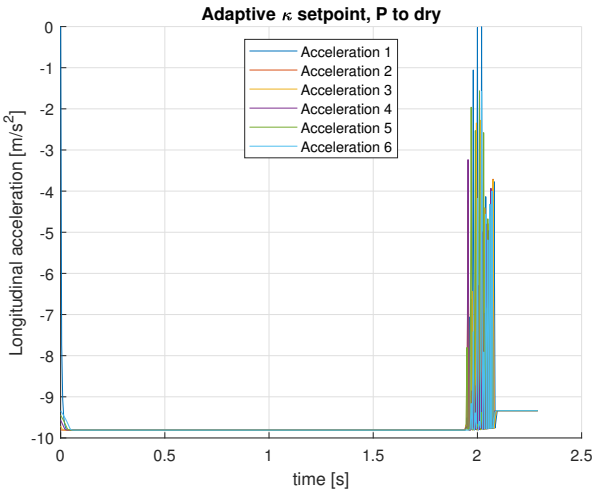


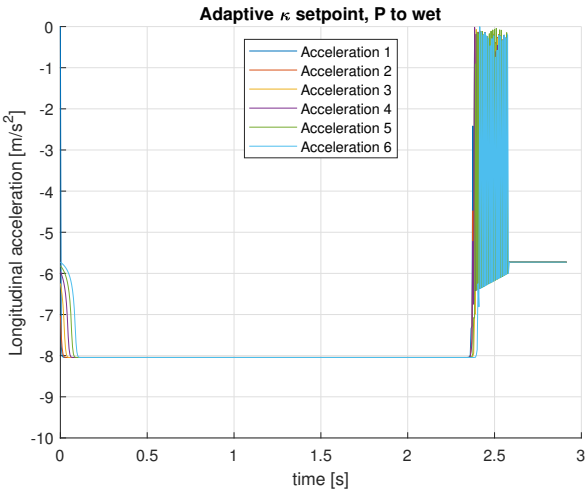


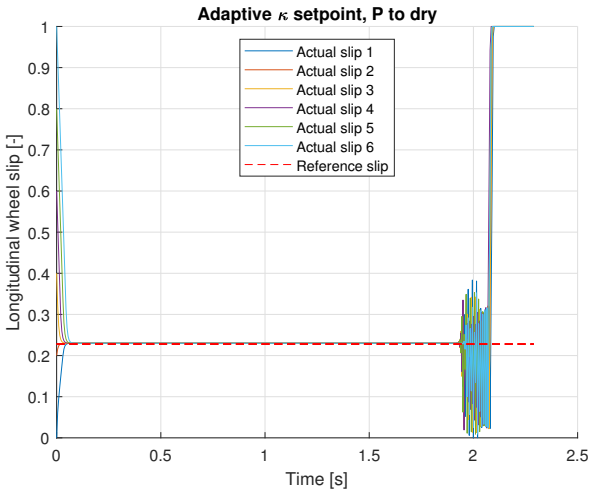


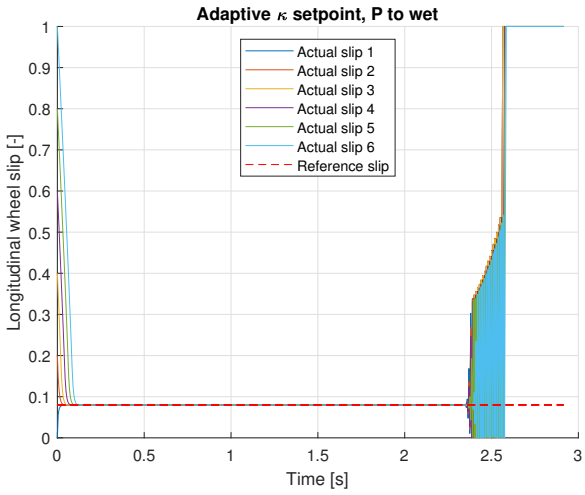












For dry asphalt, the speeds converge to zero in the same time for all the initial conditions with minor oscillations in the angular wheel velocity and and wheel slip for lower initial slips just before ABS is switched off. These oscillations are higher for linear deceleration since it is proportional to the rate of change of angular speed. This behaviour is due to minor chattering in the control input at very low speeds (approx 2.5 m/s). On wet asphalt, the speeds converge to zero in the same time and there are larger oscillations in the wheel angular velocity, deceleration and wheel slip for lower initial slips at speeds just before and after ABS is switched off, similar to those of the previous P controller. Chattering

in the control input occurs at approx 3 m/s. This controller performs similar to the previous P controller on both dry and wet asphalt.

