Predictive Control of Electric Motors Drives for Unmanned Off-Road Wheeled Vehicles

By

Mostafa Ahmed Ismail Mohammed

Thesis submitted to the Faculty of Graduate and Postdoctoral Studies in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in Mechanical Engineering

Ottawa-Carleton Institute for Mechanical and Aerospace Engineering
Faculty of Engineering
University of Ottawa

March 2013

Abstract

Starting a few decades ago, the unmanned wheeled vehicle research has drawn lately more attention, especially for off-road environment. As the demand to use electric vehicles increased, the need to conceptualize the use of electrically driven vehicles in autonomous operations became a target. That is because in addition to the fact that they are more environmentally friendly, they are also easier to control. This also gives another reason to enhance further the energy economy of those unmanned electric vehicles. Off-road vehicles research was always challenging, but in the present work the nature of the off-road land is utilized to benefit from in order to enhance the energy consumption of those vehicles. An algorithm for energy consumption optimization for electrically driven unmanned wheeled vehicles is presented. The algorithm idea is based on the fact that in off-road conditions, when the vehicle passes a ditch or a hole, the kinetic energy gained while moving downhill could be utilized to reduce the energy consumption for moving uphill if the dimensions of the ditch/hole were known a distance ahead. Two manipulated variables are evaluated: the wheels DC motors supply voltage and the DC armature current. The developed algorithm is analysed and compared to the PID speed

controller and to the open-loop control of DC motors. The developed predictive controller achieved encouraging results compared to the PID speed control and also compared to the open-loop control. Also, the use of the DC armature current as a manipulated variable showed more noticeable improvement over using the DC input voltage. Experimental work was carried out to validate the predictive control algorithm. A mobile robot with two DC motor driven wheels was deployed to overcome a ditch-like hindrance. The experimental results verified the simulation results. A parametric study for the predictive control is conducted. The effect of changing the downhill angle and the uphill angle as well as the size of the prediction horizon on the consumed electric energy by the DC motors is addressed. The simulation results showed that, when using the proposed approach, the larger the prediction horizon, the lower the energy consumption is.

Acknowledgements

First and foremost, I would like to express my deepest thanks to Prof. Dan Necsulescu for his guidance, support, and friendship. To me he is the role model of an exceptional scientist and teacher. He has always been supportive, not only when I made progress but also when I made mistakes. I have also learned from him how to be professional in an academic career. It has been my great privilege to study my PhD under his guidance.

Lastly, I would like to thank my mother, my wife, and my kids for their love, support, patience, cooperation, and understanding throughout the course of my studies.

Contents

List of figures

List of tables

Nomenclature

Symbol	Description
A, B, C, D	discrete-time state-space matrices of the system
$\hat{A}, \hat{C}$	transformed discrete-time state-space matrices (for tuning observer dynamics)
A_C, B_C, C_C, D_C	continuous-time state-space matrices
a	car acceleration
$B_d(k)$	the unmeasured disturbance matrix (discrete-time)
$B_u(k)$	the input matrix (discrete-time)
$B_v(k)$	the output matrix (discrete-time)
B_1	the input matrix (in a system includes input derivatives)
B_2	the input derivative matrix
b	motor output shaft damping coefficient
C_m	the measured output matrix (discrete-time)
C_u	the unmeasured output matrix (discrete-time)
D_{dm}	the unmeasured disturbance feedforward to the measured output matrix (discrete-time)
D_{du}	the unmeasured disturbance feedforward to the unmeasured output matrix (discrete-time)
D_{vu}	the measured disturbance feedforward to the unmeasured output matrix
D_{vm}	the measured disturbance feedforward to the measured output matrix (discrete-time)
$d(k)$	the unmeasured disturbance
E_a	induced e.m.f. of the DC motor
f	coefficient of rolling resistance
G	gross vehicle/robot weight
$G(s)$	transfer function matrix
$G(z)$	discrete-time transfer function matrix
$G_{PID}(s)$	PID controller transfer function
g	gravitational acceleration
I_a	armature current
$i_{g.b}$	reduction gearbox ratio

J	total mass moment of inertia of the motor drive shaft and wheel assembly as expressed in motor rotor coordinates
J_k	cost function
J_m	mass moment of inertia of motor rotor
J_w	mass moment of inertia of mobile robot wheel
j	denotes the j^{th} component of a vector
K_a	motor constant
K_p,K_i,K_d	PID controller proportional, integral and derivative gains, respectively
K_{ob}	observer gain matrix/ Kalman filter gain matrix
$\widehat{K}_{ob}$	resultant observer gain matrix (for tuning observer dynamics)
L_a	inductance of motor coil
L_c	controllability matrix
L_0	observability matrix
M	optimization horizon
m_{robot}	mobile robot total mass
m_w	mass of mobile robot wheel
N or m	control horizon
n_y, n_u	number of output and input variables, respectively
$n1$	number of original state variables
n	the new number of state variables (the alternative state vector)
$P(i)$	the posteriori error covariance matrix at time i
P_{in}	input electric power
P_{out}	output mechanical power
p	prediction horizon
$\hat{Q}$, $\hat{R}$	nonnegative diagonal weighting matrices
q, r	number of outputs and inputs, respectively
R_a	armature coil wire resistance
$r(k)$	the current sample of the output reference
r_d	tire dynamic radius
S_a	integration of the armature voltage V_a
s	Laplace transform operator
T	sampling time
T_{em}	electromagnetic torque of the DC motor
T_{mech}	mechanical load torque applied to the motor shaft
U	composite vector contains the future control signal vectors
u_{k+i}	future control signal vector
$u_{jmin}, y_{jmin}, \Delta u_{jmin}$	the minimum values of the input, output and increment input, respectively

$u_{jmax}, y_{jmax}, \Delta u_{jmax}$	the maximum values of the input, output and increment input, respectively
$u_{jtarget}$	the set-point for the input variable
Δu_j	the increment for the j th input variable
V	vehicle/robot forward speed
V_a	input DC voltage
V_{brush}	voltage drop in brushes
$V_{min}^{u}, V_{min}^{y}, V_{min}^{\Delta u},$ $V_{max}^{u}, V_{max}^{y}, V_{max}^{\Delta u}$	the Equal Concern for the Relaxation (ECR) vectors which have nonnegative entries that quantify the concern for relaxing the corresponding constraints; the larger V, the softer the constraint
$v(k)$	the measured disturbance
W	composite vector contains the future reference trajectory vectors
w_{k+i}	future reference trajectory vector
$w(k)$	a vector of random variables represent the process noise
$w_{i,j}^{y}, w_{i,j}^{u}, w_{i,j}^{\Delta u}$	the nonnegative weights for the outputs, inputs and increment inputs respectively
x_k or $x(k)$	discrete-time system state-space vector at time k
$\hat{x}_k$ or $\hat{x}(k)$	predicted state vector at time k
$\tilde{x}(k)$	state error
$\tilde{x}_t(k)$	transformed state vector error (for tuning the observer dynamics)
Y	composite vector contains the future output vectors
y_{k+i}	future plant output vector
$y_m(k)$	the measured output
$y_u(k)$	the unmeasured output
z	z-transform operator
z_k	observation/measurement at time k
$\hat{z}_k$	predicted observation at time k
α	radius of a circle inside which the closed-loop observer pools lie
$\alpha(t)$	road grade angle
Γ	covariance matrix of the measurement noise vector
$\epsilon(k)$	a vector of random variables represent the measurement noise
ε	slack variable (for softening or relaxing the constraints)
$\zeta(t)$	the new continuous-time input vector after eliminating the input derivatives
$\eta(t)$	the new continuous-time output vector after eliminating the input derivatives
η_m	mechanical efficiency of the gearbox

Θ	covariance matrix of the process noise vector
Θ_0	mass factor (dimensionless)
ξ	normalized acceleration of the car
ρ_ε	the weight of the slack variable
$\chi(t)$	the new continuous-time state vector after eliminating the derivative inputs
ω	motor shaft angular speed

Chapter 1

Literature review and introduction

1.1 History of Unmanned Ground Vehicles

The Unmanned Ground Vehicle (UGV) is defined as "Any piece of mechanized equipment that moves across the surface of the ground and serves as a means of carrying or transporting something, but explicitly does NOT carry a human being" [1]. In robotics community, UGV is usually considered as an automated motion platform or just a mobile robot. A UGV is sometimes called Autonomous Land Vehicle (ALV). A discussion of such a broad universe of possible UGV systems needs some organizing principle, and in fact a taxonomy of UGV systems could be based upon any of a number of characteristics of each system, including:

- The purpose of the development effort (often, but not always, the performance of some application-specific mission);

- The specific reasons for choosing a UGV solution for the application (e.g., hazardous environment, strength or endurance requirements, size limitation);
- The "long pole" technological challenges, in terms of functionality, performance, or cost, posed by the application;
- The system's intended operating area (e.g., indoor environments, anywhere indoors, outdoors on roads, general cross-country terrain, the deep seafloor, etc.);
- The vehicle's mode of locomotion (e.g., wheels, tracks, or legs);
- How the vehicle's path is determined (i.e., control and navigation techniques employed).

The first major mobile robot was Shakey, Figure 1- 1, developed in the late 1960s to serve as a testbed for an Artificial Intelligence (AI) work funded by the US Defense Advanced Research Projects Agency (DARPA) at Stanford Research Institute (SRI) [2]. Shakey was a two-stepper motor-driven wheeled platform equipped with steerable TV camera, ultrasonic range finder, and touch sensors, connected via a Radio Frequency (RF) link to its SDS-940 mainframe computer that performed navigation and exploration tasks. As befit an AI testbed, the Shakey system could accept English sentence commands from the terminal operator and parse the sentence and call the appropriate FORTRAN or LISP

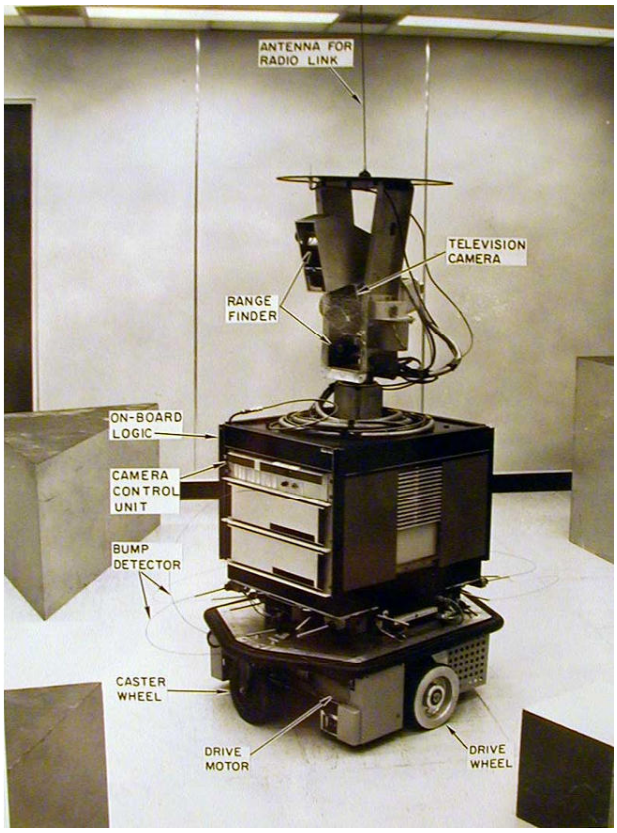


Figure 1- 1: The first major mobile robot, Shakey [2].

programs to carry out the command, directing the robot to push large wooden blocks around in its lab environment "world". While Shakey was considered a failure in its day because it never achieved autonomous operation, the project established functional and performance baselines for mobile robots, identified technological deficiencies, and helped to define the AI research agenda in such areas as planning, vision, and natural language processing [3].

From 1973 to 1981, Hans Moravec led the Stanford Cart project at the Stanford University AI lab, exploring navigation and obstacle avoidance issues

using a sophisticated stereo vision system [4]. The cart was driven by four steerable electric motors. The Cart's single TV camera was moved to each of nine different positions atop its simple mobility base, and the resulting images were processed by the off-board KL-10 mainframe. The program has successfully driven the cart through several 20 meter indoor courses with 3 or 4 avoiding swerves in each of them, Figure 1- 2. The system was too slow to the extent that it took 5 hours to accomplish each of these courses. A less successful outdoor run, in which the cart swerves around 2 obstacles but collided with the third, was also done. Harsh lighting, very bright surfaces next to dark shadows, and moving of shadows during the cart's creeping progress caused poor pictures which resulted in poor outdoor performance.

Moravec moved to Carnegie Mellon University (CMU) in 1981 and continued his work on the smaller CMU Rover [5] indoor platform. CMU became a major leader in mobile robot research during the 1980s, with its Navlab vehicle as the focus for much of the work [6]. Thrope led a team to develop a testbed to be used for integrating perception and navigation capabilities. The sensory capabilities of the system were color vision and 3D vision using an active sensor. The Navlab was based on a commercial van chassis with hydraulic drive and electric steering. It had room for onboard computers including Sun workstation and a Warp systolic computer. The Navlab was also equipped a TV

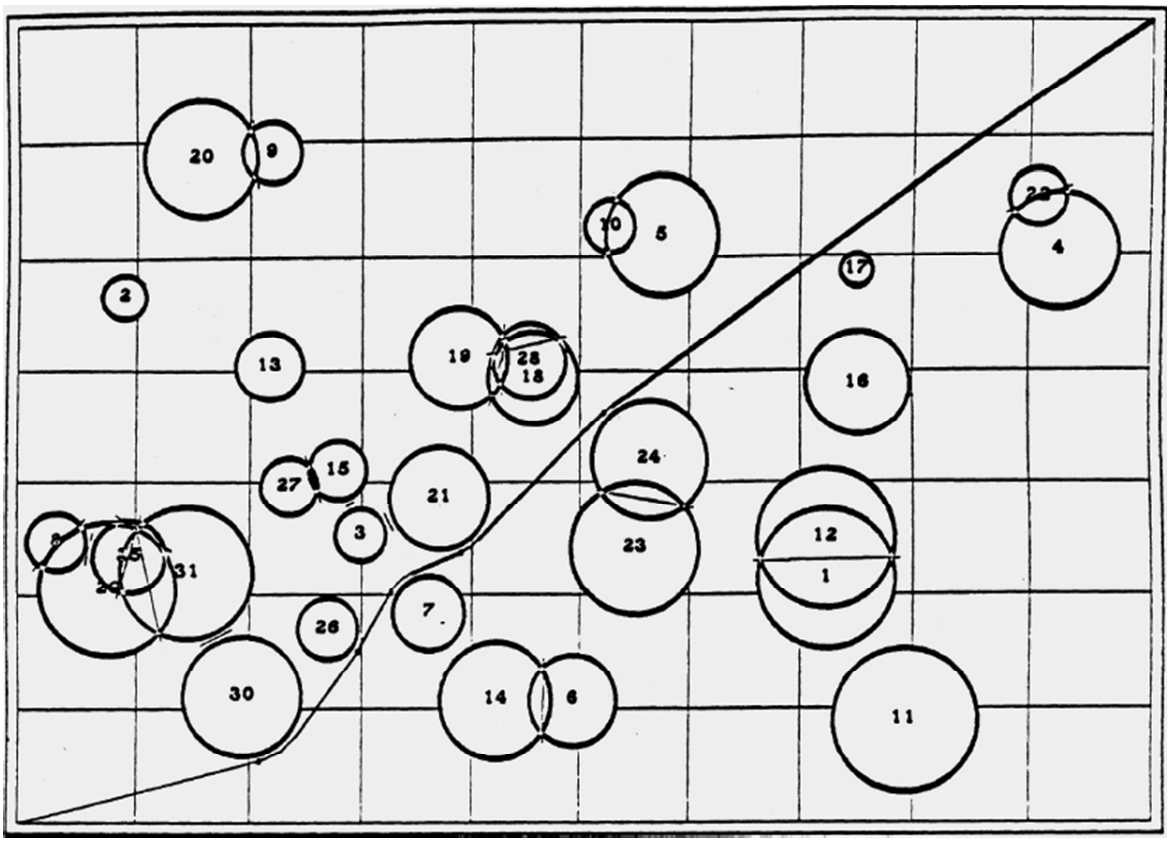


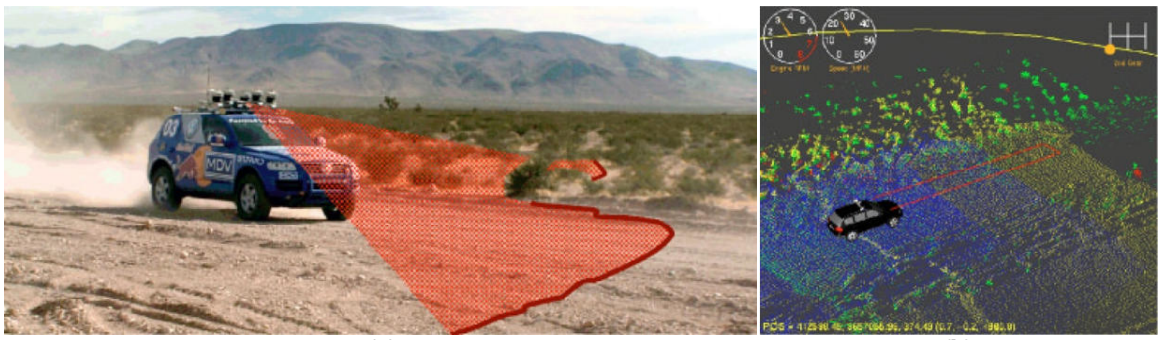
Figure 1- 2: The shortest path finder's solution to a randomly constructed problem. The route is from the lower left corner to the upper right. The numbered circles are the obstacles, and the wiggly line is the solution, Moravec [4].

camera and a laser range finder. The interface between the computers and the driving hardware was provided by a controller which could steer the van along circular arcs. The vehicle had room for up to four researchers. The system developed successfully driven the Navlab, but the speed was still relatively slow, 50 cm/s, and the vehicle negotiated a seriously flawed curves. Also, the Navlab wasn't able to recognize road intersections.

DARPA started to develop an outdoor application which was the Autonomous Land Vehicle (ALV) as a part of the Strategic Computing (SC) program from 1975 to 1988. The program focused on autonomous navigation and tactical decision making. The ALV was an eight-wheeled, eight ton vehicle. The wheels did not pivot as in conventional vehicles. Instead the vehicle was "skid steered" [7]. With the cooperation of the Environmental Research Institute of Michigan (ERIM), vision sensors were supported to the ALV program [8].

Following the success of DARPA in ALV program, effort were concentrated in DARPA DEMO I, II, and III programs from 1990 to 2001 [9], [10]. DEMO I (1991) was accomplished using High Mobility Multi-purpose Wheeled Vehicle (HMMWV) platforms, equipped with all sensors and actuators, to demonstrate mobility on off-road and on-road environments, primarily in teleoperated mode. In DEMO II, three Semiautonomous Surrogate Vehicles (SSVs), automated by HMMWVs and outfitted with Forward Looking Infrared (FLIR), Light Detection and Ranging (LIDAR), and Charge-Coupled Device (CCD) camera sensors, were incorporated in reconnaissance, surveillance, and target acquisition tasks. The DEMO III focused on advanced technologies for small survival UGVs. Equipped with CCD/FLIR stereo vision and LIDAR sensors for daylight/night obstacle detection, the vehicles could navigate autonomously to target destinations, set by operator, at maximum speeds of 32 km/h during the day time and 16 km/h at night on rough terrain.

Starting in 2004, DARPA has initiated a new program called DARPA Grand Challenge. It was a prize competition for UGVs, funded by the DARPA. Congress has authorized DARPA to award cash prizes to further DARPA's mission to sponsor revolutionary, high-payoff research that bridges the gap between fundamental discoveries and military use. DARPA had technologies needed to create the first fully autonomous ground vehicles capable of completing a substantial off-road course within a limited time. The second DARPA Grand Challenge competition, which took place in October 2005 [11], had further advanced vehicle requirements to include autonomous operation in a mock environment. Five UGVs out of twenty three finished the off-road trade with a distance of 212 km without stop, collision, or even human interference. The first place UGV, Stanley (developed by Stanford University team, Figure 1- 3) [12], finished the route in 6 hours 53 minutes, and 58 seconds while the last place



(a) (b)

Figure 1- 3: Stanley: The robot that won the DARPA grand challenge 2005.
(a) Illustration of a laser sensor: The sensor is angled downward to scan the terrain in front of the vehicle as it moves. Stanley possesses five such sensors, mounted at five different angles. (b) Each laser acquires a three-dimensional (3D) point cloud over time. The point cloud is analyzed for drivable terrain and potential obstacles [12].

UGV, TerraMax (developed by Oshkosh Truck Corporation), finished the race in 12 hours and 51 minutes.

1.2 Power consumption enhancements

Recently, as the world goes in the direction of alternative cleaner energy sources, developing a UGV that doesn't depend on fossil fuels became a point of interest. Different models were developed to increase the battery life of a battery operated mobile robot.

In 1994, Spangelo & Egeland, [13], applied an energy-optimal control method for trajectory planning and collision avoidance for underwater vehicles. They modeled the vehicle with six-dimensional nonlinear and coupled equations of motion. The vehicle was controlled in all six degrees of freedom by dc-motor driven thrusters. The controls were approximated by piecewise constant and piecewise linear parameterization. In both cases, computations were performed using different numbers of parameters and good results were achieved for a moderate number of parameters. Best results were achieved using piecewise linear parameterization. Also, they found, for such vehicles that are symmetric about the longitudinal axis, that computational effort could be reduced by removing the roll from the state-space equations resulting in 10 states and 5 controls instead of 12 states and 6 controls, Figure 1- 4.

Another technique is to increase the battery life of a skid steered wheeled ground vehicle by modifying its path planning and its inner and outer wheel motors during turning to not violate the power limitations as in [14]. Another way presented by Zhang *et al.* which is controlling the speed of the mobile robot

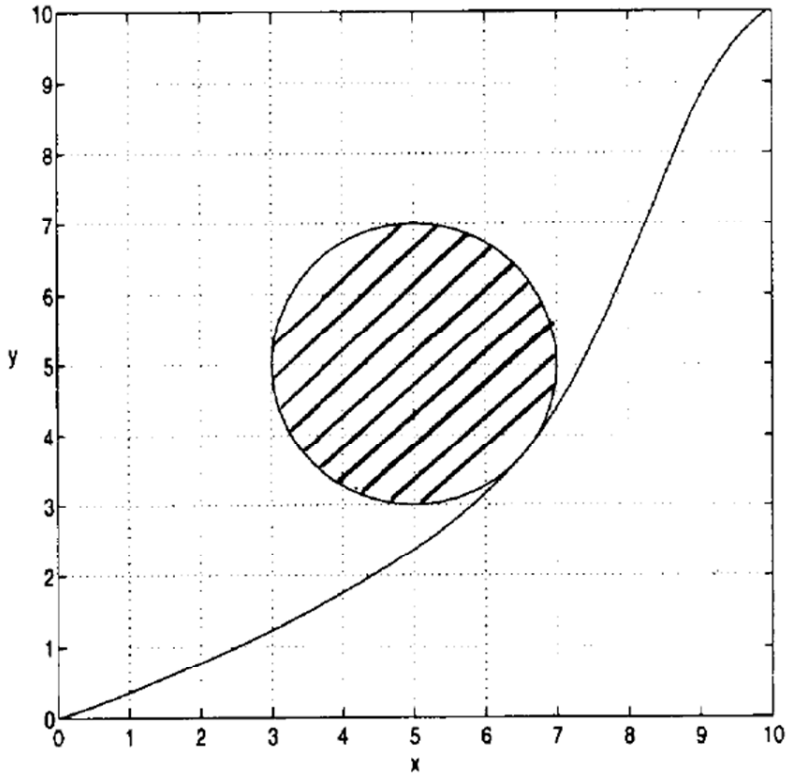


Figure 1- 4: x-y coordinates plot of optimal trajectory with the path constraints of underwater vehicle (units in meters), Spangelo & Egeland [13].

as well as the frequency of its on-board processor to optimize the power consumption [15].

In 2008, Vanualailai *et al*. [16] introduced an asymptotically stable collision-avoidance system applied on a point mass using the artificial potential

fields with a solution to the local minima problem. Since the system is asymptotically stable, its Lyapunov function, which produces artificial potential fields around the goal and the obstacle, has no local minima, Figure 1- 5. The system was applied successfully to a non-holonomic planner mobile robot moving in the presence of a static obstacle. In 2009, Saravanan *et al*. [17] presented an evolution for two new general strategies for the off-line tridimen-

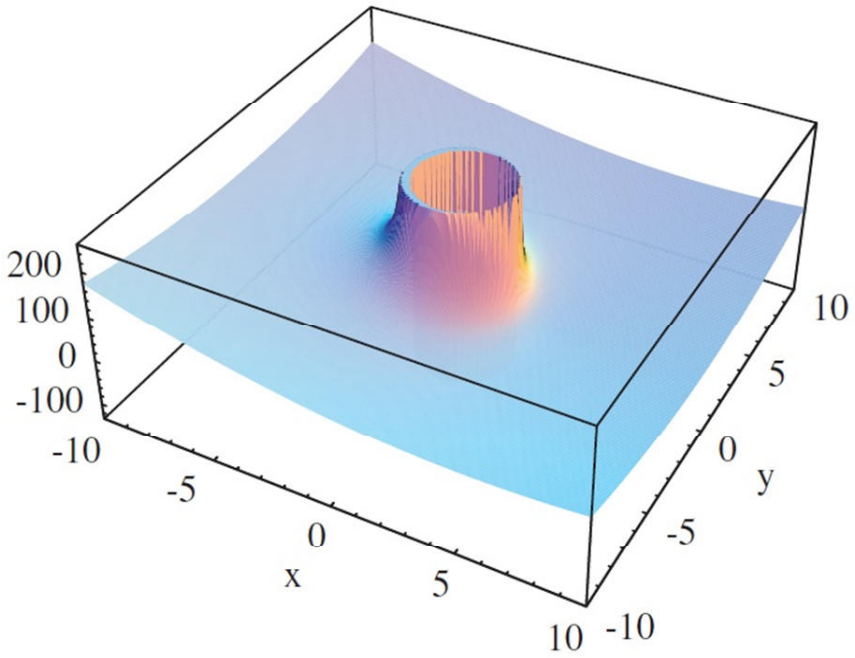


Figure 1- 5: Lyapunov function is the sum of attractive and repulsive potentials which prevented from having local minima, Vanualailai *et al*. [16].

sional optimal trajectory planning of the industrial robot manipulator (STANFORD robot) in the presence of fixed obstacles. The new two strategies used were the NGSA-II and MODE. These two strategies were also compared to the SUMT strategy presented in by Saramago and Steffen in 2001 [18]. The

analysis showed that both NGSA-II and MODE were better than SUMT. The analysis also showed that MODE technique converged quicker than NSGA-II technique. Also the computational time to find Pareto optimal front in MODE was one-third of that of the NSGA-II. This means that it was faster. On the other hand NSGA-II had more number of non-dominant solutions than the MODE. Also Pareto optimal front from NSGA-II was better than that of MODE according to the distance metric. So NSGA-II technique is best for this multi criterion optimization problem, if the user wants a larger number of solutions for this choice. This work opens the door for further investigations on how the evolutionary optimization techniques can be used to solve complex problems.