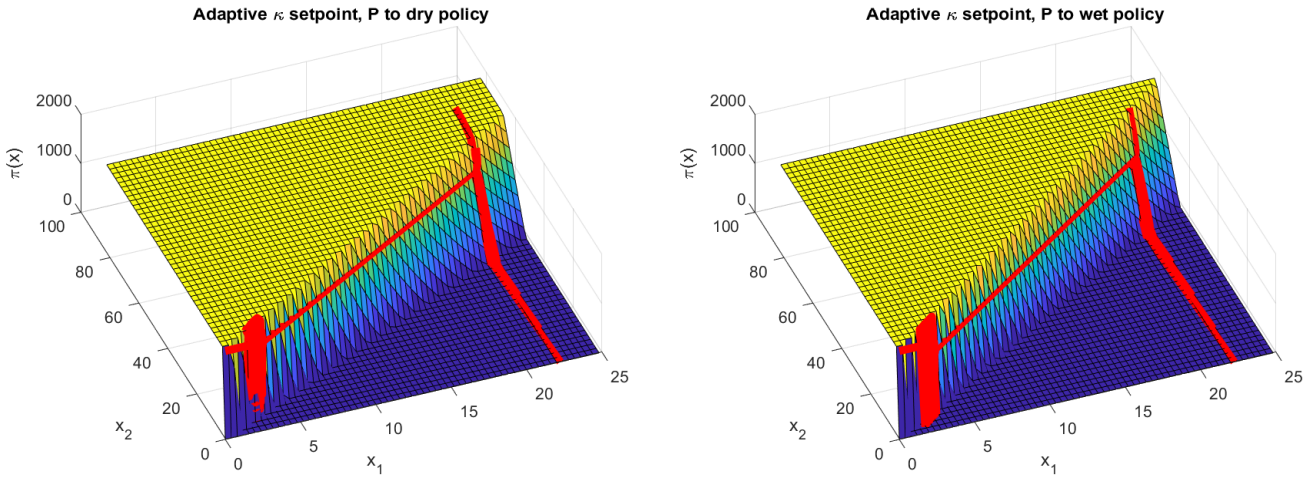


Figure 5.11: Performance comparison on dry and wet asphalt

The braking distances for this P controller on dry and wet asphalt are given in Table 5.3. When compared with the ideal braking distances from Chapter 4 i.e. 25.31 m for dry on dry and 31.04 m for wet on wet, it can be seen that on dry asphalt, all braking distances are marginally higher, and on wet asphalt, the braking distances are lower for 3 out of 6 initial conditions and higher in the other 3 with the maximum increment being 0.37 m. When compared with the distances obtained through adaptation in Table 5.1, it can be seen that the braking distances are marginally higher than those of wet to dry and ave to dry for all except 4 respective initial conditions, and are marginally higher than that of dry to wet and ave to wet for each respective initial condition with the maximum increment being 0.11 m.

Table 5.3: Braking distance [m]

Surface	Slip 1	Slip 2	Slip 3	Slip 4	Slip 5	Slip 6
Dry asphalt	25.34	25.26	25.26	25.26	25.28	25.28
Wet asphalt	30.99	30.97	31.04	31.16	31.29	31.41

The standard deviation in deceleration [m/s^2] for this P controller on dry and wet asphalt are given in Table 5.4. When compared to ideal values from Chapter 4 i.e. 0.4481 for average on dry and 0.4023 for dry on wet, the comfort levels are much worse for all initial conditions on both dry and wet asphalt. When compared with the values obtained through adaptation in Table 5.2, it can be seen that the comfort levels are much worse for all initial conditions on both dry and wet asphalt.

Table 5.4: Standard deviation in acceleration [m/s^2]

Surface	Slip 1	Slip 2	Slip 3	Slip 4	Slip 5	Slip 6
Dry asphalt	1.0551	0.7933	0.7998	0.6245	0.8508	0.699
Wet asphalt	1.562	1.5228	1.488	1.5484	1.5538	1.4935

5.4. Summary

In this chapter, the results of adaptation of the piecewise linear approximated policy using optimal and robust initial parameter variance methods was shown and discussed for four adaptations. The adaptations of average policy to dry and wet asphalt perform well for both the variance methods. To compare the performance, the results for an adaptive PI controller for 5 cases of adaptations were also shown. The P controllers with adaptive slip setpoint and fixed proportional gain, and adaptive slip setpoint and proportional gain perform better than all PI controllers. In comparison to the piecewise linear policy adaptations, the P controllers perform slightly lower in terms of braking distance but much worse in terms of comfort of the passengers.

6

Conclusions and Future Work

This chapter presents the conclusions derived from the performed simulations and methods to improve the current implementation. Other applications of the controller are also discussed.

6.1. Conclusions

The following conclusions can be made, starting from Chapter 4:

1. The piecewise linear approximated policies perform better than policies obtained through non-linear interpolation on both dry asphalt and wet asphalt in terms of safety and comfort. The braking distances are almost equal for dry asphalt but are lesser for wet asphalt, which is more critical. Comfort is significantly i.e. about 2.5 times higher on both surfaces.

2. Among linear policies, the average policy performs reasonably well on both surfaces. For the speed for which it is derived i.e. 80 km/h, its performance in terms of braking distance is behind the best by 5.58% on average while the worst is behind by 18.76% on average. The same policy can also be used at a lower speed without much drop in performance e.g 60 km/h with 1% drop in performance.

3. Adaptation of average policy to both dry and wet asphalt is faster than that of wet to dry and dry to wet respectively for both optimal and robust initial parameter variance methods. This is expected, since the parameters of the average policy are between the parameters of dry and wet policies.

4. In [17], PoWER converges for the underactuated swing-up task [28] in about 150 iterations. Compared to this, the adaptations wet to dry, ave to dry, and dry to wet take 100-150 iterations, and ave to wet with optimal performance takes 100 iterations.

5. The convergence speed and final return depend largely on the choice of exploration and parameter variance. Optimal initial variance leads to higher returns but may be slow e.g ave to wet. Robust initial variance leads to faster or slower convergence depending on the adaptation e.g. faster for ave to wet but slower for dry to wet. A trade-off has to be made between speed of convergence and the final return.

6. Ave to dry performs equally well as the ideal policy i.e. dry on dry in terms of braking distance for all initial conditions and and marginally lower in terms of comfort than the ideal i.e. average on dry for only one initial condition, for both optimal and robust variance methods. Ave to wet performs better than the ideal i.e. wet on wet in terms of braking distance for 3 out of 6 initial conditions but worse in terms of comfort for all initial conditions, for both optimal and robust variance methods. Since safety is the the primary concern and the drop in comfort level is not too much, the average policy is a better choice as it is good on both dry and wet asphalt.

7. Ave to dry and ave to wet give the same braking distances for optimal and robust variance methods for 5 out of 6 and 3 out of 6 initial conditions respectively. in terms of comfort, the optimal variance method performs slightly better than the robust variance method for ave to dry and ave to wet for all initial conditions.

8. Among adaptive PI controllers, the third controller, with adaptive slip setpoint and fixed gains performs best on dry asphalt in terms of stability of control input. The first controller, with fixed slip setpoint and adaptive gains performs best on wet asphalt, but the performance is not good as that of the model based adaptive controller for 5 out of 6 initial conditions. For all PI controllers and P controllers, gain scheduling is needed to prevent chattering at low speeds [29].
9. A simple adaptive proportional controller with adaptive slip setpoint and adaptive or fixed gain performs the better than all PI controllers on both dry and wet asphalt for all initial conditions and has minor control input chattering at low speed. Its performance can be compared to the model based RL controller only in terms of braking distance, since comfort performance is much worse.
10. In terms of stability of the control input, the adaptations of the piecewise linear policy perform much better than all the PI controllers, for which gain scheduling has to be used at low speeds to prevent chattering. No such chattering is seen for the adaptations.
11. The piecewise linear policy is found to be similar to that of proportional control with gain scheduling. This explains the lack in performance of the adaptive proportional controller without gain scheduling.