In 2010, Jang *et al*. [19] suggested a longitudinal control algorithm for controlling the UGV velocity while driven on downhill slope. They used a rough estimation of what they called “Safe Velocity” based on only the road slope and current vehicle velocity. They applied the algorithm practically on a real test vehicle using an Inertial Measurement Unit (IMU) for the slope measurement and a brushless DC motor for actuating the accelerator and brake pedal. The algorithm gave satisfactory results compared to the rough estimations, Figure 1-6 and Figure 1- 7.

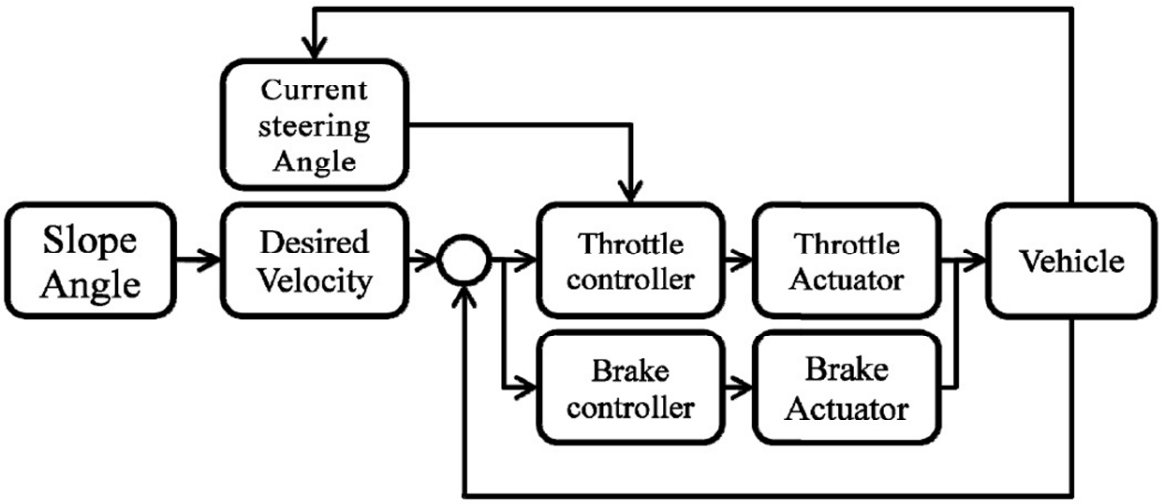


Figure 1- 6: Block diagram of slope road driving, Jang *et al.* [19].

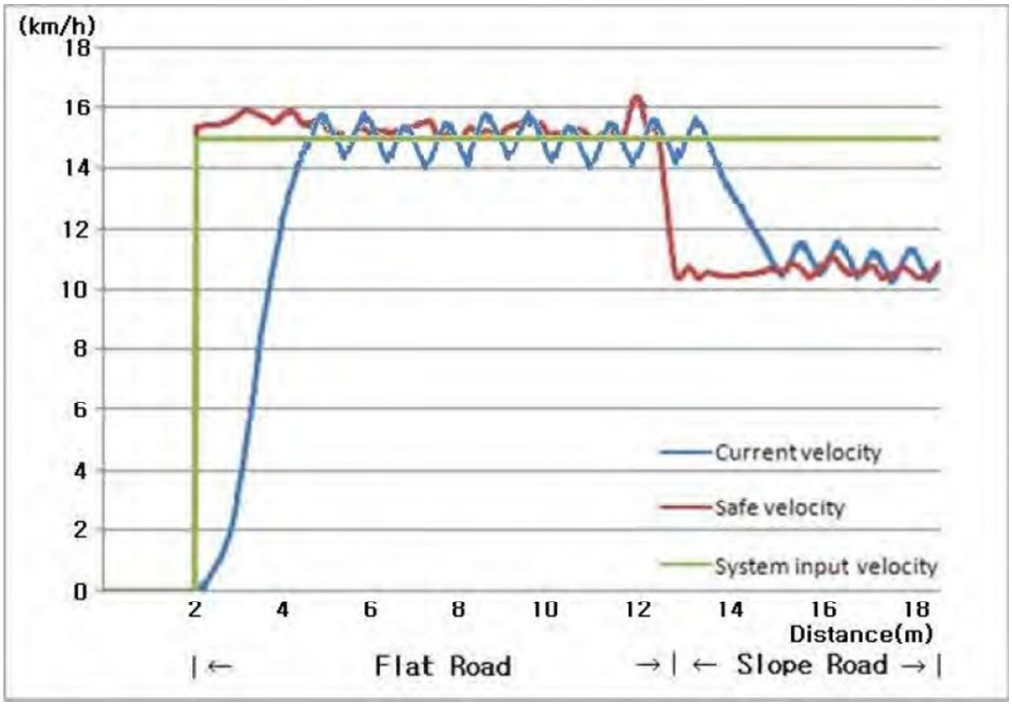


Figure 1- 7: Velocity of the vehicle on the downhill road, Jang *et al.* [19].

In 2010, Morales *et al*. [20] presented a simplified power consumption model and identification for wheeled skid-steer robotic vehicles on hard horizontal ground. This work was an extension to a similar work that they applied before on a tracked mobile robot [21]. This simplified static model estimates the motor power consumption as a function of the left- and right-side wheels' speeds. The model is defined through three different parameters: the x-coordinate of the tread's instantaneous center of rotation on the ground plane, a traction resistance constant and the ground-wheel friction coefficient. They also proposed an experimental identification procedure using the simplex optimization method [22], [23] to obtain the model parameters. They also performed a power analysis of the four wheeled mobile robot Quadriga with different loads on concrete and marble floorings. The Hall Effect current sensor was used to measure the instantaneous power consumption of each motor for not being affected by the consumption of other vehicle components unrelated to the traction system. The model and the identification procedure have been validated on the four-wheeled skid-steer Quadriga mobile robot with four combinations of load and terrain types, Figure 1- 8. The simple model has shown to approximate closely the measured power consumption.

In 2007, Ngo *et al*. [24] proposed a new concept which is the potentially distributable energy. The proposed concept enabled the robots in a multi-robot

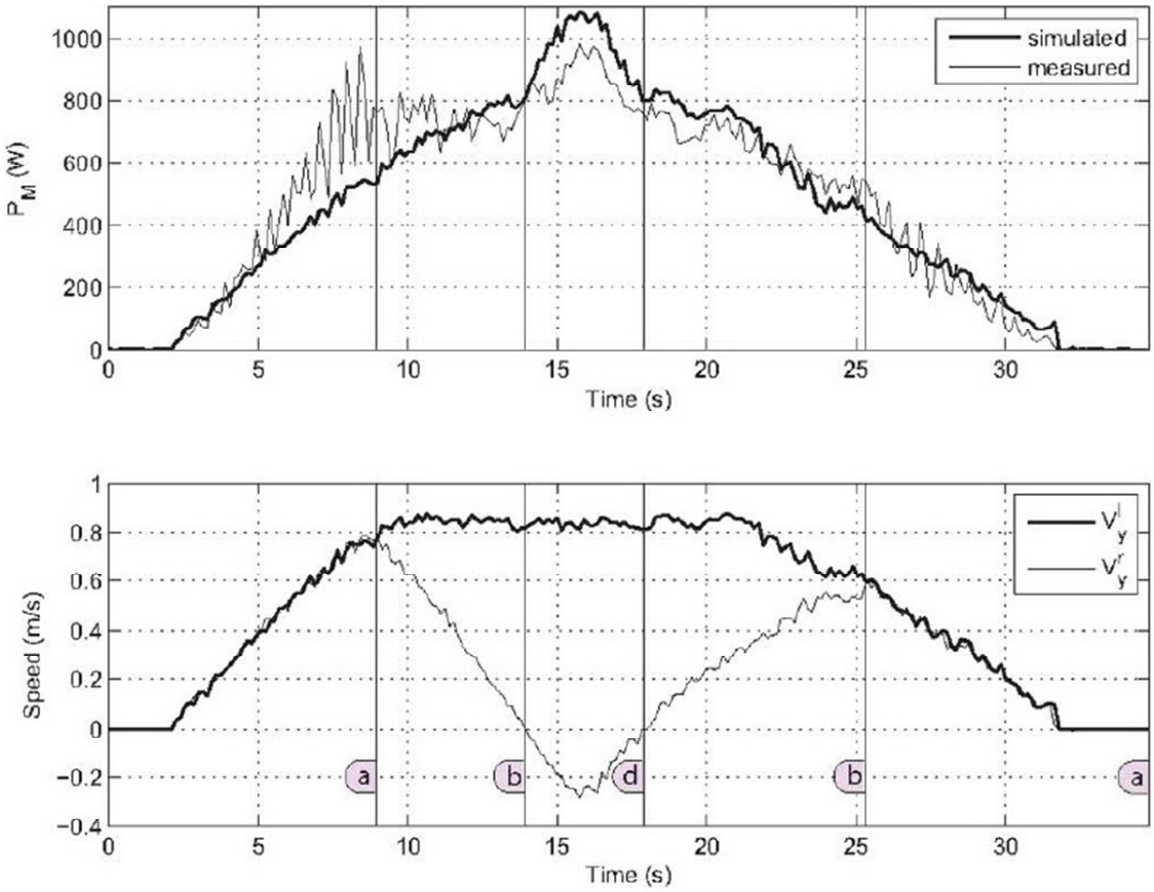


Figure 1- 8: A sample from the validation of the power consumption model of a skid-steer mobile robot during spiral-like paths developed by Morales *et al.* [21].

field to partially recharge each other's batteries whenever they went low instead of going to the fixed mother charging station, Figure 1- 9. The concept also assumed that the robots were not only capable of self-recharging, but also exchanging the batteries with each other. The concept aimed to expand the autonomy of the robots by increasing their operating time without the human intervention. The authors simulated the multiple mobile robots and then issued

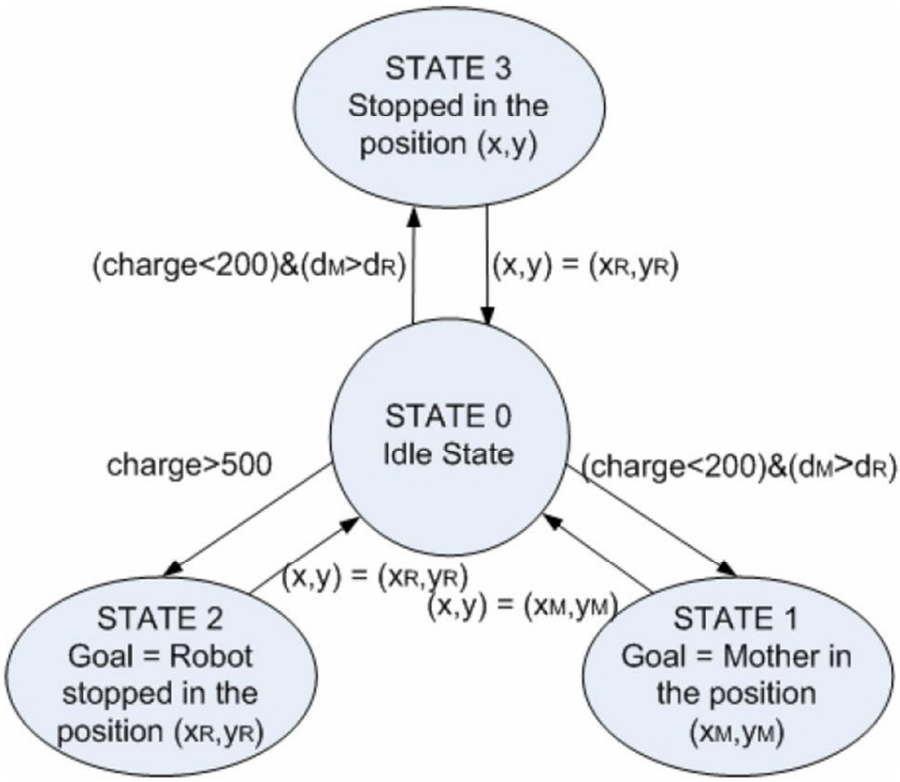


Figure 1- 9: The four robot states in the potentially distributable energy in a multi-robot field as proposed by Ngo *et al*. [24].

the rules of battery exchange which was formulated under constrains of workload, distance, and remaining capacity, Figure 1- 10. They initialized the concept borrowing from the artificial potential fields in which the gradient of attraction was used to direct the robot to reach the goal. Through a variety of locations of mother charging station, number of robots, and size of experimental field, technique of deploying a number of robots in a specific area has been examined to find out an optimized correction among parameters for overall power of the multi-robot system.

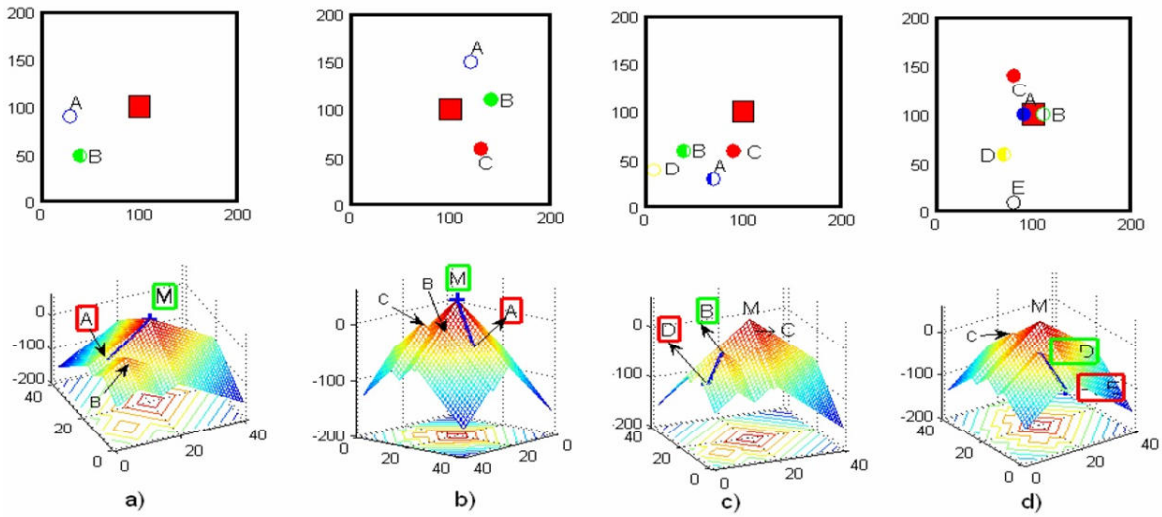


Figure 1- 10: Potentially distributable energy of multi-robot field, Ngo *et al*.[24].
Graphs a, b, c and d are different time snapshots in the simulation. The upper graphs show the positions of the robots and the mother station (red square), while the lower graphs show the energy content of each robot which is represented by the height of each robot's pyramid.

In 2007, a reconfigurable mobile robot with multi-maneuver mode was presented [25]. The developed prototype was with independent propulsion and individual steering (4WD4WS). Different prototypes could be achieved by reconfiguration of the wheelbase, wheel clearance, stability margin, and Center of Gravity (CG) height. The prototype was built to give the ability to the researcher to give the mobile robot many scenarios with full control during off-road maneuvers.

In 2009, Suntharalingam *et al*. [26], presented a gear locking mechanism to extend the consistent power operating region of an electric motor to enhance acceleration and regenerative braking efficiency, Figure 1- 11 and Figure 1- 12.

They applied that to a hybrid vehicle using simulation. The gear switching consisted of three gear sets actuated by three switches which in turn were controlled by a Transmission Control Unit (TCU). Motor speed, wheel speed, acceleration demand and braking demand were the key variables fed to the TCU and, accordingly, the TCU would enable the appropriate gear ratio to make the motor/generator proposed variable gear transmission propulsion system acceleration more significant (33% higher acceleration performance) compared with the fixed transmission system. Similarly, the regenerative braking distance was reduced by 14% compared with the obtained by the fixed transmission system.

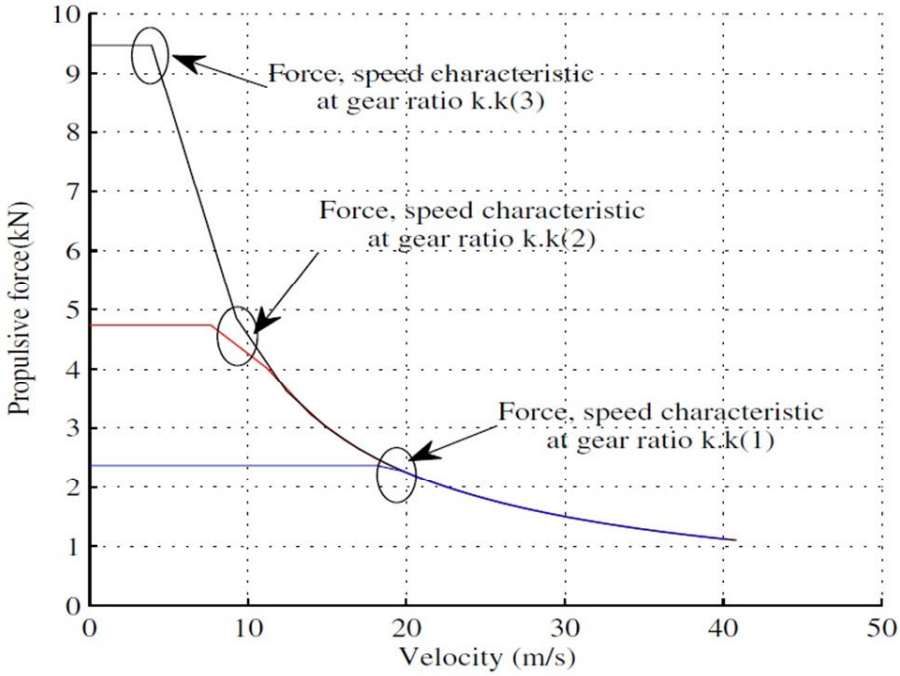


Figure 1- 11: Force speed characteristics of the electric propulsion system with different gear ratios, Suntharalingam *et al.*[26].

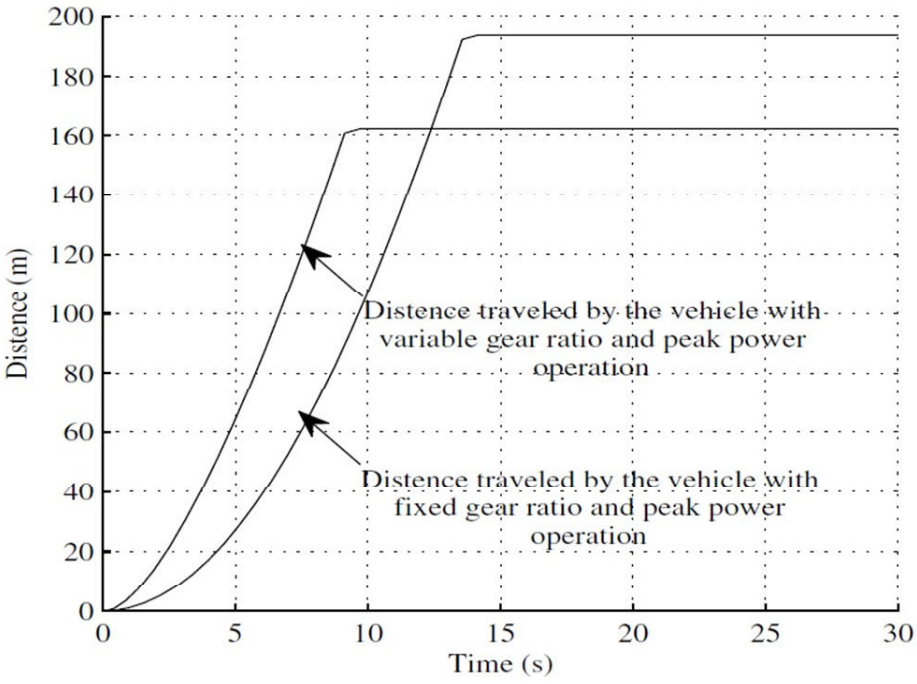
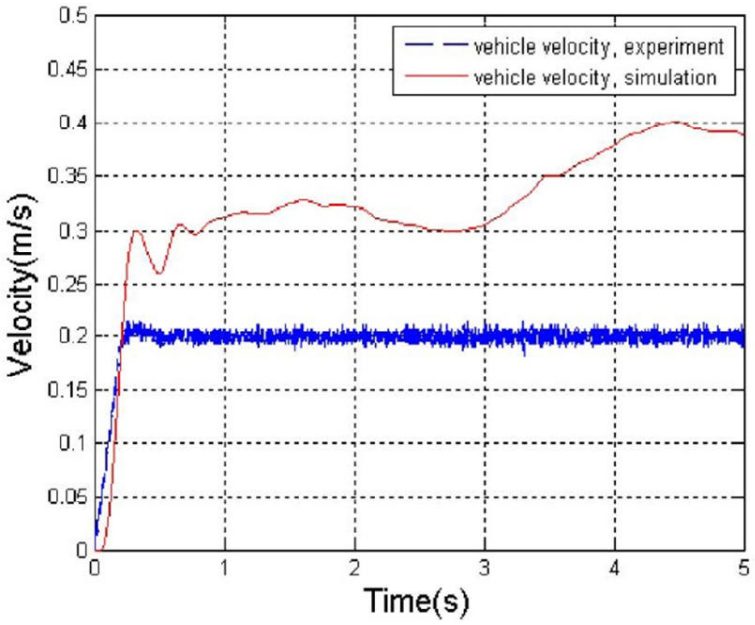
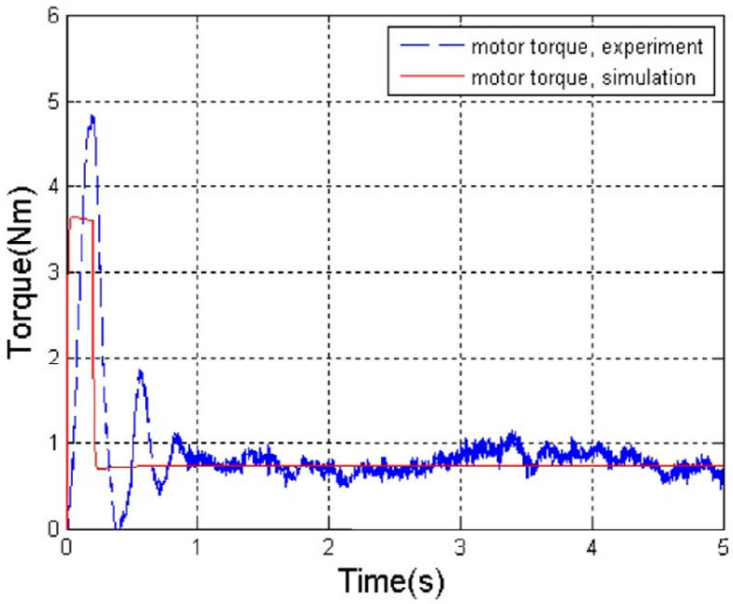


Figure 1- 12: Distance travelled by the vehicle when the motor operates at peak power rating with fixed and variable gear system topologies, Suntharalingam *et al*.[26].

In 2010, Wei Yu *et al*. [27] presented dynamic a model of a skid-steered wheeled vehicle and accomplished an experimental verification to the model. The model was for general 2D motion and linear 3D motion. The model was characterized by the coefficient of rolling resistance, the coefficient of friction and shear deformation modulus which had terrain dependent values. The model also included motor saturation and power limitations. They made a comparison between closed-loop control and open-loop control of the velocity, Figure 1- 13 and Figure 1- 14.