6.2. Future Work

The current implementation of the adaptive controller can be improved by nullifying the current assumptions made for the purpose of simplicity. Also, there are other subsystems of a vehicle for which this controller can be implemented. It can also be implemented for robots.

6.2.1. ABS (Anti-lock Braking System)

1. The road can assumed to be inclined instead of flat.
2. Suspension dynamics can be taken into account.
3. Pitch and roll dynamics can be taken into account by using the half car or full car model instead of the quarter car model on a turn or banked road instead of a straight line. This will lead to the radius of the wheel being dynamic.
4. Transient tire behaviour can be taken into account.
5. Delay due to actuator dynamics can be taken into account.
6. One or two wheels can be assumed to roll on a different surface than the other three or two wheels respectively.
7. A lower sampling period can be chosen e.g. $T_s = 0.001$ s.

6.2.2. Lateral and vertical stability control of a vehicle

1. Similar to controlling the linear chassis velocity and wheel angular velocity for longitudinal stability, this controller can also be implemented for controlling the lateral velocity, yaw rate, and body roll angle for lateral stability during cornering. [30] implements neural network based RL for active roll control of a heavy vehicle.
2. It can also be implemented in active or semi-active suspensions for continuously controlling the sprung mass acceleration and suspension travel, and improving the road holding capacity on a straight road or while turning. [31] uses Continuous Action Reinforcement Learning Automata (CARLA) for control of a semi-active suspension.

6.2.3. Motion control of a robot

The controller can be used for improvement of motion control of a robot such as the Jackal [32] which uses a skid-steering mechanism [33] . Currently, it has pre-defined discrete levels of speed. After a speed is chosen, the voltage to the DC motors in the wheels is controlled automatically. Due to switching between discrete levels of speed, the motion is jerky at times and can be improved by using this controller to make it continuous through continuous voltage control.

References

[1] S. Baldi, *Adaptive and predictive control: Lecture notes,* (2017).

[2] S. G. Anavatti, F. Santoso, and M. A. Garratt, *Progress in adaptive control systems: past, present, and future,* in *2015 International Conference on Advanced Mechatronics, Intelligent Manufacture, and Industrial Automation (ICAMIMIA)* (2015) pp. 1–8.

[3] H. P. Whitaker, J. Yamron, and A. Kezer, *Design of model-reference adaptive control systems for aircraft* (Massachusetts Institute of Technology, Instrumentation Laboratory, 1958).

[4] R. E. Kalman, *Design of self-optimizing control system,* Trans. ASME **80**, 468 (1958).

[5] R. Bellman, *The theory of dynamic programming,* Bulletin of the American Mathematical Society **60**, 503 (1954).

[6] K. Narendra, Y.-H. Lin, and L. Valavani, *Stable adaptive controller design, part ii: Proof of stability,* IEEE Transactions on Automatic Control **25**, 440 (1980).

[7] M. Radac, R. Precup, and R. Roman, *Anti-lock braking systems data-driven control using q-learning,* in *2017 IEEE 26th International Symposium on Industrial Electronics (ISIE)* (2017) pp. 418–423.

[8] T. Sardarmehni and A. Heydari, *Optimal switching in anti-lock brake systems of ground vehicles based on approximate dynamic programming,* in *ASME 2015 Dynamic Systems and Control Conference* (American Society of Mechanical Engineers, 2015) pp. V003T50A010–V003T50A010.

[9] J. Kubalík, E. Alibekov, and R. Babuška, *Optimal control via reinforcement learning with symbolic policy approximation,* IFAC-PapersOnLine **50**, 4162 (2017), 20th IFAC World Congress.

[10] G. F. Mauer, *A fuzzy logic controller for an abs braking system,* IEEE Transactions on Fuzzy Systems **3**, 381 (1995).