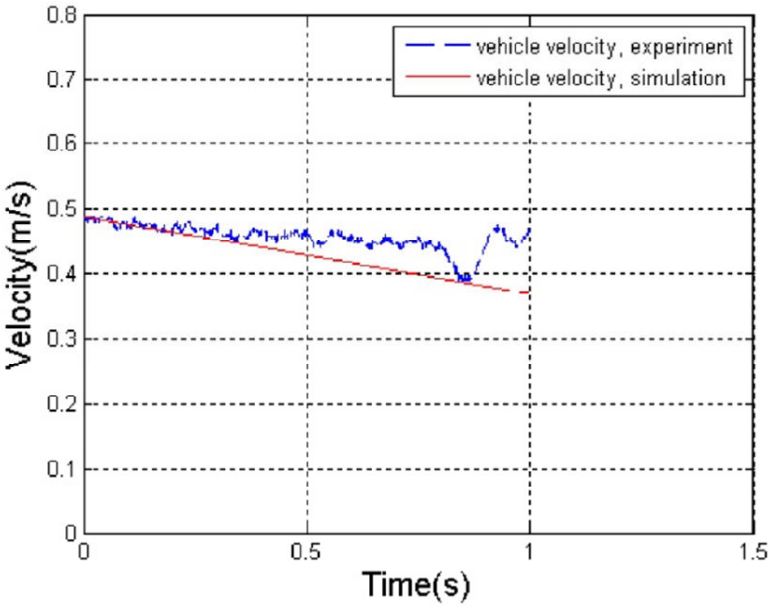


(a)

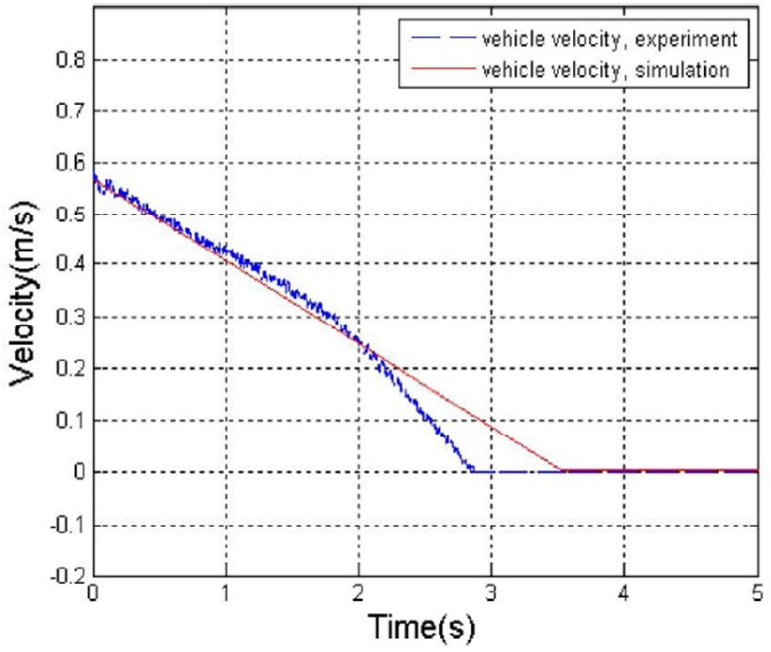


(b)

Figure 1- 13: Open-loop velocity comparison when the vehicle is commanded with the voltage V (a) that corresponds to the torque τ (b), Wei Yu *et al.*[27].



(a)



(b)

Figure 1- 14: Closed-loop vehicle velocity comparison when commanded linear velocity 1.2 m/s, Wei Yu *et al.*[27].

(a) White-board hill climbing with slope $\beta=15^{o}$ and initial velocity 0.57 m/s.
(b) White-board hill climbing with slope $\beta=12.5^{o}$ and initial velocity 0.49 m/s.

The models with the closed-loop modified PID controller gave accurate predictions for constant linear and angular velocities only, while they tended to give prediction inaccuracy for large accelerations. In addition, the models were limited to hard terrain where significant sinkage doesn't occur.

In 2012, Sadrpour *et al.* [28] introduced a prediction framework of end-of-mission energy to help preventing the UGV from failure due to power exhaustion. Two approaches where presented: a linear regression model motivated by the vehicle longitudinal dynamics in the absence of prior knowledge of the driving conditions, e.g. road average grade and velocity profiles; and a Bayesian regression, in case of the presence of a prior knowledge of the driving conditions, that integrates the prior knowledge with real-time measurements for improved prediction. Figure 1- 15 shows how the framework connected the two approaches. The hypothetical mission consisted of two road segments were used for the evaluation of the two approaches: a very rough flat road segment A to B, and a roughly paved flat road segment B to A, Figure 1- 16. The linear regression model overestimated the required end-of-mission energy prediction, while the Bayesian regression showed more accurate prediction, specially, at the initial stages of the mission, Figure 1- 17.

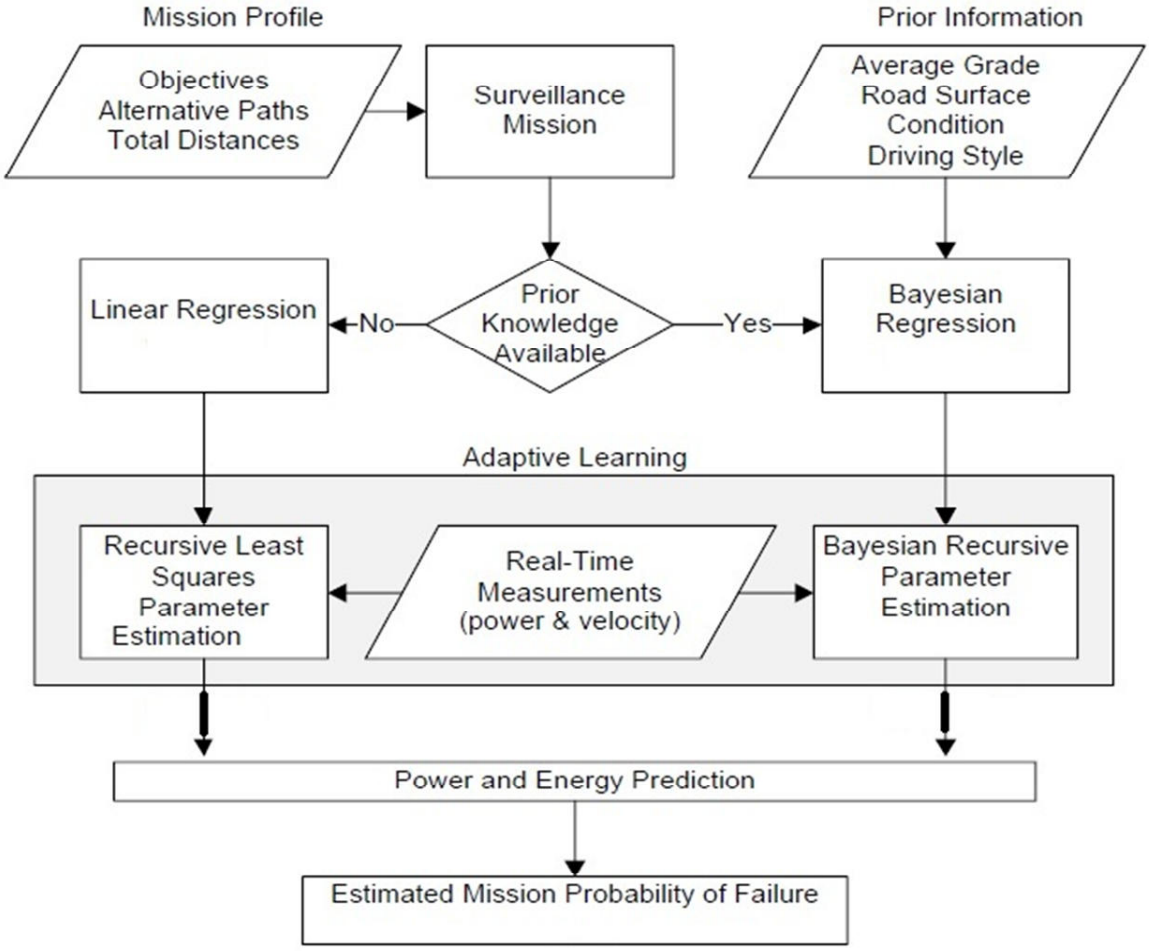


Figure 1- 15: Overview of the methodology for prediction of mission energy, Sadrpour *et al.* [28].

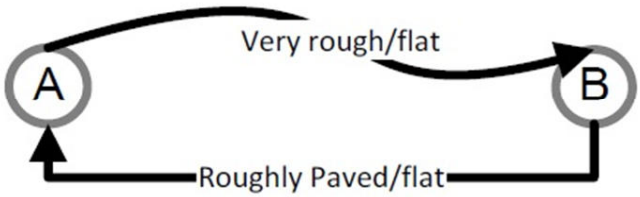


Figure 1- 16: The mission is composed of two road segments, Sadrpour *et al.*[28].

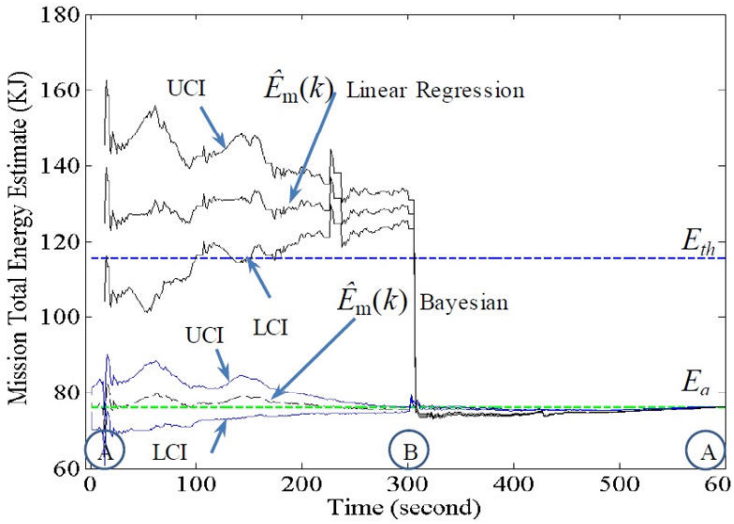


Figure 1- 17: End-of-mission energy prediction vs. time, Sadrpour *et al.*[28].
The linear regression model overestimates the energy requirement in first road segment. E_a is the actual total energy used (green dashed line). E_{th} is the failure threshold (blue dashed line) and is defined as 1.5 times the expected total mission energy based on the initial prior information.

1.3 Model Predictive Control

The Model Predictive Control (MPC) refers to a class of computer control algorithms that utilize an explicit process model to predict the future response of a plant, [29], Figure 1- 18. At each control interval an MPC algorithm attempts to optimize future plant behaviour by computing a sequence of future manipulated variable adjustments. The first input in the optimal sequence is then sent into the plant, and the entire calculation is repeated at subsequent control intervals. Originally developed to meet the specialized control needs of power plants and petroleum refineries, MPC technology can now be found in a wide variety of application areas including chemicals, food processing,

automotive, and aerospace applications. For more information about the history and development of the MPC, look in [29] and [30].

In many plants, the simultaneous minimization of the size and duration of the state constraint violations is not a conflicting objective. The optimal way to handle infeasibility is simply to minimize both size and duration; regulator performance may then be optimized, subject to the "optimally" relaxed state

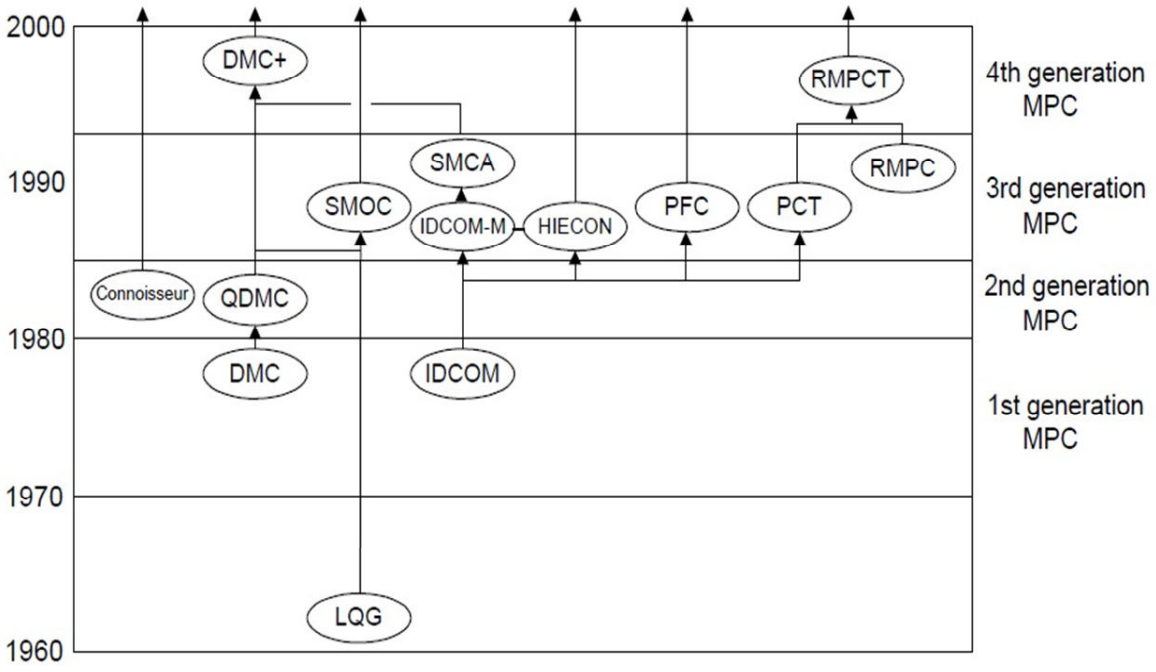


Figure 1- 18: Approximate genealogy of linear MPC algorithms, Qin and Badgwell [29].

constraints. Unfortunately, not all infeasibilities are as easily resolved. In some cases, such as non-minimum phase plants, a reduction in size of violation can be obtained at the cost of a large increase in duration of violation, and vice versa. The optimization of constraint violations then becomes a multi-objective

problem. Figure 1- 19 shows two different MPC controllers' resolutions of an infeasibility problem.

There are different types of the MPC, but here Dynamic Matrix Control (DMC) is first considered due to its simplicity and effectiveness in the same time. The work presented by Berlin and Frank in 1994 [31] was studied as it showed

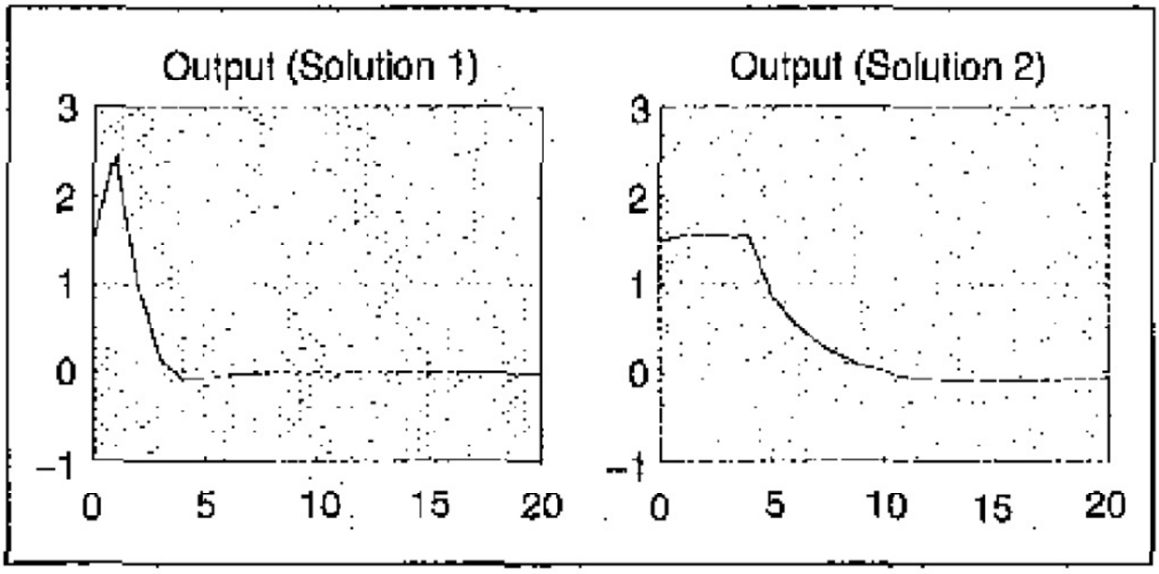


Figure 1- 19: Two MPC controllers' resolution of output infeasibility: output versus time, Rawlings [30].

Solution (1) minimizes duration of constraint violation; Solution (2) minimizes peak size of constraint violation.

interesting results. They presented a design and realization of a Multi-Input-Multi-Output (MIMO) predictive controller for a 3-tank system. They based their model on an extended Dynamic Matrix Control (DMC) law. The law used the parametric plant model which yields an easy and effective controller design method for discrete time tracking controller. They showed how the predictive

controllers can guarantee an excellent reference feedforward decoupling without the necessity for additional computation. They, also, showed how it is possible to attenuate known or assumed disturbance models using the proposed controller. They illustrated their theoretical results considering a 3-tank system with two inputs, two outputs and multiple disturbances. The proposed controller was designed using a linearized model of the nonlinear system. Modeling errors and leaks are considered as first order disturbances. Realizing the predictive controller, an excellent tracking behaviour can be observed together with a very good reference feedforward decoupling.

1.4 Statement of the problem

The autonomy property is enhanced by the reduction of energy consumption. That's because the more we leave the UGV without a human intervention, the more is the UGV autonomous. In order to expand the range of operation of the UGV, it is required to modulate the energy consumption during the routine operation.

The fact that this work is concerned with the off-road operation, leads to utilizing the road irregularities to reduce the energy consumption, specially the up hills and down slopes. For instance, when the UGV faces a deep and large hole, will it be feasible from the energy consumption point of view, to pass through it taking the benefit of the kinetic energy to arrive to the other side of it?

If yes, what speed profile should the UGV follow to overcome that hole and reach the other side of it? The same questions apply when the UGV faces a hill. Of course, there are some situations when the answer to those questions is “no”. This is not concerning the energy consumption question only, but it could be because of the width of the hole is narrower than that of the UGV, or at least some portions of it, or it could be very irregular to the extent that may cause the UGV to tip over, or it might be so steep to the extent that it may cause the UGV to pitch over and fall inside the hole. Indeed, the answers to these questions need complex algorithms to make the UGV decide what choice it should go to. Also, the first issue is, could the UGV have an accurate self-built map to make accurate decisions like that?

The main question that needs to be answered here is; *is there a way to benefit from the off-road environment to enhance the energy consumption of a UGV to expand its autonomy?*

Compared to the previously discussed contributions, the novelty in the present work consists in implementing an optimization algorithm capable of involving a prior knowledge of the road profile, that is the *Predictive Control*.

1.5 Methodology

The work is divided into two main phases:

1.5.1 Algorithm to enhance energy economy during off-road motion

The approach used in the present work is to test the UGV while negotiating a hole/ditch in a straight line motion. The standard profile shape of the hole/ditch is as follows: level road, downhill, level road, uphill, and finally, level road again. The ability of the UGV to overcome this hole/ditch is evaluated using different algorithms: first, without using any closed loop control, secondly, using a Proportional Integral Controller (PID), thirdly, using an MPC controller. An indoor robot is modeled to get the preliminary results of the simulation to be able to compare it with the experimental work. Each algorithm is tested twice to control the DC motors speeds: first by controlling the DC motor armature voltage, and then by controlling the DC motor armature current.

1.5.2 Experimental work to validate the simulation results

A two driven wheel mobile robot is implemented to experimentally verify the results obtained from the simulation. A wood made ramp is built to reproduce the ditch. The experiments are conducted using open-loop control, PID control and MPC control. Different ramp angle settings are considered. The energy consumption is evaluated in all the experiments. The MPC controller showed encouraging results. The experimental results showed that if the robot

knew in advance the path ahead, the MPC could give noticeable improvement in the energy consumption while negotiating a hole/ditch. Also, the experimental results showed good agreement with the simulation results. The detailed analysis and results discussion will be explained in the following chapters.

Chapter 2

Simulation

2.1 Simulation of a PM DC motor

In the present work it is assumed first that we have two Permanent Magnet (PM) DC motors as shown in Figure 2-1. The equations of each of them are as follows [32]:

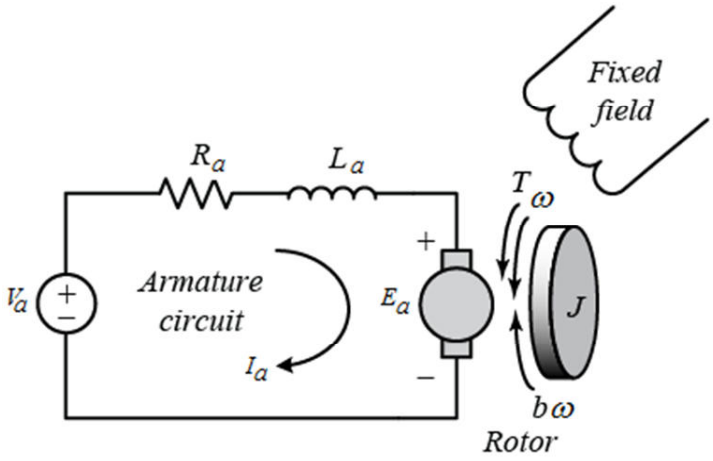


Figure 2-1: PM DC motor [32].

$$V_a - V_{brush} - R_a.I_a - K_a.\omega = L_a.dI_a/dt \quad (2\text{-}1)$$

$$I_a.K_a - \omega.b - T_{mech} = J.d\omega/dt \quad (2\text{-}2)$$

where:

V_a is the input DC voltage, V_{brush} is the voltage drop in brushes, R_a is the armature coil wire resistance, I_a is the armature current, K_a motor constant, ω is the motor angular speed, b is the damping coefficient, T_{mech} is the mechanical load torque applied to the motor shaft, L_a is the inductance of the motor coil, and J is the mass moment of inertia of the motor drive shaft and wheel assembly with respect to the motor rotor.

2.1.1 Equations of motion of the vehicle

The vehicle is assumed to move forward with relatively low speed and air resistance is neglected. The total resisting force due the car motion, ΣR, can be represented as follows [33]:

$$\Sigma R = F_a + F_r + F_\alpha \quad (2\text{-}3)$$

where, F_a is the inertia force resistance, F_r is the rolling resistance force, and F_α is the grade resistance. Also, it could be represented as

$$\Sigma R = \Theta_0 ma + Gf\cos\alpha(t) + G\sin\alpha(t) \quad (2\text{-}4)$$

where, G is the gross car weight, Θ_0 is the mass factor (dimensionless usually has values nearby 1.05 [33]) which includes the contribution of the rotating wheels and other rotating masses in the car, m_{robot} is the car (mobile robot) total mass, a is the car acceleration, f is the coefficient of rolling resistance between the wheel and the ground, and $\alpha(t)$ is the angle of the slope of the road.

The total resisting load torque on each of the two motors due the car motion, T_{mech}, can be represented in the following equation:

$$T_{mech} = \frac{G.r_d}{2}\left[\frac{\Theta_0.r_d}{g}\frac{d\omega}{dt} + f\cos\alpha(t) + \sin\alpha(t)\right] \quad (2\text{-}5)$$

where, r_d is the dynamic radius of the car wheel, and g is the gravitational acceleration. If a reduction gearbox is added to each motor, then the total mechanical load torque will be modified to be

$$T_{mech} = \frac{G.r_d.i_{g.b}}{2\eta_m}\left[\frac{\Theta_0.r_d.i_{g.b}}{g}\frac{d\omega}{dt} + f\cos\alpha(t) + \sin\alpha(t)\right] \quad (2\text{-}6)$$

where, $i_{g.b}$ is the reduction gearbox ratio, and η_m is the mechanical efficiency of the gearbox.

All simulations are done using MATLAB™ SIMULINK™ for a total sampling time of 14s. The solver used is ode45 (Dormand-Prince) with a variable-step size and a relative tolerance of 10^{-3}. The robot is attempting to

cross a ditch 4.3 m wide with 1.2 level road and 1.5 m downhill ramp before it reaches a 1.1 m uphill ramp and 0.5 m level road again, see Figure 2- 2. The downhill slope angle of the ditch is -47^{o} and the uphill angle is 13^{o}. The lengths of the ramps portions are chosen to accommodate the lab size while the angles are chosen to match the gradeability of the robot. The block named "Road data" in the model shown in Figure 2-3 converts the road profile geometry into a time varying slope angle to be fed into the car (vehicle) model.

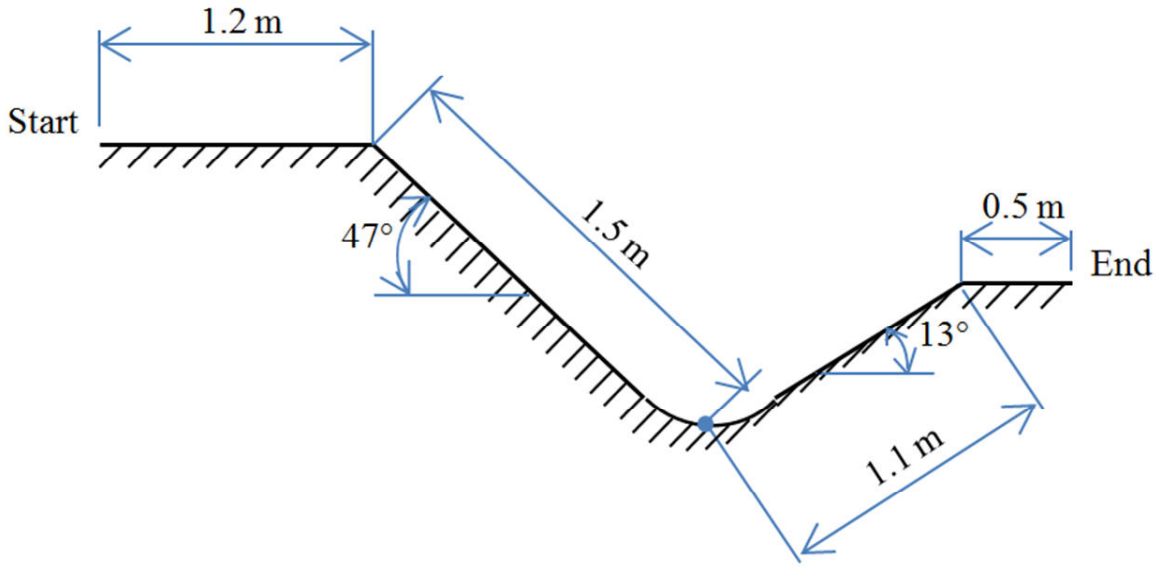


Figure 2- 2: Dimensions of the proposed ditch.

2.1.2 Constant input DC voltage

Figure 2-3 shows the SIMULINK model for this electric car. Eq. (2-1) and eq. (2-2) represent the DC motor model block in Figure 2-3, while eq. (2-6) represents the car model block. Table 2- 1 shows the car parameters which were chosen to match the parameters of the Dr. Robot X80® mobile robot [34], the Canon In. FP-

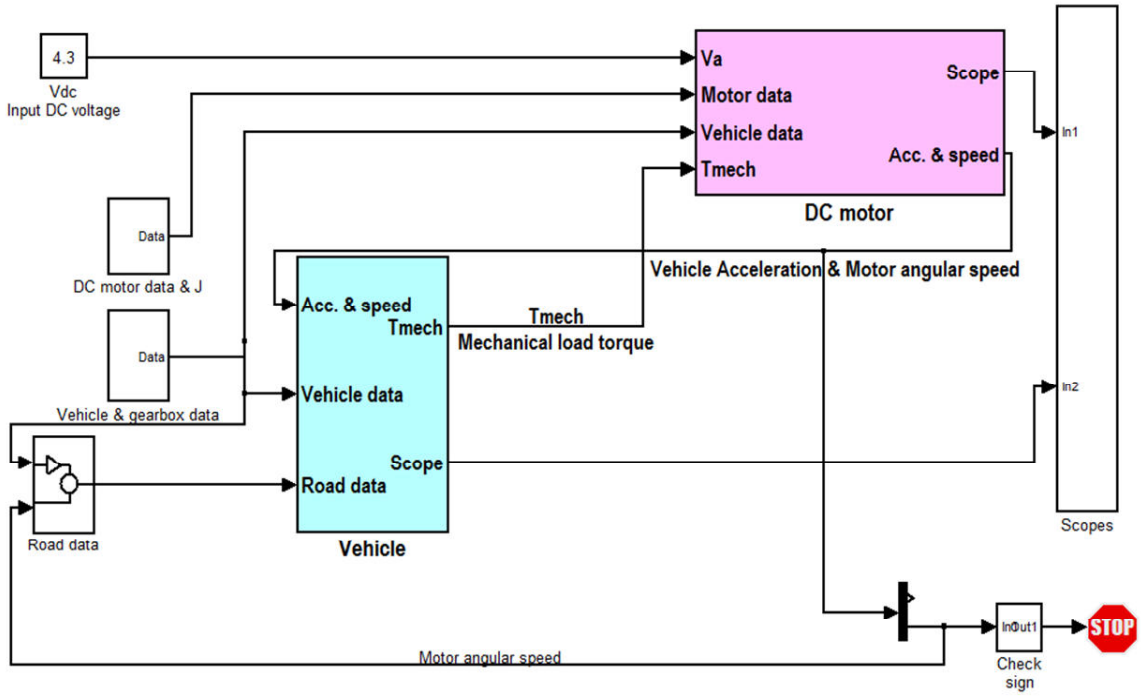


Figure 2-3: SIMULINK model of a car driven by 2 PM DC electric motors without a controller.

38 DC motor datasheet [35] for the DC motor parameters and Vernon NiMH battery [36].

The DC motor angular speed is fed to the car model to calculate the output mechanical power of the motor, P_{out}, where

$$P_{out} = T_{mech} . \omega \tag{2-7}$$

while the input electric power, P_{in}, is represented in the DC motor model by the following equation:

$$P_{in} = I_a . V_a \quad . \tag{2-8}$$

Table 2- 1: Car and motor parameters.
*assumed values.

Parameter	Value	Parameter	Value
Mass (m_{robot})	3.5 (curb weight) +1.5 (pay load) =5 kg (gross mass)	Rolling resistance coefficient (f)	0.02*
Wheel dynamic radius (r_d)	0.089 m	Gearbox mechanical efficiency (η_m)	85%*
Gearbox ratio ($i_{g.b}$)	1:12.5	Wheel mass moment of inertia ($J_w=\frac{m_w r_d^2}{2}$)	0.00118 kg.m^2
Winding inductance (L_a)	0.7 mH	Motor mass moment of inertia (J_m)	3.2 x 10^{-7} kg.m^2
Winding resistance (R_a)	1.5 Ω	Total mass moment of inertia expressed in motor shaft coordinates (J)	9 x 10^{-6} kg.m^2
Torque constant (K_a)	0.0167 N.m/A	Battery voltage	7.2 v

Figure 2-4-a shows the change of the road angle with time in degrees. The road angle is chosen to act as if the robot is negotiating a ditch. This means that the road is assumed to be initially level then negative slope then level for a short time then positive slope then level again. This will affect the mechanical load torque on the motors, T_{mech}, as shown in Figure 2-4-b. The input DC voltage to the armature, V_a, is assumed to be constant as shown in Figure 2-4-f.

Figure 2-4-c shows the motor angular speed. Figure 2-4-d shows the delivered mechanical power on each wheel. The normalized acceleration of the car ($\xi = \frac{1}{g}.\frac{dv}{dt}$) is shown in Figure 2-4-e, where v is the car speed.

Figure 2-4-g shows the armature winding current. Figure 2-4-h shows the input electric power for each motor, while Figure 2-4-i shows the induced e.m.f. Finally, Figure 2-4-j shows the electromagnetic torque T_{em}.

It can be noticed from the simulation results that when applying a constant DC voltage of 4.3V to the motors without any closed-loop control, the car couldn't overcome the previously described ditch. Saturation occurred during going uphill. So, the robot is stuck in the ditch. This could be seen from the motor angular speed, Figure 2-4-c, as the robot stalled. At this moment it could be seen that the input voltage and the armature current are at their maximum values, which led to higher electric power consumption, 7 Watt, and consequently higher energy consumption, 77 J, till the end of simulation time only. Also, the kinetic energy of the car gained during going downhill didn't help overcome the uphill. It is essential to mention here that there is a noticeable difference between the input electric power and the delivered mechanical power on the wheels, see Figure 2-4-d and Figure 2-4-i. During the time interval 1.8 s to 2.8 s, the delivered mechanical power is increased dramatically because of the inertia of the car due to going downhill first then going uphill. This caused a decrease in the armature current, Figure 2-4-g, which consequently decreased the delivered electric power to the motors during this time interval.

2.1.3 Constant input DC current

Instead of a constant DC input voltage, a DC current input of 135 mA, chosen for comparison reasons as it will be explained later, is fed to the armature to see how this will affect the DC motor performance.

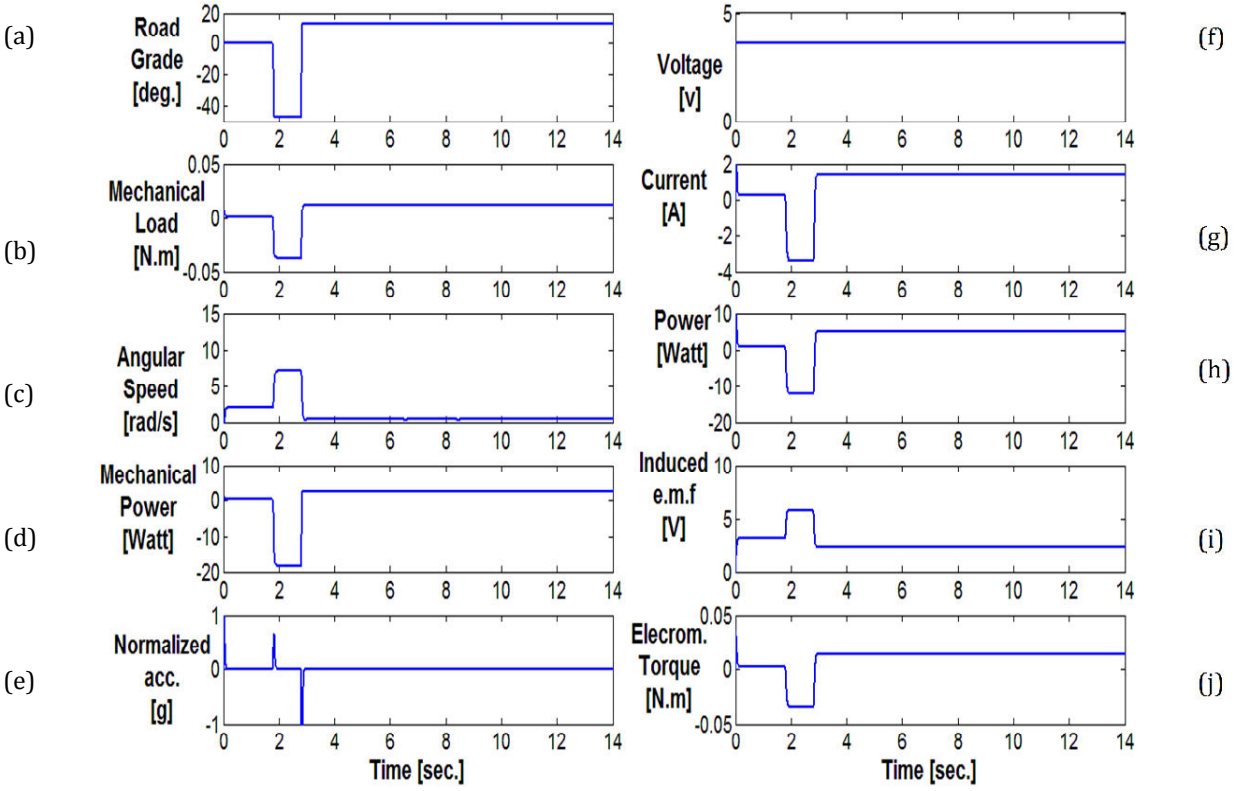


Figure 2-4: DC motor performance with open-loop control with constant DC voltage input.

(a) Road slope α, (b) Mechanical load torque T_{mech}, (c) Motor speed ω, (d) Delivered mechanical power on each wheel P_{out}, (e) Car normalized acceleration ξ, (f) Input armature voltage V_a, (g) Armature current I_a, (h) Induced e.m.f E_a, (i) Input electric power to each motor P_{in}, and (j) Electromagnetic torque T_{em}.

Figure 2-5 shows the block diagram of this model. The equations used in the vehicle and DC motor models are the same as those used in the previous

model except that in the new DC motor model a voltage limiter is added (saturation block in SIMULINK®) to maintain the limits of the voltage source as from -4.3 to 4.3V. Figure 2-6 shows the performance of the DC motor when the input armature current is constant. As it could be seen from the speed curve, Figure 2-6-c, the car is moving at a low speed until it reaches the ditch at which the car accelerates and moves downward at very high speed. From 10s to 10.8s

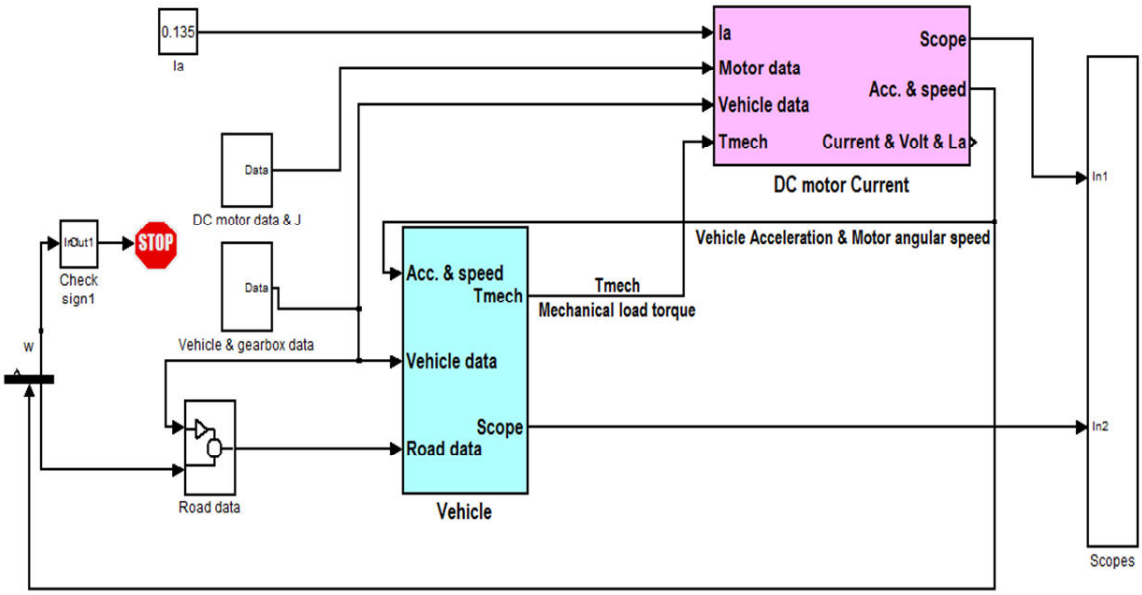


Figure 2-5: Block diagram of a DC motor with a constant armature current input.

the car speed reaches more than ten times the DC motor's rated speed. This is dangerous and unimaginable practically. In the same time, the very low speed at which the car is moving in the beginning is also not acceptable in practice. The value of the constant input DC current could be higher, but this will cause the car to reach very high speed values to the extent that when the input reaches the

rated current value the car speed will reach ten times the rated speed even before reaching the ditch. On the other hand, when the input DC current is reduced, the car speed before reaching the ditch will be very low and still we will have an unimaginable speed profile at the ditch. The choice of 135mA input DC current value is to allow the car to cover the total distance in a reasonable time, about 14s, as compared to the other control schemes that will be presented later. This choice of the low current level led to a reduced power and energy

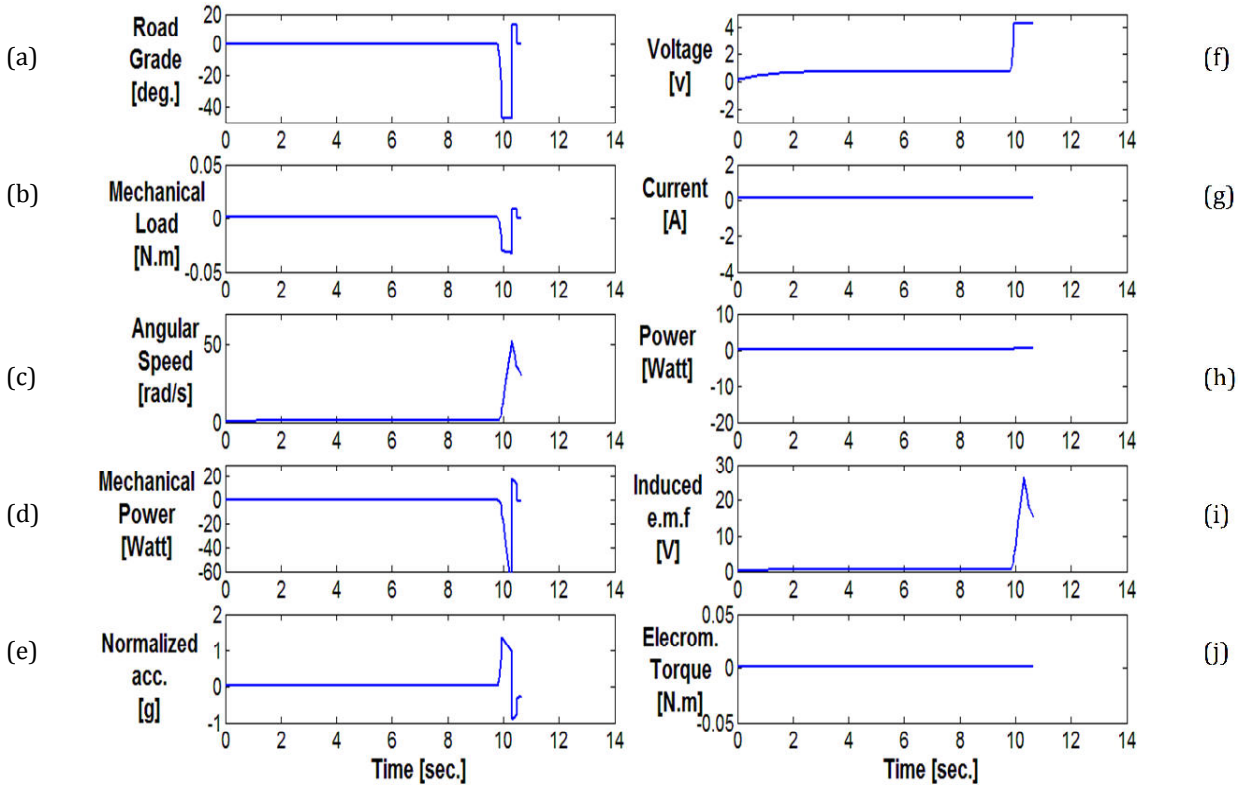


Figure 2-6: DC motor performance without a controller with constant DC current input.
(a) Road slope α, (b) Mechanical load torque T_{mech}, (c) Motor speed ω, (d) Delivered mechanical power on each wheel P_{out}, (e) Car normalized acceleration ξ, (f) Input armature voltage Va, (g) Armature current I_a, (h) Induced e.m.f E_a, (i) Input electric power to each motor P_{in}, and (j) Electromagnetic torque T_{em}.

consumption, 0.1 Watt (1.5 J), but as mentioned earlier this case is not acceptable practically.

2.2 PID speed controller

2.2.1 PID speed control by controlling the input DC voltage

A Proportional-Integral-Derivative controller and a dc-dc converter are added to each motor to control its speed in an efficient way using the Pulse Width Modulation (PWM) technique. The actuating signal in this case is the input voltage to the DC motors. Therefore, the DC motor model used here is the same as the one used in the open-loop control with DC voltage input. Figure 2-7 shows the closed loop control of the DC motor. The controller aims to maintain the motor speed at a constant value, chosen to be the rated speed of the motors, of 5.3 rad/s whatever is the slope of the road. A speed feedback is fed to the controller using the encoder equipped with the DC motor. The sensor gain is assumed to be unity. The DC - DC converter is presented here as a simple constant gain multiplied by the duty cycle fed to the converter by the controller. The speed sensor is an optical encoder represented here as a simple constant gain of one. The PID controller, Figure 2-7-b, is given by:

$$G_{PID}(s)=K_p+K_i/s+sK_d \tag{2-9}$$

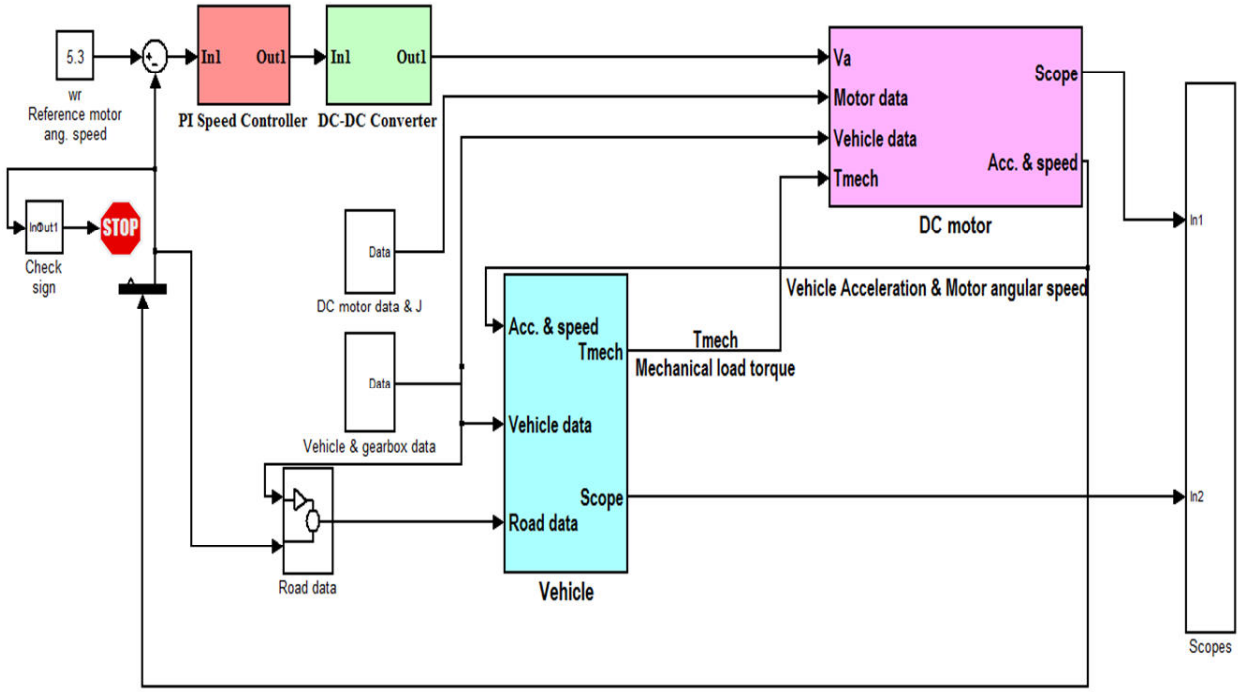
The values of the PID controller gains could not be obtained by Ziegler-Nichols methods as the system has integrators (method 1 cannot be applied) and the system also doesn't oscillate (method 2 cannot be applied). The values of the PID controller gains are obtained by trial-and-error starting with low values, 0.001, and ending with high values, 10,000, with logarithmically ascending increment. The trial-and-error method was implemented and repeated for the PID controller gains until a reasonable performance, minimum speed error, is achieved. The best performance was achieved when the gains are K_p=10, K_i=0.7, and K_d=0.09.

The output of the PID controller is fed to the DC-DC converter and considered as the required duty cycle of the voltage PWM which, in turn, is represented here by a simple gain of unity and a saturation block to limit the DC input voltage values between -4.3 and 4.3V, Figure 2-7-c.

The results of the car model with PID motor speed controller are shown in Figure 2-8. The vehicle in this case is moving straight forward and negotiates a ditch as described earlier.

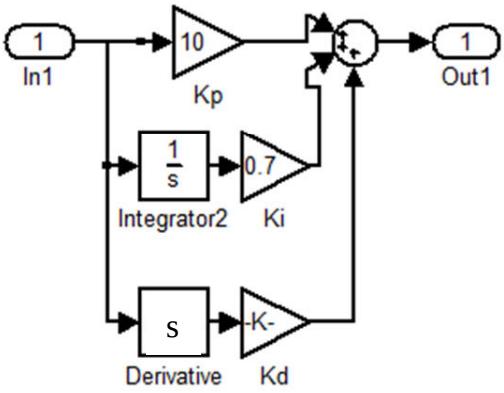
From Figure 2-8-d, it is clear that the controller achieved the desired speed, 5.3 rad/s, before entering the ditch and after crossing it. It is also noticed from Figure 2-8-h that the power and energy consumption are high, 7 Watt and 45 J, respectively.

Another benefit achieved here, by adding a PID speed controller to control the DC input voltage, which is limiting the speed during going downhill.

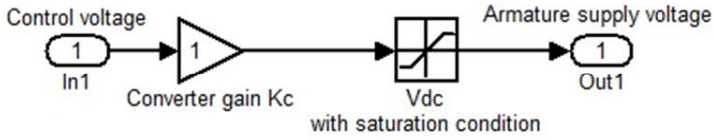


Speed feedback

(a)



(b)



(c)

Figure 2-7: Closed loop control of the DC motor speed control by controlling the voltage.

(a) Closed loop control layout, (b) The PID controller, and (c) DC-DC converter.