[11] W. P. R. J. Kamalapurkar, R. and W. Dixon, *Reinforcement Learning for Optimal Feedback Control: A Lyapunov-Based Approach* (Springer, 2018).

[12] L. Buşoniu, D. Ernst, B. De Schutter, and R. Babuška, *Approximate dynamic programming with a fuzzy parameterization,* Automatica **46**, 804 (2010).

[13] I. Grondman, L. Busoniu, G. A. D. Lopes, and R. Babuska, *A survey of actor-critic reinforcement learning: Standard and natural policy gradients,* Trans. Sys. Man Cyber Part C **42**, 1291 (2012).

[14] S.-I. Amari and S. C. Douglas, *Why natural gradient?* in *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE international conference on*, Vol. 2 (IEEE, 1998) pp. 1213–1216.

[15] S. M. Kakade, *A natural policy gradient,* in *Advances in neural information processing systems* (2002) pp. 1531–1538.

[16] P. Dayan and G. E. Hinton, *Using expectation-maximization for reinforcement learning,* Neural Computation **9**, 271 (1997).

[17] J. Kober and J. R. Peters, *Policy search for motor primitives in robotics,* in *Advances in neural information processing systems* (2009) pp. 849–856.

[18] R. J. Williams, *Simple statistical gradient-following algorithms for connectionist reinforcement learning,* in *Reinforcement Learning* (Springer, 1992) pp. 5–32.

[19] F. Stulp and O. Sigaud, *Robot skill learning: From reinforcement learning to evolution strategies,* Paladyn, Journal of Behavioral Robotics **4**, 49 (2013).

[20] J. Peters and S. Schaal, *Reinforcement learning of motor skills with policy gradients,* Neural networks **21**, 682 (2008).

[21] T. Rückstieß, M. Felder, and J. Schmidhuber, *State-dependent exploration for policy gradient methods,* in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (Springer, 2008) pp. 234–249.

[22] R. S. Sutton, A. G. Barto, F. Bach, *et al.*, *Reinforcement learning: An introduction* (MIT press, 1998).

[23] H. Pacejka, *Tire and vehicle dynamics* (Elsevier, 2005).

[24] Mathworks, *https://nl.mathworks.com/help/physmod/sdl/ref/tireroadinteractionmagicformula.html,* .

[25] S. M. Savaresi and M. Tanelli, *Active braking control systems design for vehicles* (Springer Science & Business Media, 2010).

[26] S. Savaresi, M. Tanelli, C. Cantoni, D. Charalambakis, F. Previdi, S. Bittanti, *et al.*, *Slip-deceleration control in anti-lock braking systems,* in *Proc. 16th IFAC world congress, Prague, Czech Republic* (2005).

[27] E. Ono, K. Asano, M. Sugai, S. Ito, M. Yamamoto, M. Sawada, and Y. Yasui, *Estimation of automotive tire force characteristics using wheel velocity,* Control engineering practice **11**, 1361 (2003).

[28] C. G. Atkeson, *Using local trajectory optimizers to speed up global optimization in dynamic programming,* in *Advances in neural information processing systems* (1994) pp. 663–670.

[29] T. A. Johansen, I. Petersen, J. Kalkkuhl, and J. Ludemann, *Gain-scheduled wheel slip control in automotive brake systems,* IEEE Transactions on Control Systems Technology **11**, 799 (2003).

[30] M. J. L. Boada, B. L. Boada, A. Gauchia Babe, J. A. Calvo Ramos, and V. D. Lopez, *Active roll control using reinforcement learning for a single unit heavy vehicle,* International Journal of Heavy Vehicle Systems **16**, 412 (2009).

[31] M. N. Howell, G. P. Frost, T. J. Gordon, Q. H. Wu, *et al.*, *Continuous action reinforcement learning applied to vehicle suspension control,* Mechatronics **7**, 263 (1997).

[32] C. robotics, *https://www.clearpathrobotics.com/jackal-small-unmanned-ground-vehicle/,* .

[33] K. Kozłowski and D. Pazderski, *Modeling and control of a 4-wheel skid-steering mobile robot,* International journal of applied mathematics and computer science **14**, 477 (2004).