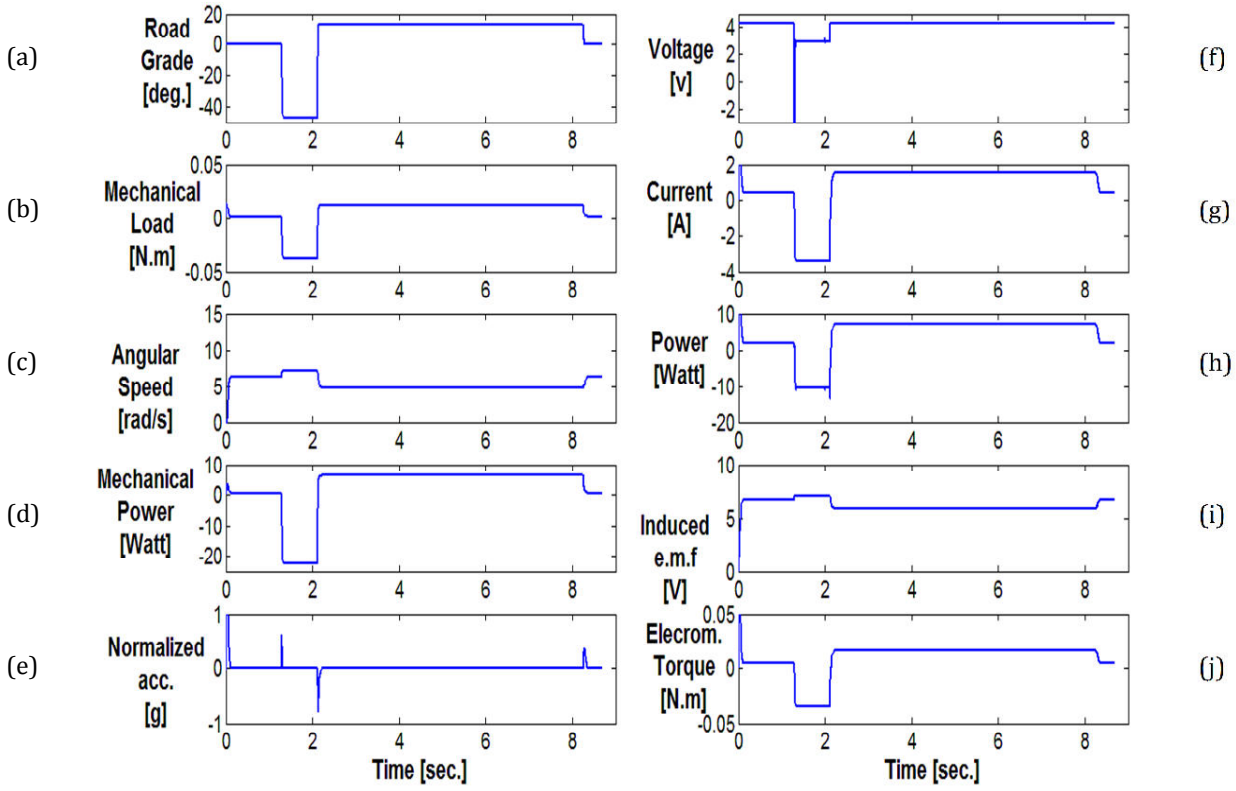
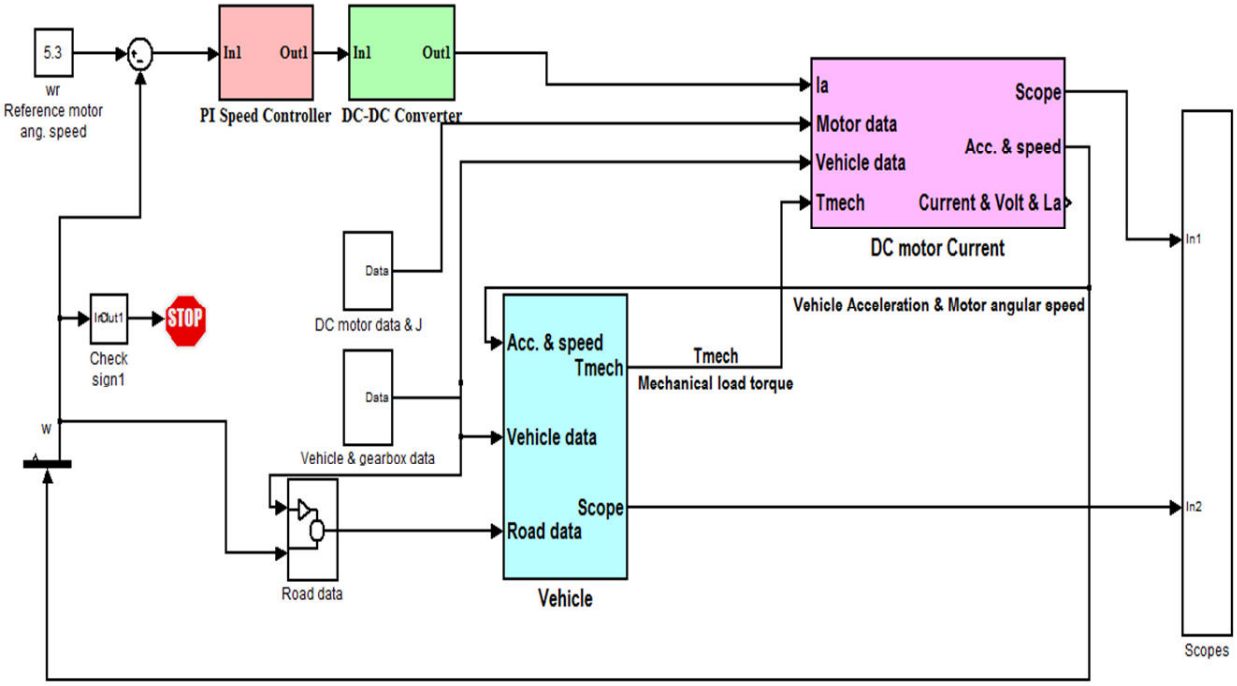


Figure 2-8: DC motor performance with a PID speed controller using voltage control input. (a) Road slope α, (b) Mechanical load torque T_{mech}, (c) Motor speed ω, (d) Delivered mechanical power on each wheel P_{out}, (e) Car normalized acceleration ξ, (f) Input armature voltage Va, (g) Armature current I_a, (h) Induced e.m.f E_a, (i) Input electric power to each motor P_{in}, and (j) Electromagnetic torque T_{em}.

2.2.2 Proportional speed control by controlling the armature current

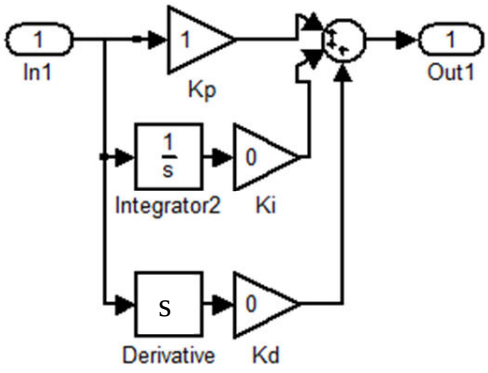
Another approach is used here which is controlling the speed using a current control command. A P-controller with unity gain is used along with a DC-DC converter to control the input armature current, I_a. The saturation block inside the DC-DC converter is set to the values -8 and 8A as that is the maximum current of the DC motor. The same equations are used as the previous vehicle and motor models except that a saturation limit for the DC voltage supply is

added in the motor's model. The block diagram is shown in Figure 2-9. The results are shown in Figure 2-10.

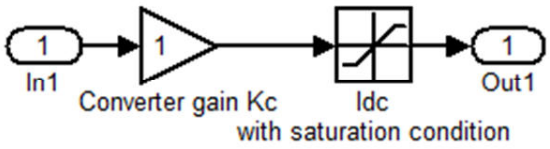


Speed feedback

(a)



(b)



(c)

Figure 2-9: Closed loop control of the DC motor speed by controlling the current.
(a) Closed loop control layout, (b) The P controller, and (c) DC-DC converter.

It can be seen from the speed response as shown in Figure 2-10-c that the controller succeeded to maintain constant speed throughout the simulation even

during going downhill. The price is the consumed electric power and energy in the interval from 5 to 9.5s which is high, 4.5 Watt and 29 J respectively, as seen in Figure 2-10-h.

The overall performance of the vehicle with the P speed controller of the current is satisfactory from the point of view of the speed as well as reasonable energy consumption. However, if the main concern is the power consumption, more improvements are needed.

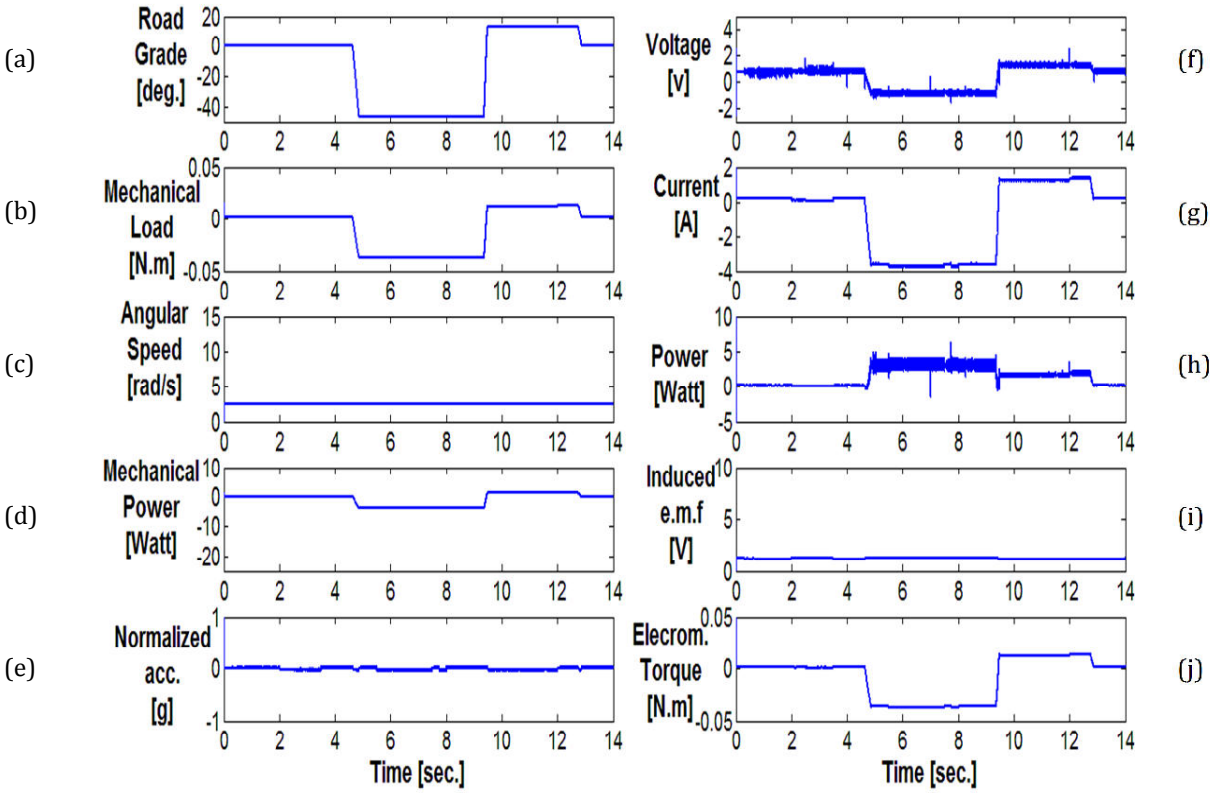


Figure 2-10: DC motor performance with a P speed controller using armature current control. (a) Road slope α, (b) Mechanical load torque T_{mech}, (c) Motor speed ω, (d) Delivered mechanical power on each wheel P_{out}, (e) Car normalized acceleration ξ, (f) Input armature voltage Va, (g) Armature current I_a, (h) Induced e.m.f E_a, (i) Input electric power to each motor P_{in}, and (j) Electromagnetic torque T_{em}.

2.3 Predictive Control

All physical systems have constraints: physical constraints, e.g. actuators constraints; safety constraints, e.g. temperature/ pressure limits; and performance constraints, e.g. on overshoot. Also, optimal operating points are often near constraints. The predictive control had a significant impact on the industrial control engineering. It has so far been applied more often in petrochemical, paper, and pulp industries. It was limited for a while to those sectors because of its demand for significant mathematical computation required during operation. The sample time in those industries ranges from one second to hours. But at this time, since the computation hardware significantly improved, there is a noticeable expansion of the use of predictive control in faster applications like automotive traction and engine control, aerospace applications, and autonomous vehicles.

Figure 2-11 shows the basic idea about the predictive controller and the receding horizon idea as presented in [37]. At each time, a predictive controller takes a measurement of the system state/output, Figure 2-11-a. Secondly; it computes the finite horizon control sequence (a finite number of future samples ahead) that uses an internal model to predict system behaviour. Then minimizes some cost function, and verifies that it does not violate any constraints, Figure 2-11-b. Then, it implements the first step of the optimal sequence. Finally, it moves

to the next time sample and repeats all the aforementioned steps, Figure 2-11-c. The horizon then moves one step forward and that's exactly where the name Receding Horizon Control comes from.

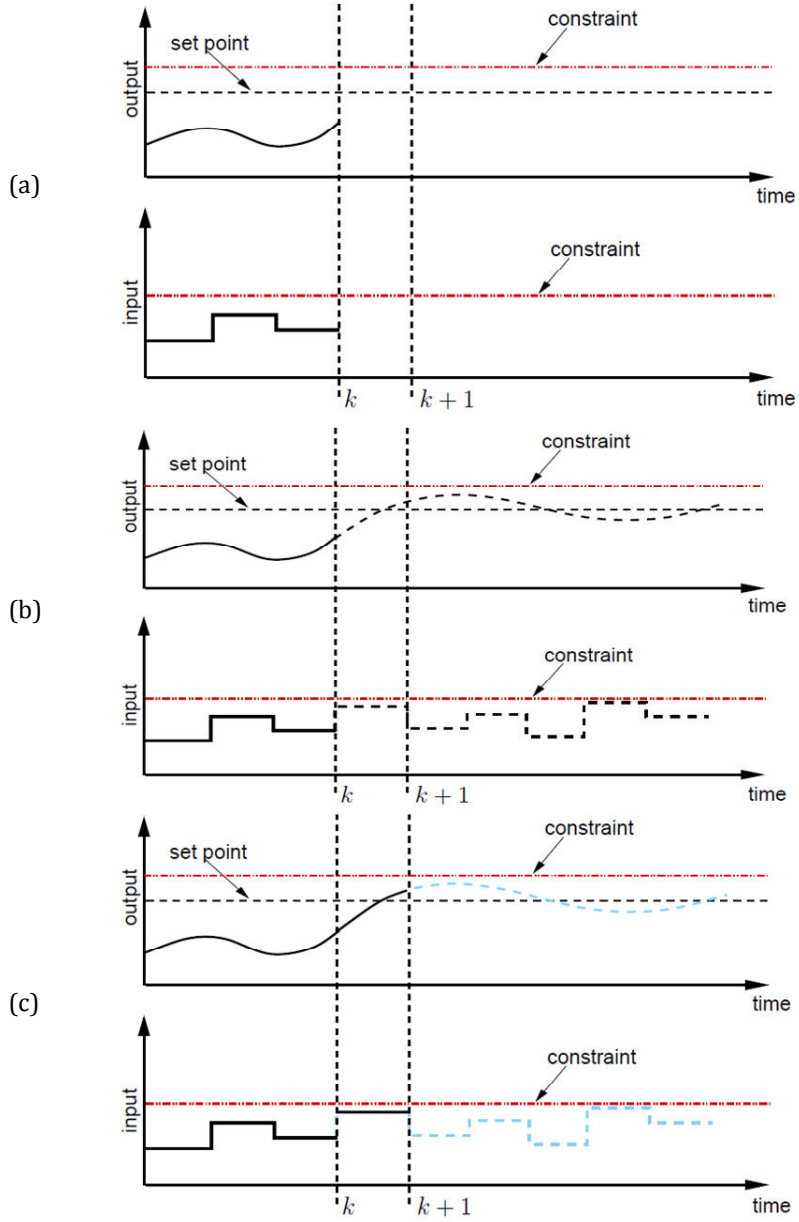


Figure 2-11: Predictive control idea.

(a) Taking measurement of the system state/output, (b) Computing a finite horizon control sequence, and (c) Implementing the first part of the optimal sequence and computing the next finite control sequence [37].

2.3.1 Dynamic Matrix Control

The predictive control has other names in the industry and academia: Dynamic Matrix Control (DMC), and Generalized Predictive Control (GPC). Also, it has more generic names: Model Predictive Control (MPC), Model Based Predictive Control (MBPC), and Receding Horizon Control (RHC). The Dynamic Matrix Control (DMC) is the one that will be discussed in this subsection due to its simplicity [31]. Then a more general definition, the Model Predictive Control, will be discussed in the next subsection as it is the base to the MATLAB MPC Toolbox™ [38].

To explain the DMC algorithm a discrete time state space description of a *n1*-th order Multi-Input-Multi-Output (MIMO) system with r inputs and q outputs is considered

$$x_{k+1}=Ax_k+Bu_k$$

$$y_k=Cx_k \tag{2-10}$$

where, x_{k+1} is the state vector at time $k+1$, x_k is the state vector at time k , u_k is the input vector at time k , y_k is the output vector at time k , A, B, and C are the system discrete state space matrices. The DMC can be obtained by minimizing the a quadratic cost function

$$J_k = \sum_{i=1}^{M} e_{k+i}^T Q_i e_{k+i} + \sum_{i=1}^{N} u_{k+i}^T R_i u_{k+1} \tag{2-11}$$

where,

$e_{k+i}=w_{k+i}-y_{k+i}$ future tracking error vector,

w_{k+i} future reference trajectory vector,

y_{k+i} future plant output vector,

u_{k+i} future control signal vector,

Q_i, R_i nonnegative diagonal weighting matrices,

M optimization horizon,

N control horizon.

The $q \times q$ and $r \times r$ weighting matrices Q_i and R_i can contain different weighting factors for each input and output. Using the state space description in eq. (2-10) future values of the output vector, y, can be calculated as

$$\underbrace{\begin{bmatrix} y_{k+1} \\ y_{k+2} \\ \cdot \\ \cdot \\ \cdot \\ y_{k+M} \end{bmatrix}}_{Y} = \underbrace{\begin{bmatrix} CA \\ CA^2 \\ \cdot \\ \cdot \\ \cdot \\ CA^M \end{bmatrix}}_{T} x_k + \underbrace{\begin{bmatrix} CB & 0 & . & . & . & 0 \\ CAB & CB & & & & 0 \\ \cdot & \cdot & & & & \cdot \\ \cdot & \cdot & & & & \cdot \\ \cdot & \cdot & & & & \cdot \\ CA^{M-1}B & . & . & . & CA^{M-N}B & 0 \end{bmatrix}}_{S} \underbrace{\begin{bmatrix} u_{k+1} \\ u_{k+2} \\ \cdot \\ \cdot \\ \cdot \\ u_{k+N} \end{bmatrix}}_{U} \tag{2-12}$$

Depending on the actual state vector x_k and the future control signals u_{k+I}, the composite vector Y contains the future output vectors, while the matrix T describes the free motion of the MIMO system in eq. (2-10). Eq. (2-11) can be rewritten as

$$J_k = (W - Y)^T \hat{Q}(W - Y) + U^T \hat{R} U \tag{2-13}$$

where the composite vector W contains the future reference vectors. The composite weighting matrices can be calculated as

$$\hat{Q} = \begin{bmatrix} Q_1 & 0 & . & . & . & 0 \\ 0 & Q_2 & & & & . \\ . & & . & & & . \\ . & & & . & & . \\ . & & & & . & 0 \\ 0 & . & . & . & 0 & Q_M \end{bmatrix}, \quad \hat{R} = \begin{bmatrix} R_1 & 0 & . & . & . & 0 \\ 0 & R_2 & & & & . \\ . & & . & & & . \\ . & & & . & & . \\ . & & & & . & 0 \\ 0 & . & . & . & 0 & R_N \end{bmatrix} \tag{2-14}$$

The optimal composite control vector U can be obtained by minimizing the cost function with respect to U.

$$U = (\hat{R} + S^T \hat{Q} S)^{-1} S^T \hat{Q}(W - T x_k) \tag{2-15}$$

Only the first vector u_k is going to be fed to the MIMO plant.

2.3.2 Model Predictive Control

Since the system is multi-input multi-output system, the following analysis will be concerned with the MIMO systems. The linear model used in the Model Predictive Control Toolbox™ software is illustrated in Figure 2- 12. The MPC controller consists of two models: a plant model and a model regarding the unmeasured disturbances.

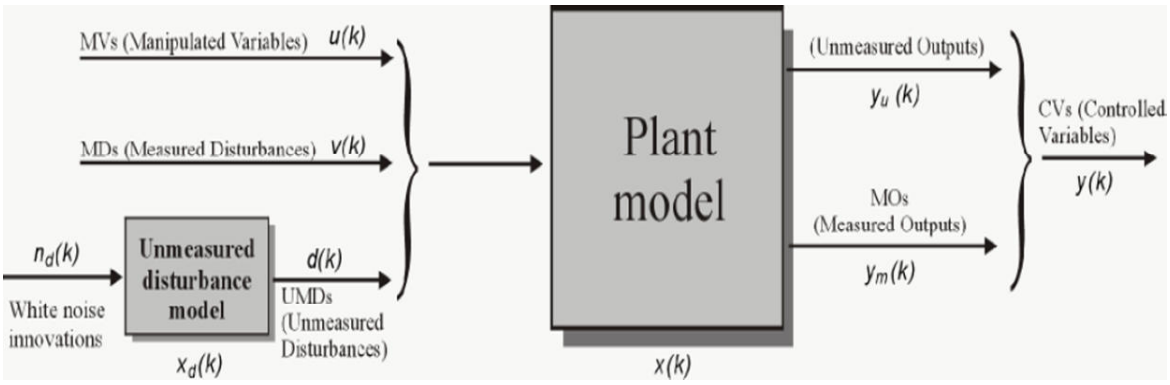


Figure 2- 12: Model Predictive Control block diagram [38].

The plant model is a linear time-invariant system described by the following equations [38]:

$$x(k+1) = Ax(k) + B_u u(k) + B_v v(k) + B_d d(k)$$

$$y_m(k) = C_m x(k) + D_{vm} v(k) + D_{dm} d(k) \quad (2\text{-}16)$$

$$y_u(k) = C_u x(k) + D_{vu} v(k) + D_{du} d(k)$$

where,

$u(k)$ is the manipulated variable (control variable)

$v(k)$ is the measured disturbance

$d(k)$ is the unmeasured disturbance

$y_m(k)$ is the measured output

$y_u(k)$ is the unmeasured output

$B_u(k)$ is the input matrix

$B_v(k)$ is the measured disturbance matrix

$B_d(k)$ is the unmeasured disturbance matrix

C_m is the measured output matrix

D_{vm} is the measured disturbance feedforward to the measured output matrix

D_{dm} is the unmeasured disturbance feedforward to the measured output matrix

C_u is the unmeasured output matrix

D_{vu} is the measured disturbance feedforward to the unmeasured output matrix

D_{du} is the unmeasured disturbance feedforward to the unmeasured output matrix

The cost function is

$$J = \sum_{i=0}^{p-1} \left(\sum_{j=1}^{n_y} \left| w_{i+1,j}^{y} \left(y_j(k+i+1|k) - r_j(k+i+1) \right) \right|^2 + \sum_{j=1}^{n_u} \left| w_{i,j}^{\Delta u} \Delta u_j(k+i|k) \right|^2 + \sum_{j=1}^{n_u} \left| w_{i,j}^{u} \left(u(k+i|k) - u_{jtarget}(k+i) \right) \right|^2 \right) + \rho_\varepsilon \varepsilon^2 \tag{2-17}$$

where, the subscript j denotes the component of a vector and $(k+i|k)$ denotes the value predicted at time $k+i$ based on the information available at time k,

$r(k)$ is the current sample of the output reference,

Δu_j is the increment for the j th input variable,

$w_{i,j}^{y}$,$w_{i,j}^{u}$, and $w_{i,j}^{\Delta u}$ are the nonnegative weights for the outputs, inputs and increment inputs respectively,

n_y and n_u are the number of output and input variables respectively,

p is the prediction horizon,

$u_{jtarget}$ is the set-point for the input variable,

ε is the slack variable (for softening or relaxing the constraints),

and ρ_ε is the weight of the slack variable.

The cost function is subject to the following constraints

$$u_{jmin}(i) - \varepsilon V_{jmin}^{u}(i) \leq u_j(k+i|k) \leq u_{jmax}(i) + \varepsilon V_{jmax}^{u}(i)$$

$$\Delta u_{jmin}(i) - \varepsilon V_{jmin}^{\Delta u}(i) \leq \Delta u_j(k+i|k) \leq \Delta u_{jmax}(i) + \varepsilon V_{jmax}^{\Delta u}(i)$$

$$y_{jmin}(i) - \varepsilon V_{jmin}^{y}(i) \leq y_j(k+i+1|k) \leq y_{jmax}(i) + \varepsilon V_{jmax}^{y}(i)$$

$$\Delta u(k+h|k) = 0$$

$$\varepsilon \geq 0$$

where, *i=0,…,p-1*

h=m,…,p-1 (m denotes the control horizon),

$u_{jmin}, y_{jmin}, and\ \Delta u_{jmin}$ are the minimum values of the input, output and increment input, respectively,

$u_{jmax}, y_{jmax}, and\ \Delta u_{jmax}$ are the maximum values of the input, output and increment input, respectively,

$V_{min}^{u}, V_{min}^{y}, V_{min}^{\Delta u}, V_{max}^{u}, V_{max}^{y}, and\ V_{max}^{\Delta u}$ are the Equal Concern for the Relaxation (ECR) vectors which have nonnegative entries that quantify the concern for relaxing the corresponding constraints; the larger *V*, the softer the constraint.

As mentioned earlier, only $\Delta u(k|k)$ is actually used to generate $u(k)$. The remaining values $\Delta u(k+i|k)$ are discarded, and a new optimization problem based on $y_m(k+1)$ is solved at the next sampling step *k*+1.

2.3.2.1 State Estimation

In the design of the model predictive controller, it is assumed that the information $x(k+i)$ is available at time *k+i*. This assumes that all the state variables are measurable. In most of the real applications, not all state variables are measurable. Some of them may be impossible to measure. One approach is to estimate the state variable from the from the process measurement. The "soft" instrument that is used to estimate the unknown state variables from measurements, in a control engineering context, is called an observer. The concept of an observer has been widely used in the science and engineering fields.

An observer is constructed from the mathematical model of the plant itself, Figure 2- 13. So, if the state space model of the plant is

$$x(k+1) = Ax(k) + Bu(k) \tag{2-18}$$

then this model can be used to calculate the state vector $\hat{x}(k), k = 1,2, \ldots,$ with an initial state condition $\hat{x}(0)$ and input signal $u(k)$ as

$$\hat{x}(k+1) = A\hat{x}(k) + Bu(k) \tag{2-19}$$

This approach would work after some transient time if the plant model is stable and our guess of the initial conditions is nearly correct. The main drawback of this estimation is that it is actually considered an open-loop prediction. The error in estimating the state is $\tilde{x}(k) = x(k) - \hat{x}(k)$. So, for a given initial state error $\tilde{x}(0) \neq 0$, we have

$$\tilde{x}(k) = A^k \tilde{x}(0). \tag{2-20}$$

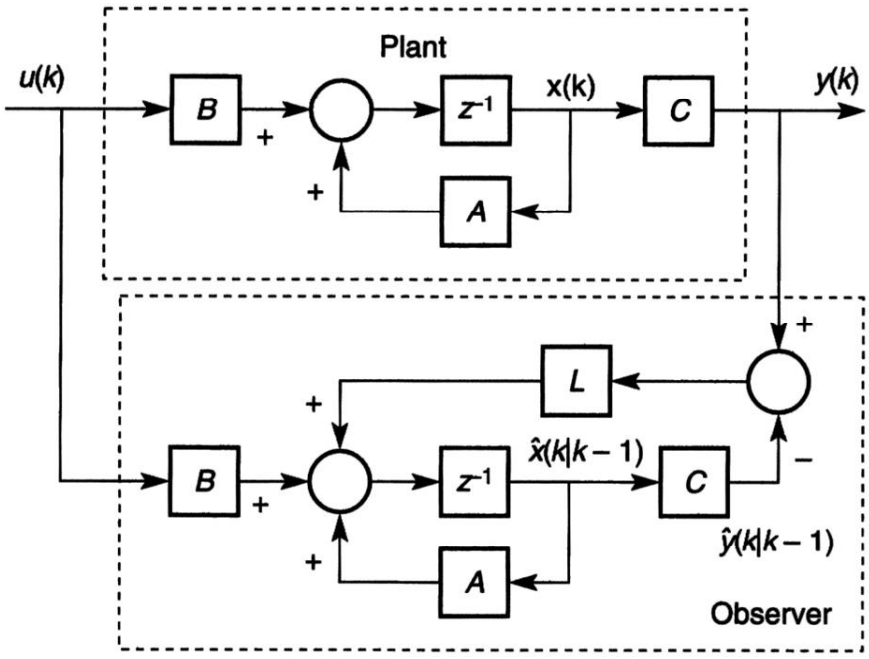


Figure 2- 13: A state observer is a copy of the plant (z is the z-transform operator).

Two points need to be discussed here. First is the stability of the prediction. If A has all eigenvalues inside the unit circle, then the error system is stable which means that the estimated variable $\hat{x}(k)$ converges to *x(k)*. However, if A has one or more eigenvalues outside the unit circle, then the error system is unstable which means that the prediction $\hat{x}(k)$ doesn't converge to *x(k)*. The second point to discuss is the rate of convergence of the prediction. That is in case of having a stable plant model A, we have no "control" on the convergence rate of error. Which means if the plant model has poles close to the origin of the complex plane, then the error converges at a fast rate to zero; otherwise, the convergence rate could be slow.

To improve the estimate of *x(k)*, a feedback principle is used where an error signal is deployed. The observer could be represented using the equation [39]:

$$\hat{x}(k+1) = A\hat{x}(k) + Bu(k) + K_{ob}(y(k - C\hat{x}(k)), \qquad (2\text{-}21)$$

where, K_{ob} is the observer gain matrix. The last term, $K_{ob}(y(k) - C\hat{x}(k))$, is the correction term in the observer.

To choose the observer gain matrix, K_{ob}, the closed-loop equation is examined. By substituting $y(k) = Cx(k)$ into eq. (2-21), with the definition of the error state, $\tilde{x}(k) = x(k) - \hat{x}(k)$, the following equation is obtained

$$\tilde{x}(k+1) = A\tilde{x}(k) - K_{ob}C\tilde{x}(k)$$

$$= (A - K_{ob}C)\tilde{x}(k). \quad (2\text{-}22)$$

If we have an initial error of $\tilde{x}(0)$, then

$$\tilde{x}(k+1) = (A - K_{ob}C)^k\tilde{x}(0). \quad (2\text{-}23)$$

It is apparent from the comparison between the error given by eq. (2-23) and the error given by the eq. (2-20) that the observer gain K_{ob} can be used to manipulate the conversion rate of the error.

2.3.2.2 Alternative state vector

Our plant model expresses the plant state *x* in terms of the values of the input *u* [40]. The cost function, however, penalizes changes in the input, Δu, rather than the input values themselves. The predictive control will actually produce changes Δu rather than *u*. It is therefore convenient for many purposes to regard the "controller" as producing the signal Δu, and the "plant" as having this signal as its input. That is, it is often convenient to regard the "discrete-time integration" from Δu to *u* as being included in the plant dynamics. Figure 2- 14 shows the real controller producing the signal *u*, which is passed to the real plant. It also shows the MPC "controller" producing the signal Δu, which is passed to the MPC "plant". One of the ways to include this "integration" in a state-space model of the MPC "plant" is to involve augmenting the state vector.

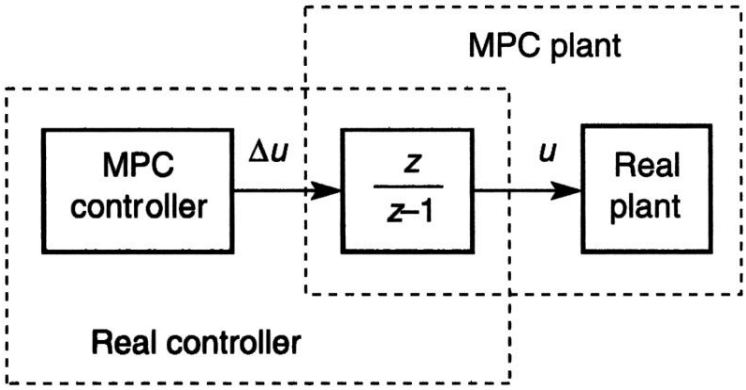


Figure 2- 14: The boundary between controller and plant (z is the z-transform operator).

Recalling the state-space model presented earlier

$$x(k+1) = Ax(k) + Bu(k) \tag{2-24}$$

$$y(k) = Cx(k), \tag{2-25}$$

the following difference equation is presented

$$x(k) = Ax(k-1) + Bu(k-1). \tag{2-26}$$

By defining $\Delta x(k) = x(k) - x(k-1)$ and $\Delta u(k) = u(k) - u(k-1)$, then subtracting eq. (2-26) from eq. (2-24) leads to

$$\Delta x(k+1) = A\Delta x(k) + B\Delta u(k). \tag{2-27}$$

In order to relate the output *y(k)* to the state variable $\Delta x(k)$, we deduce that

$$\Delta y(k+1) = Cx(k+1) = CAx(k) + CBu(k),$$

where $\Delta y(k+1) = y(k+1) - y(k)$.

Choosing a new state variable vector $x(k) = [\Delta x(k)^T \quad y(k)^T]^T$, we have:

$$\begin{bmatrix} \Delta x(k+1) \\ y(k+1) \end{bmatrix} = \begin{bmatrix} A & O_{n_y}^T \\ CA & I_{n_y \times n_y} \end{bmatrix} \begin{bmatrix} \Delta x(k) \\ y(k) \end{bmatrix} + \begin{bmatrix} B \\ CB \end{bmatrix} \Delta u(k)$$

$$y(k) = \begin{bmatrix} O_{n_u} & I_{n_y \times n_y} \end{bmatrix} \begin{bmatrix} \Delta x(k) \\ y(k) \end{bmatrix}, \tag{2-28}$$

where, $I_{n_y \times n_y}$ is the identity matrix with dimensions $n_y \times n_y$, which is the number of outputs; and $O_{n_y}$is a $n_y \times n1$ zero matrix, where *n1* is the number of original state variables. In eq. (2-28), A, B and C have the dimension $n1 \times n1$, $n1 \times n_u$ and $n_y \times n1$, respectively.

For notation simplicity, we denote eq. (2-28) by

$$x(k+1) = Ax(k) + B\Delta u(k)$$

$$y(k) = Cx(k), \tag{2-29}$$

where, A, B and C are matrices corresponding to the forms given in eq. (2-28). In what follows, the dimensionality of the augmented state-space equation is taken to be n $(= n1 + n_y)$.

2.3.2.3 Controllability and observability of the augmented model

Because the original plant model is augmented with integrators and the MPC design is performed on the basis of the augmented state-space model, it is important for control system design that the augmented model does not become uncontrollable or unobservable, particularly with respect to the unstable dynamics of the system. Controllability is a pre-requisite for the predictive control system to achieve the desired closed-loop control performance and observability is a pre-requisite for a successful design of an observer. However, the conditions may be relaxed to the requirement of stabilizability and detectability, if only closed-loop stability is of concern.

The simplest way for the investigation is based on the assumption of minimal realization of the plant model. The discussion of minimal realization, controllability and observability can be found in [41], [42]. The definition of the realization states that; A realization of transfer function *G(z)* is any state-space triplet (A, B and C) such that $\mathrm{G(z)} = \mathrm{C(zI - A)^{-1}B}$. If such a set ($A$, B and C) exists, then *G(z)* is said to be realizable. A realization (A, B and C) is called a minimal realization of a transfer function if no other realization of smaller dimension of the triplet exists.

A minimal realization has the distinctive feature summarized in the theorem below.

Theorem 2.1 A minimal realization is both controllable and observable, [41], [42].

With this theorem, it is possible is to prove the controllability and the observability of the augmented state-space model through the argument of the minimal realization.

Another approach is to proof the observability and the controllability separately. For observability; a state variable model of a dynamic system is said to be completely observable if, for any sample time *k0*, there exists a sample time *k1 > k0* such that a knowledge of the output *y(k)* and input *u(k)* in the time interval $k0 \leq k \leq k1$ is sufficient to determine the initial state *x(k0)* and as a consequence, *x(k)*, for all *k* between *k0* and *k1*. A necessary and sufficient condition for a linear discrete-time system to be completely observable is, if the observability matrix, L_0, has rank *n*, where *n* is the dimension of the state variable model.

$$L_0 = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix}$$

Depending on the context, the controllability could be interpreted as state controllability or output controllability. Here we are concerned with the state

controllability. The complete state controllability (or simply controllability if no other context is given) describes the ability of an external input to move the internal state of a system from any initial state to any other final state in a finite time interval [43]. The system is said to be controllable if the controllability matrix, L_c, has full rank *n,* where

$$L_c = [B \quad AB \quad A^2B \quad \dots \quad A^{n-1}B].$$

2.3.2.4 Kalman filter

If the pair (A, C) is observable, a pole assignment strategy can be used to determine K_{ob} such that the eigenvalues of the observer (*i.e.* of the matrix $A - K_{ob}C$) are at desired locations as mentioned earlier. For a multi-output system, K_{ob} can be calculated recursively using a Kalman filter. Kalman filters are proposed in a stochastic setting. In addition, it is implicitly used in the MPC Toolbox™. To this end, we assume that

$$x(k+1) = Ax(k) + Bu(k) + w(k)$$

$$y(k) = Cx(k) + \epsilon(k), \tag{2-30}$$

where, w and ϵ are vectors of random variables with the covariance matrices defined by, respectively

$$E\{w(k)w(k)^T\} = \Theta\delta(k-\tau)$$

$$E\{\in(k) \in (k)^T\} = \Gamma\delta(k-\tau)$$

where, $\delta(k-\tau) = 1$, if $k = \tau$ and $\delta(k-\tau) = 0$ if $k \neq \tau$.

The optimal observer gain K_{ob} is solved recursively for *i = 0, 1, . . .*, using

$$K_{ob}(i) = AP(i)C^T(\Gamma + CP(i)C^T)^{-1}, \tag{2-31}$$

and

$$P(i+1) = A\{P(i) - P(i)C^T(\Gamma + CP(i)C^T)^{-1}CP(i)\}A^T + \Theta. \tag{2-32}$$

More specifically, *P(0)* satisfies

$$E\{[x(0) - \hat{x}(0)][x(0) - \hat{x}(0)]^T\} = P(0).$$

Assuming that the system (C , A) is detectable from the output *y(k)* (i.e., there are no unstable states whose response cannot be 'seen' from the output) and (A , $\Theta^{1/2}$) is stabilizable, then, as $k \to \infty$, the steady-state solutions of eq. (2-31) and eq. (2-32) satisfy the discrete-time algebraic Riccati equation:

$$P(\infty) = A\{P(\infty) - P(\infty)C^T(\Gamma + CP(\infty)C^T)^{-1}CP(\infty)\}A^T + \Theta, \tag{2-33}$$

$$K_{ob}(\infty) = AP(\infty)C^T(\Gamma + CP(\infty)C^T)^{-1}. \tag{2-34}$$

Also, the eigenvalues of $A - K_{ob}(\infty)C$ are guaranteed to be inside the unit circle (i.e. stable). To avoid confusion, it is emphasized that the iterative solution of the Riccati eq. (2-32) is not required in real time. The observer gain is calculated off-line for predictive control applications.

For more details on Kalman filters refer to [44] and [45].

2.3.2.5 Tuning observer dynamics

It is often the case that the covariance matrices Θ and Γ, corresponding to the characteristics of the disturbances, are unknown. Thus, in practice, we

choose Θ, Γ and an initial *P(0)* to calculate an observer gain K_{ob} by solving the Riccati equation iteratively until the solution converges to a constant matrix. Then, the closed-loop system obtained is analyzed with respect to the location of eigenvalues contained in $A- K_{ob}C$, the transient response of the observer, robustness and effect of noise on the response. The elements of the covariance matrices are modified until a desired result is obtained. Such a trial-and-error procedure can be time consuming and frustrating, and is one of the challenges faced when using Kalman-filter-based multivariable system design. In some circumstances, however, it is possible to specify a region in which the closed-loop observer error system poles should reside and to enforce this in the solution. A simple approach is followed here along similar lines to the classic approach in [39] and [44], in which the closed-loop observer poles are assigned inside a circle with a pre-specified radius α ($0 < \alpha < 1$). The procedure is summarized as follows. If the observer error system is:

$$\tilde{x}(k+1) = (A - K_{ob}C)\tilde{x}(k). \tag{2-35}$$

The transformation is performed so that $\hat{A} = \frac{A}{\alpha}$ and $\hat{C} = \frac{C}{\alpha}$ where $0<\alpha<1$, leading to a transformed system:

$$\tilde{x}_t(k+1) = \frac{1}{\alpha}\left(A - \hat{K}_{ob}C\right)\tilde{x}_t(k) = \left(\hat{A} - \hat{K}_{ob}\hat{C}\right)\tilde{x}_t(k). \tag{2-36}$$

Solving the iterative eqns. (2-31) and (2-32), or the steady-state Riccati eq. (2-33) by using $\hat{A}$ and $\hat{C}$ to replace A and C matrices, and then the eigenvalues of $\hat{A} - \widehat{K}_{ob}(\infty)\hat{C}$ are guaranteed to be inside the unit circle (*i.e.*, stable). The resultant observer gain $\widehat{K}_{ob}$ is then applied to the original observer system eq. (2-35), leading to the closed-loop characteristic equation:

$$det\left(zI - \left(A - \widehat{K}_{ob}C\right)\right) = det\left(zI - \left(\hat{A} - \widehat{K}_{ob}\hat{C}\right) \times \alpha\right) = 0. \qquad (2\text{-}37)$$

Therefore, the eigenvalues of $A - \widehat{K}_{ob}C$ are equal to the eigenvalues of $\hat{A} - \widehat{K}_{ob}\hat{C}$ multiplied by the factor α, which guarantees that the eigenvalues of the observer error system with $\widehat{K}_{ob}$ reside inside the circle of radius α. This procedure makes a direct connection to the observer dynamics via the choice of α. The trial-and-error procedure can be reduced to choose a suitable α along with Θ and Γ to achieve the desired closed-loop performance.

2.3.3 Model Predictive Control of the vehicle using the DC input voltage as the manipulated variable

Eq. (2-1) and eq. (2-2) are used to derive a continuous state space model of the DC motor. This continuous state-space model is going to be used inside the MPC controller to predict the behaviour of the plant. When using MATLAB® MPC Toolbox™, it is required to derive a continuous state-space model of the system and fed it to the toolbox as observer. The toolbox also gives the ability to

add noise to the prediction as well as measurement noise. All the noise added to the system in the current study is white noise with identity autocorrelation matrix. The MPC Toolbox® does all the optimization and calculations for us. The state space model developed here includes the state vector as the motor angular speed, ω, and the armature current I_a. The inputs to the system are the armature DC voltage supply, V_a, as a manipulated variable and the mechanical load torque, T_{mech} , as a measured disturbance. Another input is added here which is the voltage drop on the brushes, V_{brush}, but it is assumed to be a constant unmeasured disturbance. The outputs are assumed to be as the state vector and both states are assumed to be measurable. The continuous time state space model for the system is

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} \dot{\omega} \\ \dot{I}_a \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{-b}{J} & \frac{K_a}{J} \\ \frac{-K_a}{L_a} & \frac{-R_a}{L_a} \end{bmatrix}}_{A_C} \begin{bmatrix} \omega \\ I_a \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & \frac{-1}{J} & 0 \\ \frac{1}{L_a} & 0 & \frac{-1}{L_a} \end{bmatrix}}_{B_C} \begin{bmatrix} V_a \\ T_{mech} \\ V_{brush} \end{bmatrix} \tag{2-38}$$

$$\begin{bmatrix} y \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}_{C_C} \begin{bmatrix} \omega \\ I_a \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{D_C} \begin{bmatrix} V_a \\ T_{mech} \\ V_{brush} \end{bmatrix} \tag{2-39}$$

The MPC controller design was chosen with a prediction horizon of 400 time steps, a control horizon of 400 time steps, and a sample time of 0.02 s. Also, an optimization weighting factor of unity is assumed for the DC voltage supply

and 0.1 for both the speed and armature current error minimization. Those weighting factors were obtained using trial-and-error. The constraints for the voltage and speed are

$$-4.3 \leq V_a \leq 4.3V \tag{2-40}$$

$$\omega > 0 \tag{2-41}$$

The reference values are assumed to be constant and equal to the DC motor rated speed and rated current. The block diagram of the MPC control using the voltage as a manipulated variable is shown in Figure 2-15. The DC motor and the vehicle models are the same as those used before in the constant input voltage case, but the input voltage this time is fed to the DC motor by the controller. The MPC controller block is a built-in block in MATLAB™ SIMULINK™. The input load torque is the same as that was used in the previous cases. The results obtained from the simulation are shown in Figure 2-16.

It can be seen from the overall performance of the vehicle with the MPC controller when using the voltage as a manipulated variable that the vehicle becomes able to overcome the ditch and in the same time it has a reduced consumed electric energy compared with the PID controller with the voltage control. The electric power and energy consumption is also reduced, 3 Watt and 16.5 J, respectively. This is mainly because the controller tries to minimize the

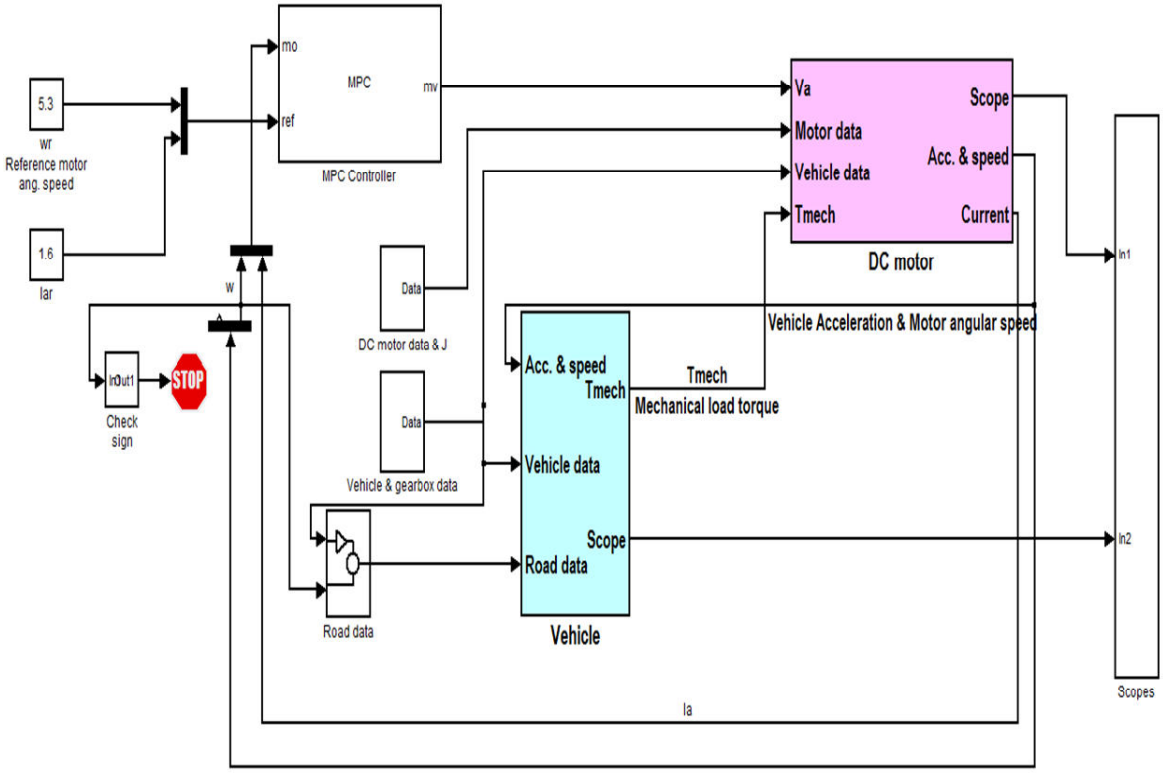


Figure 2-15: Block diagram of MPC control using the voltage as a manipulated variable.

voltage rather than to maintain a constant speed. Both the voltage and the current are reduced compared to the PID controller case.

2.3.4 Model Predictive Control using the armature current as a manipulated variable

In order to make the armature current a manipulated variable, the continuous time state space model needs to be changed. Eq. (2-1) & (2-2) could be rewritten as

$$V_a = \frac{1}{1-0.02}\left[L_a \frac{dI_a}{dt} + R_a I_a + K_a \omega \right] \tag{2-42}$$

$$\frac{d\omega}{dt} = \frac{1}{J}\left[I_a K_a - \omega \cdot b - T_{mech}\right] \tag{2-43}$$

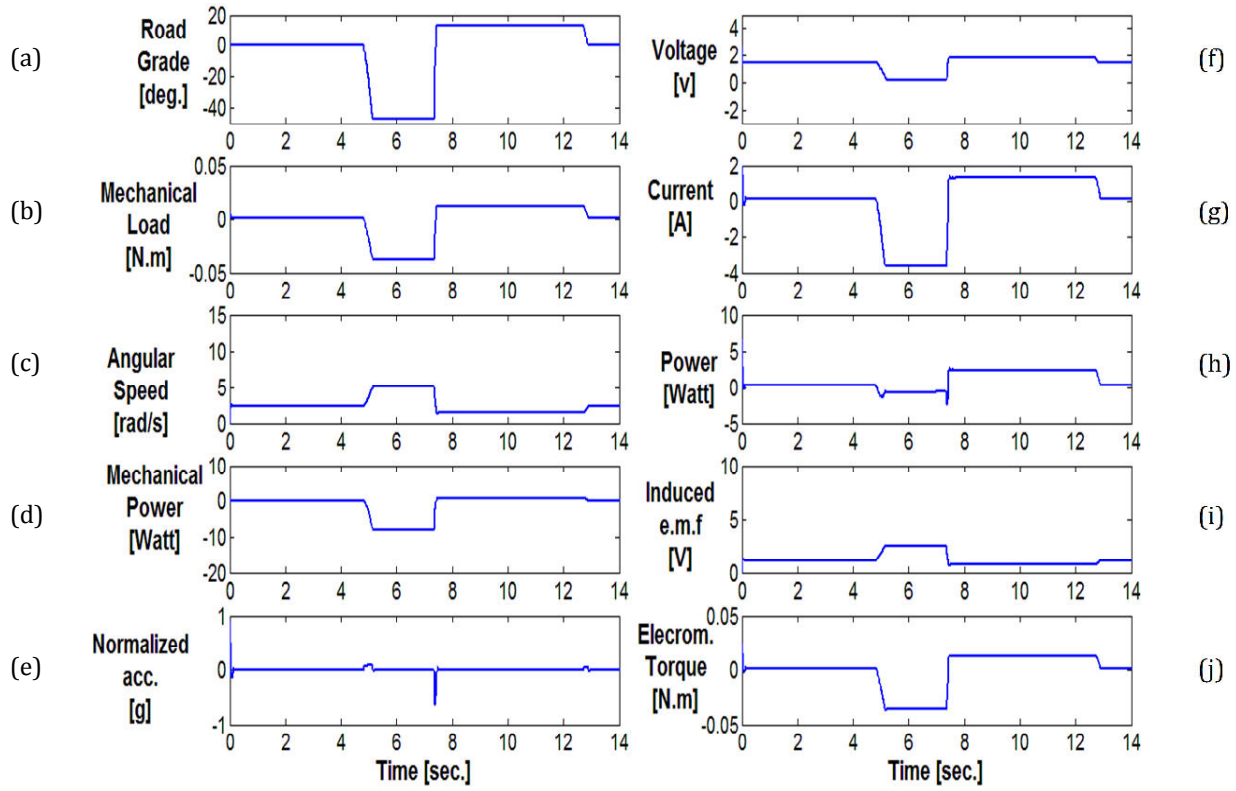


Figure 2-16: DC motor performance with an MPC controller using input voltage as a manipulated variable.

(a) Road slope α, (b) Mechanical load torque T_{mech}, (c) Motor speed ω, (d) Delivered mechanical power on each wheel P_{out}, (e) Car normalized acceleration ξ, (f) Input armature voltage Va, (g) Armature current I_a, (h) Induced e.m.f E_a, (i) Input electric power to each motor P_{in}, and (j) Electromagnetic torque T_{em}.

where the multiplier factor in the first equation arises from the assumption that the voltage drop in the brushes is 2% of the input voltage for simplicity.

To use eq. (2-42) in continuous time state space representation, a new variable is introduced, S_a , where

$$S_a = \int V_a(t)dt \tag{2-44}$$

and then eq. (2-42) becomes

$$\frac{dS_a}{dt} = V_a = \frac{1}{0.98}\left[L_a \frac{dI_a}{dt} + R_a I_a + K_a \omega \right] \tag{2-45}$$

In this case the state variables are the integration of the armature voltage, S_a, and the motor speed ω. The inputs are the current, I_a and the mechanical load torque T_{mech}. The outputs are the same as the state variables. The continuous time state space representation of the system is

$$\underbrace{\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} \frac{dS_a}{dt} \\ \frac{d\omega}{dt} \end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix} 0 & \frac{K_a}{0.98} \\ 0 & \frac{-b}{J} \end{bmatrix}}_{A_C} \underbrace{\begin{bmatrix} S_a \\ \omega \end{bmatrix}}_{x} + \underbrace{\begin{bmatrix} \frac{R_a}{0.98} & 0 \\ \frac{K_a}{J} & \frac{-1}{J} \end{bmatrix}}_{B_1} \underbrace{\begin{bmatrix} I_a \\ T_{mech} \end{bmatrix}}_{u} + \underbrace{\begin{bmatrix} \frac{L_a}{0.98} & 0 \\ 0 & 0 \end{bmatrix}}_{B_2} \underbrace{\begin{bmatrix} \dot{I}_a \\ \dot{T}_{mech} \end{bmatrix}}_{\dot{u}} \tag{2-46}$$

$$\underbrace{\begin{bmatrix} y \end{bmatrix}}_{} = \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}_{C_C} \underbrace{\begin{bmatrix} S_a \\ \omega \end{bmatrix}}_{x} + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}}_{D_C} \underbrace{\begin{bmatrix} I_a \\ T_{mech} \end{bmatrix}}_{u} \tag{2-47}$$

The fact that the system has two inputs and one of them is a derivative makes it impossible to import it to the MATLAB™ MPC toolbox the way it is. This brings a need to modify the state space representation into a suitable form that contains only the four matrices A_C, B_C, C_C and D_C. This could be done using

the approach presented by Porter & Bradshaw in [46]. A new state space representation is constructed as follows

$$\dot{\chi}(t) = A_C \chi(t) + B_C \zeta(t) \tag{2-48}$$

$$\eta(t) = \chi(t) + B_2 \zeta(t) \tag{2-49}$$

where $B_C = B_1 + A_C B_2$ (2-50)

and

$$\begin{bmatrix} \chi(t) \\ \zeta(t) \end{bmatrix} = \begin{bmatrix} I_{n1} & -B_2 \\ O_{r,n1} & I_r \end{bmatrix} \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} \tag{2-51}$$

where $n1$ is the number of the original states and r is the number of the original inputs, I_{n1} is the identity matrix of dimension $n1$ and I_r is the identity matrix of dimension r, and finally $O_{r,n1}$ is zero matrix of dimension $r \times n1$.

Applying this approach to the present system gives the new continuous time state space representation as follows:

$$\dot{\chi}(t) = \underbrace{\begin{bmatrix} 0 & \frac{K_a}{0.98} \\ 0 & \frac{-b}{J} \end{bmatrix}}_{A_C} \underbrace{\begin{bmatrix} S_a - \frac{L_a \dot{I}_a}{0.98} \\ \omega \end{bmatrix}}_{\chi(t)} + \underbrace{\begin{bmatrix} \frac{R_a}{0.98} & 0 \\ \frac{K_a}{J} & \frac{-1}{J} \end{bmatrix}}_{B_C} \underbrace{\begin{bmatrix} I_a \\ T_{mech} \end{bmatrix}}_{\zeta(t)} \tag{2-52}$$

$$\eta(t) = \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}_{C_C} \chi(t) + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}}_{D_C} \zeta(t) \tag{2-53}$$

The problem of this new output vector is a practical one because the first output contains two components that are not easily measured, S_a and dI_a/dt. This is mainly because they include integration and differentiation as well as multiplication and summation. However, from the theoretical point of view, it could be accepted. Therefore, the following analysis is done on the basis that both outputs are measurable. Otherwise, the system is considered unobservable and cannot be controlled that way using a predictive controller.

The four new obtained matrices, A_C, B_C, C_C, and D_C, are fed to the MPC toolbox. The block diagram of the system with the MPC speed controller using the armature current as a manipulated variable is shown in Figure 2-17. The DC motor block has the same equations of the model when the armature current was used as an input before. Also, the vehicle model and all the parameters of the road are the same as before. The only thing that has been added here is the block that integrates and differentiates the measured voltage and current signals, respectively, and feeds the signal $S_a - L_a \dot{I}_a$ into the MPC controller as a feedback.

The results of the simulation are shown in Figure 2-18. The simulation shows satisfactory results as the vehicle crossed the ditch while minimizing the consumed electric power and energy, 2.7 Watt and 21 J, respectively, Figure 2-18-i. It is also essential to mention here that a more robust speed control is

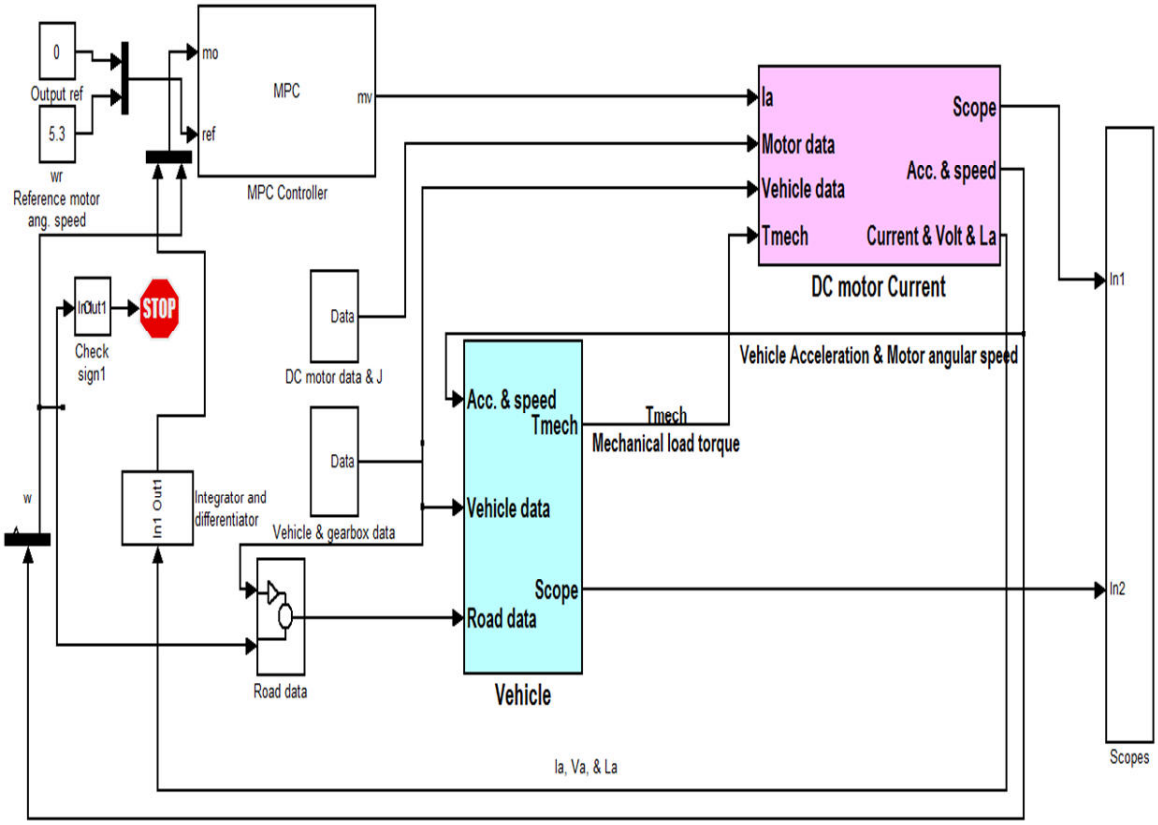


Figure 2-17: Block diagram of the car with MPC speed controller using the armature current as a manipulated variable.

obtained when using the current as a manipulated variable in the predictive control as shown in Figure 2-18-c. The price paid for this is the slight rise in energy consumption in the time duration 6 to 9s as the voltage and current polarities are changed as an attempt to brake the car during going downhill, see Figure 2-18-h. This added consumed energy doesn't affect the overall energy consumption compared with the PID controller.

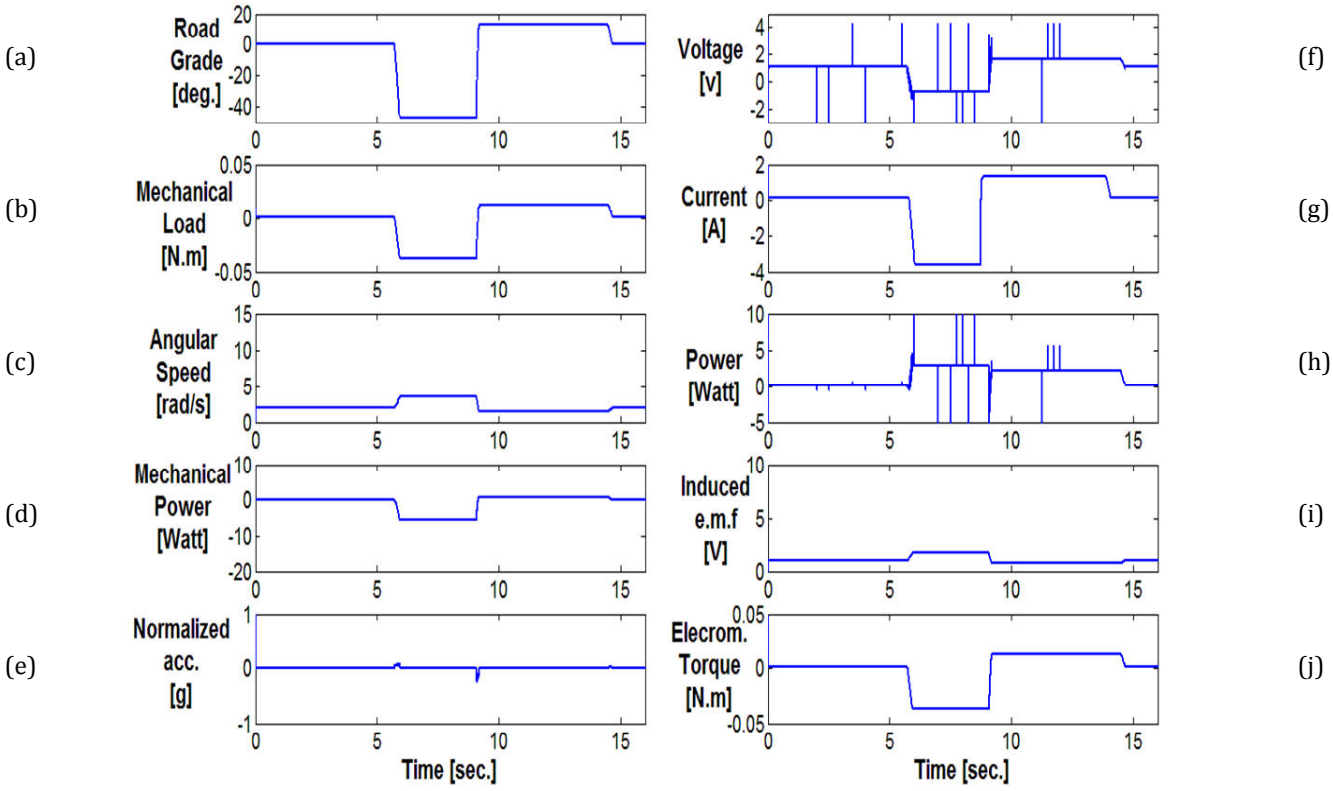


Figure 2-18: DC motor performance with an MPC controller using armature as a manipulated variable. (a) Road slope α, (b) Mechanical load torque T_{mech}, (c) Motor speed ω, (d) Delivered mechanical power on each wheel P_{out}, (e) Car normalized acceleration ξ, (f) Input armature voltage Va, (g) Armature current I_a, (h) Induced e.m.f E_a, (i) Input electric power to each motor P_{in}, and (j) Electromagnetic torque T_{em}.

2.4 Summary of the simulation results

Table 2- 2 shows a summary of the simulation results of the open-loop control, PID control and the MPC control. The results show how the MPC control is capable of reducing the energy consumption while preventing torque saturation occurrence.

Table 2- 2: Summary of simulation results.

Control scheme	Control variable	Electric power consumption [Watt]	Electric energy consumption [J]	Practicability	Torque saturation occurrence
Open-loop control	DC supply voltage	7	77	Acceptable	True
	Armature current	0.1	1.5	Not acceptable	False
PID control	DC supply voltage	7	45	Acceptable	False
	Armature current	4.5	29	Acceptable	False
MPC control	DC supply voltage	3	16.5	Acceptable	False
	Armature current	2.7	21	Acceptable	False

Chapter 3

Experimental work

3.1 Overview

The objective of the experiments is to illustrate how the Predictive Control improves the energy consumption in mobile robots during downhill and uphill motion compared to the PID control and the open-loop control.

3.2 Construction of the test rig

3.2.1 Ramps

The ramps consist of three pieces of wood ramps represent the downhill and uphill as shown in Figure 3- 1.

The total distance of this ramp is 7.2 m; of which 1.2 m level, 2.75 m downhill, 2.75 uphill and 0.5 m level. Both the downhill and the uphill angles are 30^0. The constructional drawings and some photos of the ramp are in appendix.

Figure 3- 1: The ramp consists of three pieces of ramps.

There is another ramp setup that provides other downhill and uphill angles shown in Figure 3-2. This arrangement gives a ramp with downhill angle of 47^0 and uphill angle of 13^0 with the distances 1.5 m downhill and 2.75 m uphill.

The uphill part was sprayed with an adhesive to get higher coefficient of adhesion between the ground and the wheels during the uphill motion.

Figure 3-2: A ramp setup that provides 47^{O} downhill ramp with 13^{O} uphill ramp.

3.2.2 Robot

The mobile robot used here is the Dr. Robot™ X80 mobile robot, shown in Figure 3- 3.

During the initial experiments, the original wheels of the X80 mobile robot did not provide enough traction in the high uphill angles due to their low contact area with the ground. Those wheels were replaced with wider ones with larger contact area with the ground as shown in Figure 3- 4. Photos showing the fixation of the new wheels using new adapters are shown in appendix.

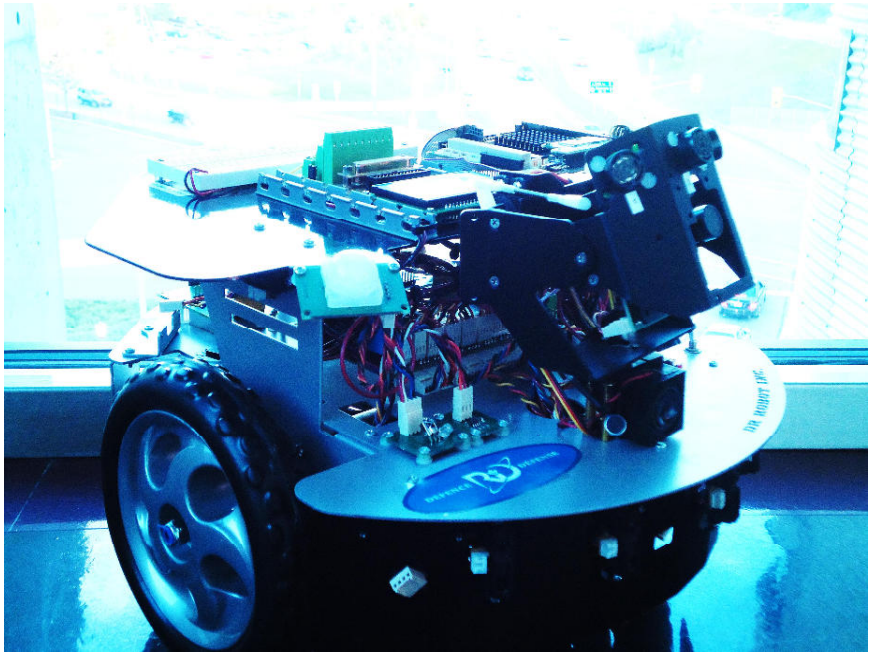


Figure 3- 3: Dr. Robot™ X80 mobile robot is used in the experiments.

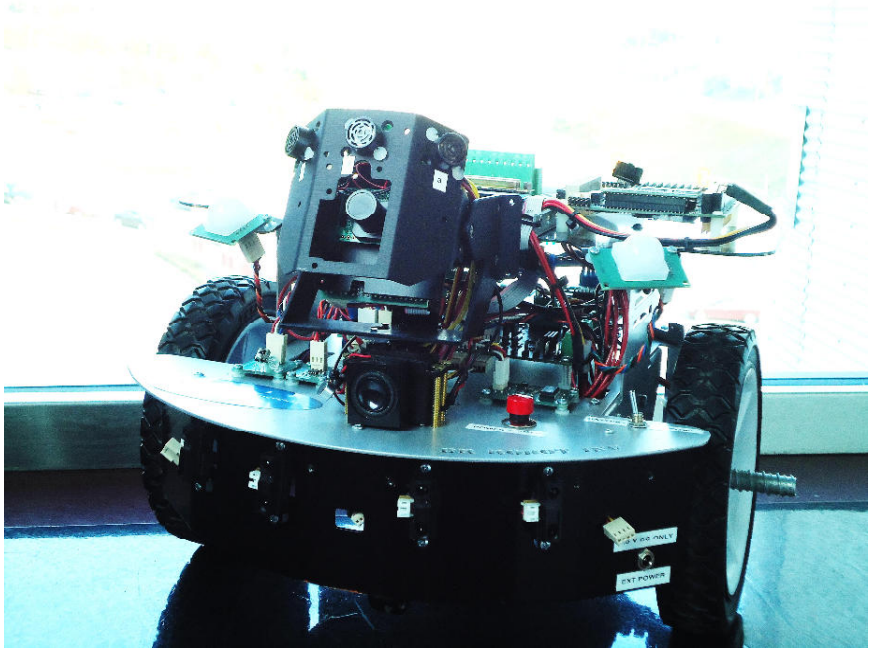


Figure 3- 4: New wider wheels with larger contact area with the ground for improved traction while going uphill.

Due to the fact that the robot tends to pitch over while going downhill, the robot needs to be driven in the reverse direction as the caster wheel comes in the front to prevent the robot from pitching over. On the other hand, due to the reduction of speed of the mobile robot after going downhill and while going uphill, the caster leaves the ground as the robot pitches backward. For that reason another caster was added in the back, but with a height that is lower than that of the original caster so that at any moment only one of the casters is touching the ground, as shown in Figure 3- 5. More photos of the additional caster wheel fixation are in appendix.

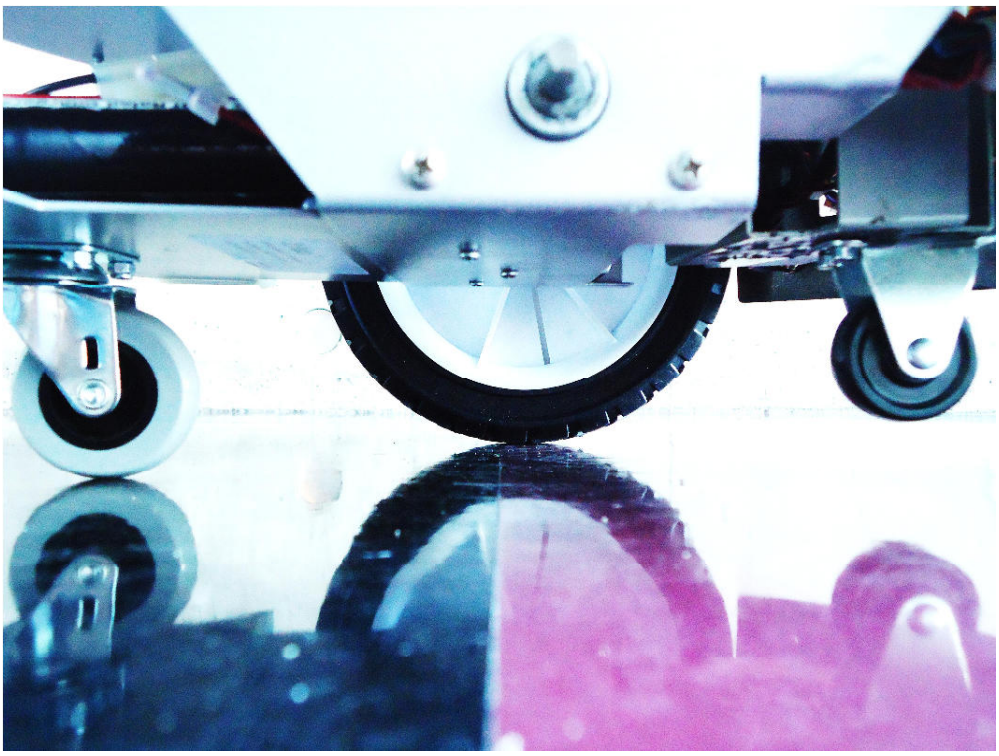


Figure 3- 5: Another caster is added in the back of the robot, but with a height that is lower than that of the front one.

3.3 The experiments

The initial experiments showed that the grade ability, limited by the coefficient of adhesion between the wheels and the ground, of the X80 mobile robot is only 12^0. Any uphill angle upper than that will produce slipping which will affect the measurements. Even using different materials of ground and wheels failed to discard that slippage. Also, different higher loads were added to the robot, but with no success. If the load exceeded a certain limit, the robot will be unstable from the pitch-over point of view.

This means that the driving torque is higher than the adhesion torque limit which led to the use of the second ramp arrangement, the 47^0 downhill and the 13^0 uphill arrangement, in all the experiments. It also led to scarifies using the full power of the motors; instead 60% of the motors power is used. This is accomplished by using a duty cycle of 0.6 in the experiments.

The Model Predictive Control (MPC) used here applies the DC motor voltage as a manipulated variable. The voltage profile fed to the DC motor here is actually obtained from previous predictive control simulations already done earlier, presented in the previous chapter as they were applied on the same robot parameters and same environment conditions. It is assumed here in the experiments that the DC voltage is changed according to the distance covered

which is measured by the motors encoders. Those distances and voltages are obtained from the previous simulations and fed to the experiments.

The robot is released to move downhill for a distance of 1.5 m, and then it goes uphill for a distance of 1.1 m.

The DC supply voltage, the motor armature current, the motor angular speed, and the displacement are measured in the experiments. The consumed electric power by the motors and the consumed electric energy are calculated from the measured voltage and current and displayed in the graphs. Three types of control are used here; the open-loop control, the PID control and the Model Predictive Control (MPC).

Figure 3- 6 shows the X80 mobile robot as visible during the experiments.

Figure 3- 6: The X80 mobile robot during the experiments.

3.4 Experimental results

3.4.1 Open-loop control experimental results

When using open-loop control, the robot failed to overcome the designated hill with the given distance 1.1 m. Figure 3- 7 shows the results obtained when the X80 mobile robot goes downhill and uphill using open-loop control. The graphs include the DC input voltage to the motors, the armature current, the consumed electric power, the consumed electric energy, the motor angular speed, and the distance covered. It can seen from the speed graph that the robot went through the downhill portion with increasing speed then the speed started to diminish while going uphill until it stopped at 7.3 s. At this moment the motor reached saturation which could be seen in the DC current graph. Beyond this point, the robot continues to consume electric energy without moving uphill.

3.4.2 PID control experimental results

The motor performance in case of using PID control is shown in Figure 3-8. The robot succeeded to overcome the uphill part with a reasonable angular speed, 5 rad/s, but the consumed electric energy is high, 72.3 J.

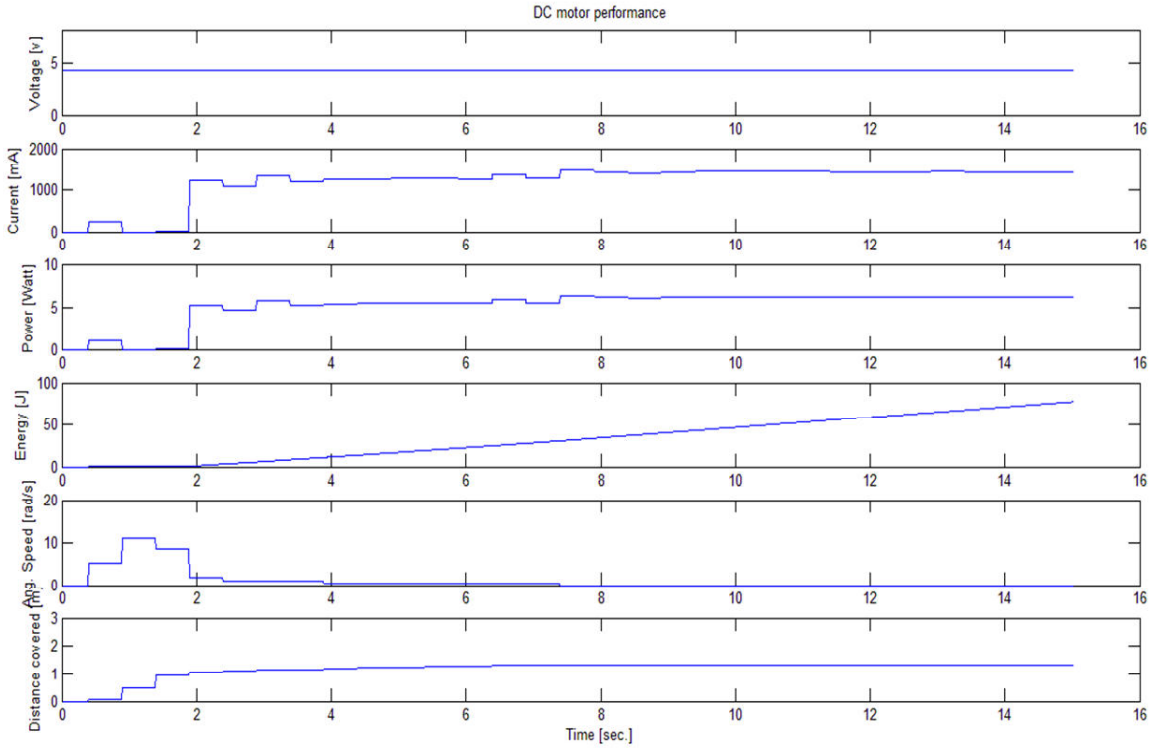


Figure 3- 7: Measured DC motor performance when using open-loop control.

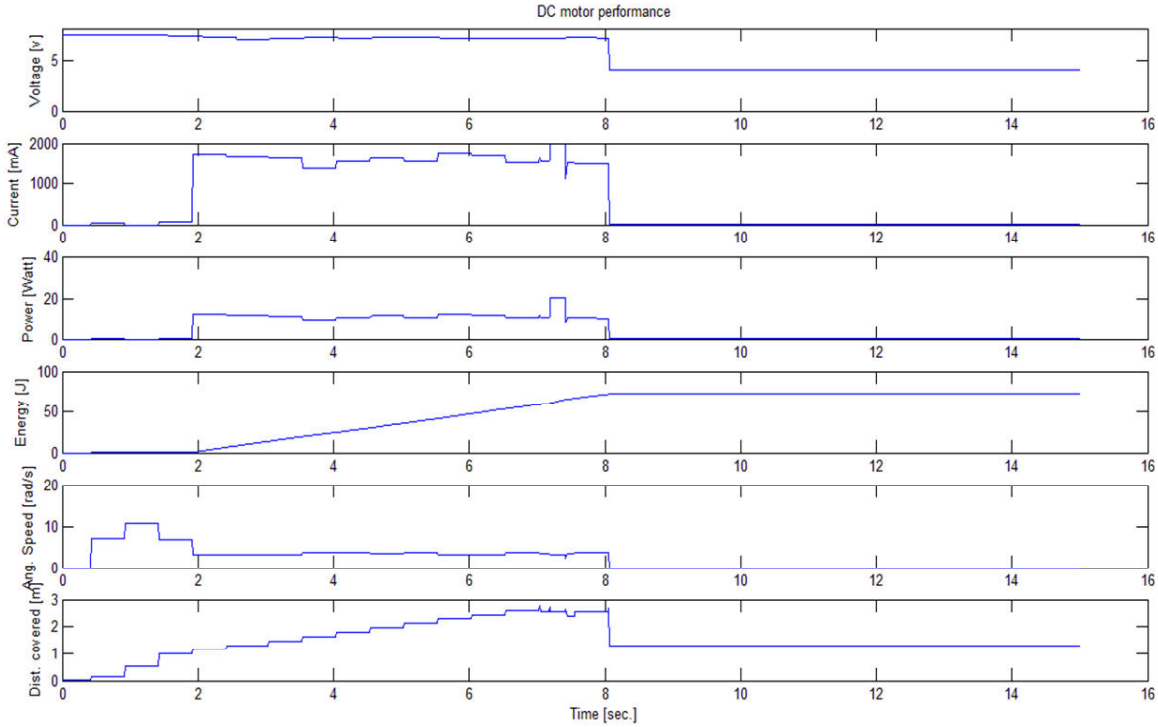


Figure 3- 8: Measured DC motor performance when using PID control.

3.4.3 MPC control experimental results

In the MPC control case, the robot not only succeeded to overcome the hill, but also with a noticeable improvement in the energy consumption, 30.2 J, as shown in Figure 3- 9.

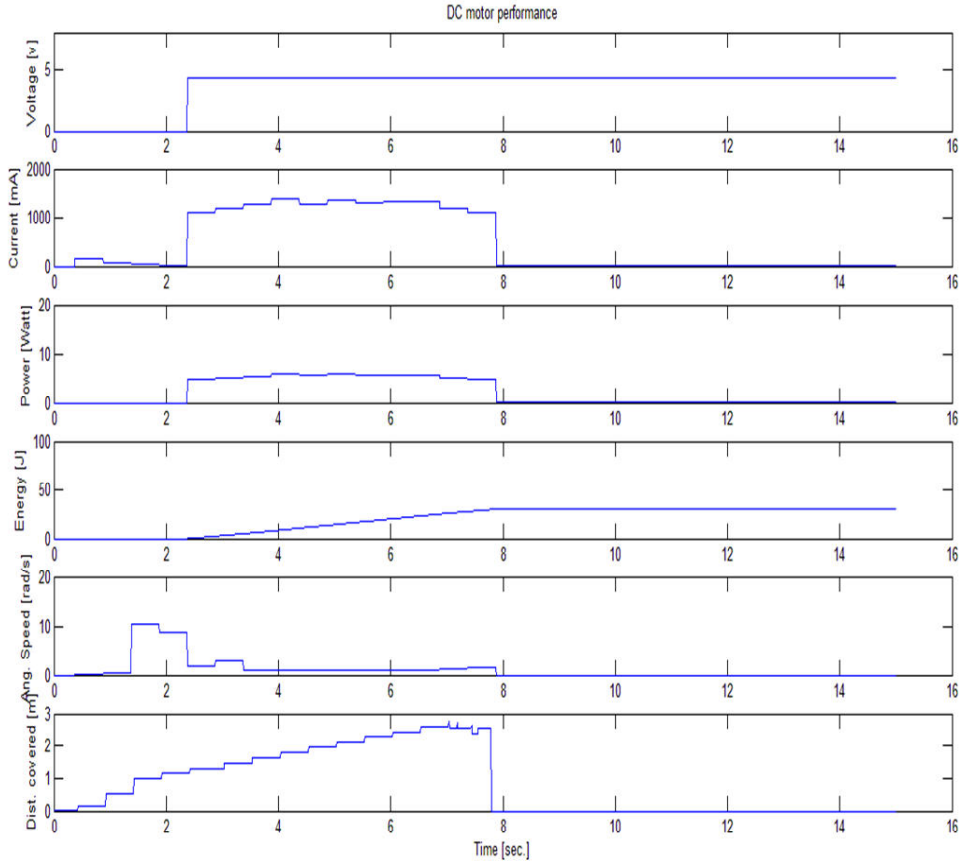


Figure 3- 9: Measured DC motor performance when using MPC control.

3.5 Conclusions

From the experiments results, it is shown that the Model Predictive Control (MPC) could be used to improve the electric energy consumed by the DC motors while crossing ditches.

Chapter 4

Results and analysis

4.1 General

In this chapter, a detailed analysis is conducted to verify how the experimental results agree with the simulation results. This would give a reason to move forward to work with the models developed in the present work in order to more study the predictive control application in mobile robots in off-road environment.

4.2 Open-loop control

The experimental results of the mobile performance when using an open-loop control with constant input DC supply voltage are compared to the simulation results of the similar case. The comparison includes the DC motors angular speed, armature supply voltage, armature current, and the consumed electric power as shown in Figure 4- 1.

As mentioned earlier, the mobile robot failed to cross the ditch in both simulation and experiments. Also, a good agreement of the experimental results with the simulation results could be noticed.

As the current sensor provided with the robot is a unidirectional current sensor, the current value where it should be negative is zero as shown in Figure 4- 2.

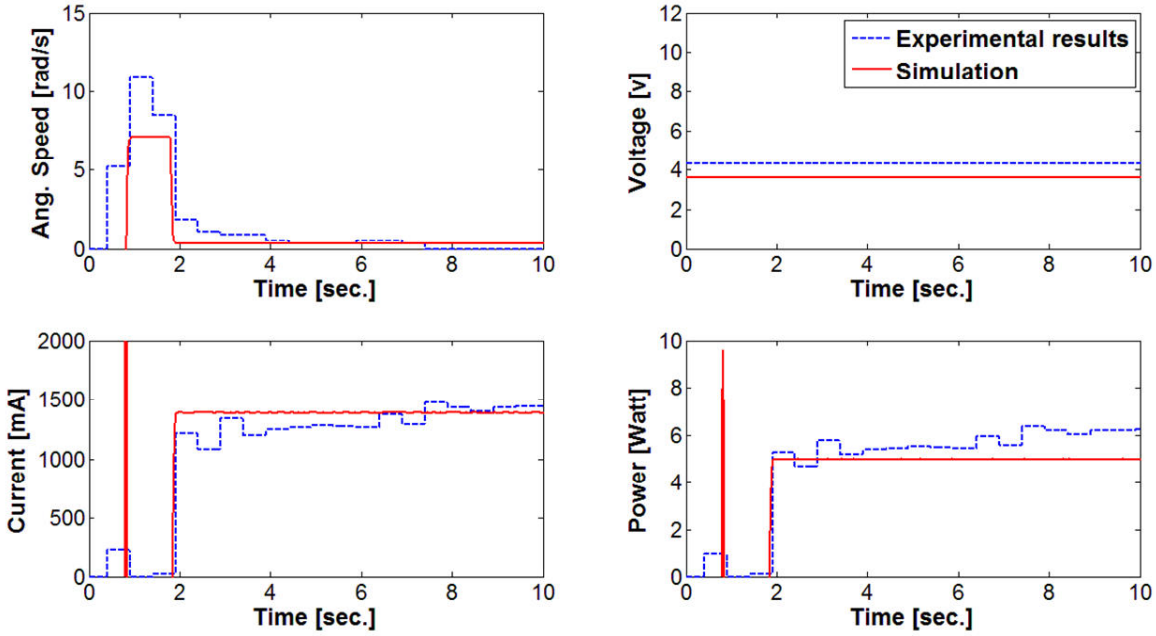


Figure 4- 1: Experimental and simulation results of the wheel DC motor open-loop control using constant supply voltage.

4.3 PID control

In the PID speed control, the experimental results of the wheels DC motors performance by controlling the DC supply voltage are compared to the simulation results. The comparison includes also the motor speed, the supply

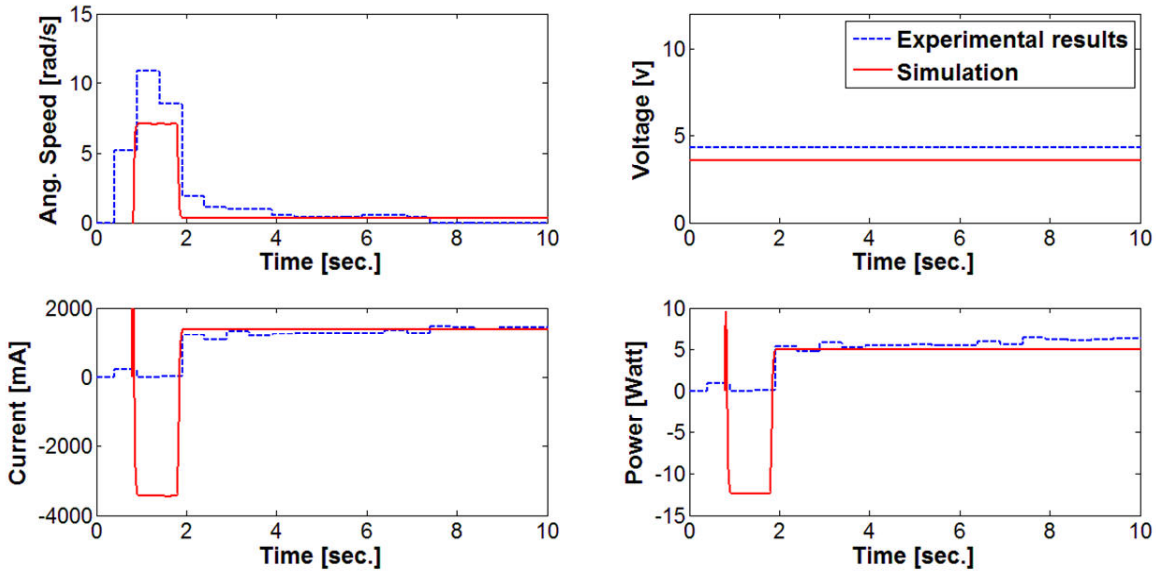


Figure 4- 2: Experimental and simulation results of the wheel DC motor open-loop control using constant supply voltage showing the effect of using a unidirectional current sensor.

voltage, the armature current, and the consumed electric energy. Figure 4- 3 shows the comparison between the experimental results and the simulation results.

It could be seen from the figures that the experimental results agreed to a great extend with the simulation results. As mentioned earlier, the mobile robot succeeded to cross the ditch in the PID control case while it consumed a noticeable amount of electric energy. Again, due to the use of the unidirectional current sensor, the negative side of the current measurements in the experiments are read as zero.

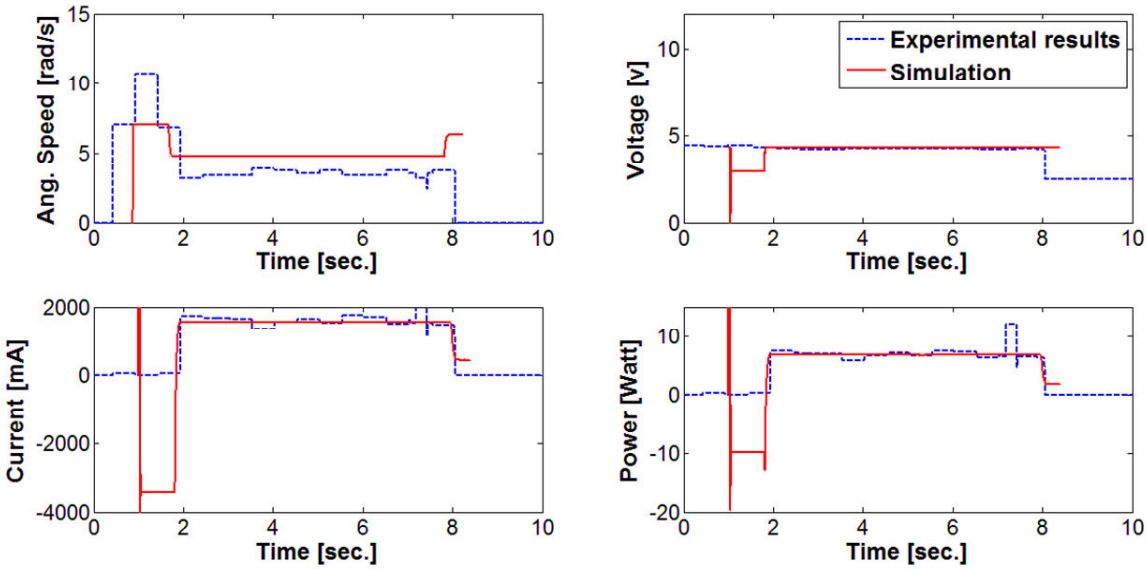


Figure 4- 3: Experimental and simulation results of PID speed control by controlling the supply voltage.

4.4 MPC control

The comparison between the experimental results and the simulation results, when using the supply voltage as a manipulated variable in the MPC control, is shown in Figure 4- 4. The comparison includes wheel speeds, voltage supply, armature current, and consumed electric power.

The results show how the experiments agreed with the simulation in the MPC control. It is also shown here the reduction in energy consumption compared to the PID control case. As mentioned earlier, the negative current is read as zero in the experiments due to the use of a unidirectional current sensor.

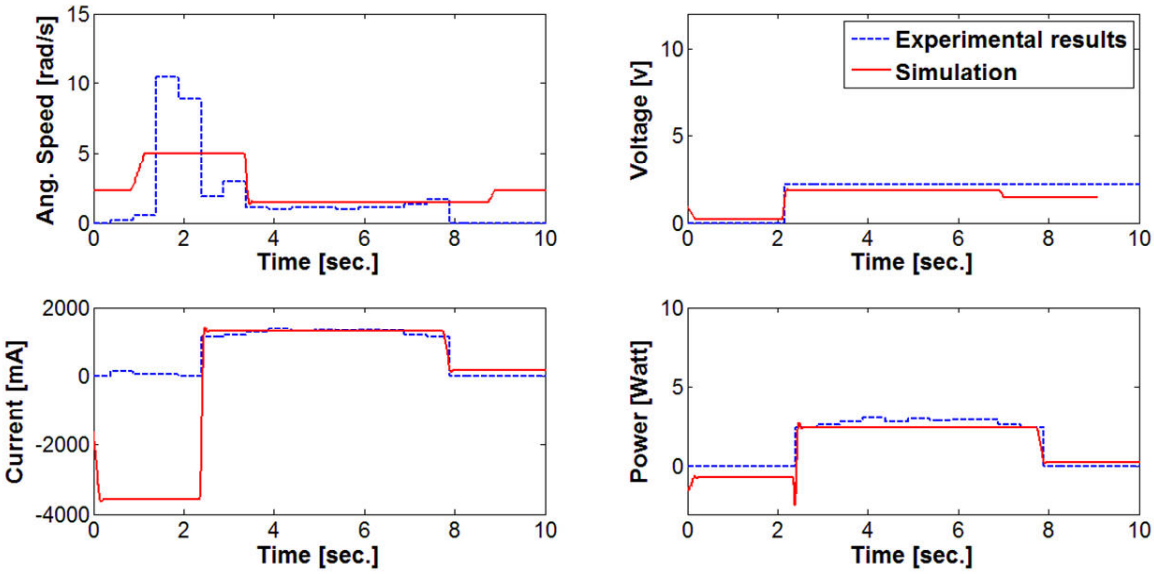


Figure 4- 4: Experimental and simulation results of MPC control when using the supply voltage as the manipulated variable.

Chapter 5

Parametric study

5.1 Overview

As discussed in the previous chapter, the experimental results showed good agreement with the simulation results. This encourages moving forward in parametric study with the developed models to address in detail the predictive control performance in off-road environment. In this chapter, the effect of changing the road slopes on the electric energy consumed by the DC motors is studied. A comparison between the PID control and the MPC control is presented. Also, the effect of changing the prediction horizon on the electric energy consumed by the DC motors is investigated.

5.2 Mathematical representation

Expanding the mathematical models of the mobile robot with the different types of control will help us to add emphasise on the nature of the relations between the studied parameters. For this reason, mathematical representations for the mobile robot models are presented here starting with the open-loop control, continuing with the PID control, and finally ending with the MPC control.

Recalling the open-loop state-space model of the mobile robot, eq.(2-38) and eq. (2-39), the model could be rewritten as

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} \dot{\omega} \\ \dot{I}_a \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{-D}{J} & \frac{K_a}{J} \\ \frac{-K_a}{L_a} & \frac{-R_a}{L_a} \end{bmatrix}}_{A_C} \begin{bmatrix} \omega \\ I_a \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & \frac{-1}{J} & 0 \\ \frac{1}{L_a} & 0 & \frac{-1}{L_a} \end{bmatrix}}_{B_C} \begin{bmatrix} V_a \\ T_{mech} \\ V_{brush} \end{bmatrix} \tag{2-38}$$

$$\begin{bmatrix} y \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}_{C_C} \begin{bmatrix} \omega \\ I_a \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{D_C} \begin{bmatrix} V_a \\ T_{mech} \\ V_{brush} \end{bmatrix} . \tag{2-39}$$

To obtain the transfer function matrix from the state space model, we use the following relation

$$G(s) = C_C(sI - A_C)^{-1}B_C + D_C,$$

where, I is a 2×2 identity matrix and s is the Laplace transform operator. Hence, the transfer function matrix is

$$G(s) = \begin{bmatrix} \frac{K_a}{H(s)} & -\frac{sL_a + R_a}{H(s)} & -\frac{K_a}{H(s)} \\ \frac{sJ + b}{H(s)} & \frac{K_a}{H(s)} & -\frac{sJ + b}{H(s)} \end{bmatrix} \tag{5-1}$$

where, $H(s) = JL_a s^2 + (JR_a + L_a b)s + R_a b + K_a.$

Using the Dr. Robot X80® mobile robot and the Canon® DC motors parameters provided in [34] and [35], respectively, we obtain the following open-loop transfer function matrix

$$G(s) = \begin{bmatrix} \frac{3.579 \times 10^5}{s^2 + 101.2s + 3711} & \frac{-250000s - 2.5 \times 10^7}{s^2 + 101.2s + 3711} & \frac{-3.579 \times 10^5}{s^2 + 101.2s + 3711} \\ \frac{142.9s + 178.6}{s^2 + 101.2s + 3711} & \frac{3.579 \times 10^5}{s^2 + 101.2s + 3711} & \frac{-142.9s - 178.6}{s^2 + 101.2s + 3711} \end{bmatrix} \tag{5-2}$$

The discrete-time transfer function matrix can be obtained after discretizing the continuous state space model, eq. (2-38) and eq. (2-39), with a sampling time T=0.1 s using the discretizing method and assuming Zero-Order Hold (ZOH) for the inputs u

$$A = e^{A_C T} = \mathcal{L}^{-1}\{(sI - A_C)^{-1}\}_{t=T}$$

$$B = \left(\int_{\tau=0}^{T} e^{A_C \tau} d\tau\right) B_C = A_C^{-1}(A - I)B_C$$

, if A_C is nonsingular (5-3)

$$C = C_C$$

$$D = D_C$$

where, A, B, C and D are the discrete-time state space matrices and they are having those denotations only in the current context. So, the discrete-time transfer function matrix is obtained using the following relation

$$G(z) = C\ (zI - A\ )^{-1}B\ + D\ .$$

After substituting with the robot parameters, the following discrete-time function matrix is obtained

$$G(z) = \begin{bmatrix} \frac{97.25z + 0.3734}{z^2 + 0.012z + 4 \times 10^{-5}} & \frac{-6783z - 37.48}{z^2 + 0.012z + 4 \times 10^{-5}} & \frac{-97.25z - 0.3734}{z^2 + 0.012z + 4 \times 10^{-5}} \\ \frac{0.04202z + 0.006696}{z^2 + 0.012z + 4 \times 10^{-5}} & \frac{97.25z + 0.3734}{z^2 + 0.012z + 4 \times 10^{-5}} & \frac{-0.04202z - 0.006696}{z^2 + 0.012z + 4 \times 10^{-5}} \end{bmatrix} \quad (5\text{-}4)$$

When the sample time decreases to T=0.02 s, the following discrete-time function matrix is obtained

$$G(z) = \begin{bmatrix} \frac{36.32z + 18.25}{z^2 - 0.566z + 0.132} & \frac{-4218z + 405.4}{z^2 - 0.566z + 0.132} & \frac{-36.32z - 18.25}{z^2 - 0.566z + 0.132} \\ \frac{0.98z - 0.9513}{z^2 - 0.566z + 0.132} & \frac{36.32z + 18.25}{z^2 - 0.566z + 0.132} & \frac{-0.98z + 0.9513}{z^2 - 0.566z + 0.132} \end{bmatrix} \quad (5\text{-}5)$$

The small sampling time was not examined practically; however, it is evaluated in the simulation. The main concern when using small sampling time in the predictive control is the computational time.

The PID controller transfer function is

$$G_{PID}(s) = K_P + \frac{K_I}{s} + sK_D$$

Converting the transfer function to discrete-time form [47], we obtain

$$G_{PID}(z) = \frac{(K_{P_z} + K_{P_z} + K_{D_z})z^2 - (K_{P_z} + 2K_{D_z})z + K_{D_z}}{z^2 - z} \tag{5-6}$$

where,

$$K_{P_z} = K_P - K_I\frac{T}{2}$$

$$K_{I_z} = K_I T$$

$$K_{D_z} = \frac{K_D}{T}$$

The previous analysis, along with the formulation developed in chapter 2, is used to study the effect of different parameters on the total electric energy consumed by the mobile robot DC motor wheels. In all cases, the consumed electric energy, E, is obtained by the following equation

$$E(t) = \int V_a . I_a dt.$$

5.3 Effect of changing the road slopes

In order to study the effect of the road slope on the energy consumption, different combinations of the downhill and uphill angles need be fed to the models. The mobile robot model presented in chapter 2 is used in this section,

but this time the downhill angle and the uphill angle are variables instead of having single value. The downhill angle is varied from -47^0 to 0^0. Also, the uphill angle is varied from 0^0 to 13^0. In order to illustrate the effect of both slopes on the consumed electric energy by the DC motors, a surface showing this relation when using PID control is presented in Figure 5- 1. As shown in the figure, the consumed energy increased the uphill grade increases. On the other hand, the consumed energy is decreased when the absolute value of the downhill grade increases. The zone at which the experiments as well as the preliminary simulations were considered is at the far right end of the graph where the downhill angle is -47^0 and the uphill angle is 13^0.

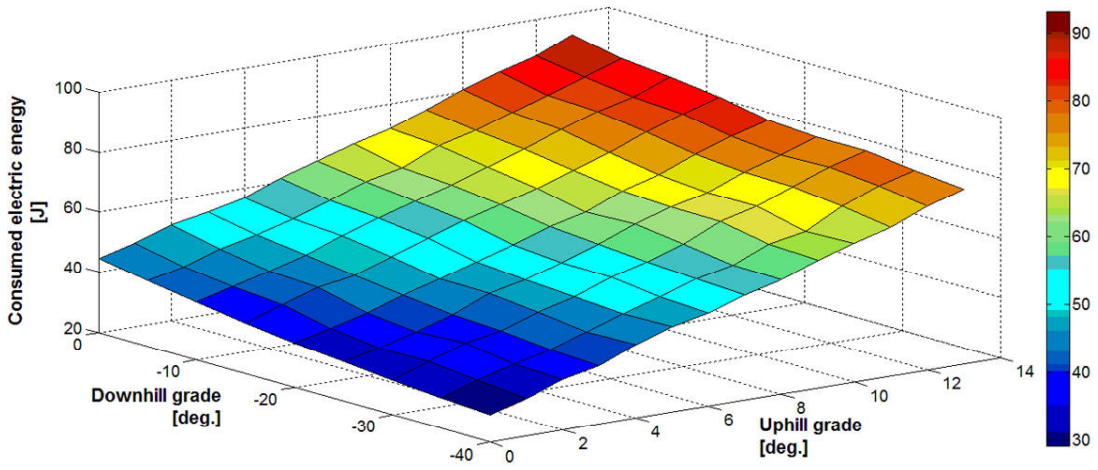


Figure 5- 1: Effect of downhill and uphill angles on the consumed electric energy by the DC motors of a mobile robot in case of PID control.

Figure 5- 2 shows the effect of the downhill and uphill grades on the consumed electric energy by the DC motors in case of MPC control. It is clear that the MPC control is more energy efficient in general compared to the PID control. Furthermore, the energy consumption increases when the uphill grade increases; however, there is an optimum downhill angle at which the energy consumed is minimized compared to other downhill angle. An explanation for this behaviour is going to be presented in the next section.

5.4 Effect of changing the prediction horizon

As mentioned earlier in chapter 2, the performance of the predictive control depends on the ability to predict the system behaviour ahead of time. The dominant factor in this ability is the prediction horizon. The longer the prediction horizon, the better this performance is. On the other hand, the computation cost is higher when using longer prediction horizon.

The prediction horizon used in Figure 5- 2 is 4 sec. Figure 5- 3 shows the consumed electric energy in case of using MPC control with 6s prediction horizon. It can be seen that the consumed electric energy in the last case is less than that of the previous case. Figure 5- 4, Figure 5- 5 and Figure 5- 6 show the consumed electric energy in case of MPC control with prediction horizons 8s, 12s, and 16s, respectively. It is clear that the consumed energy is reduced with

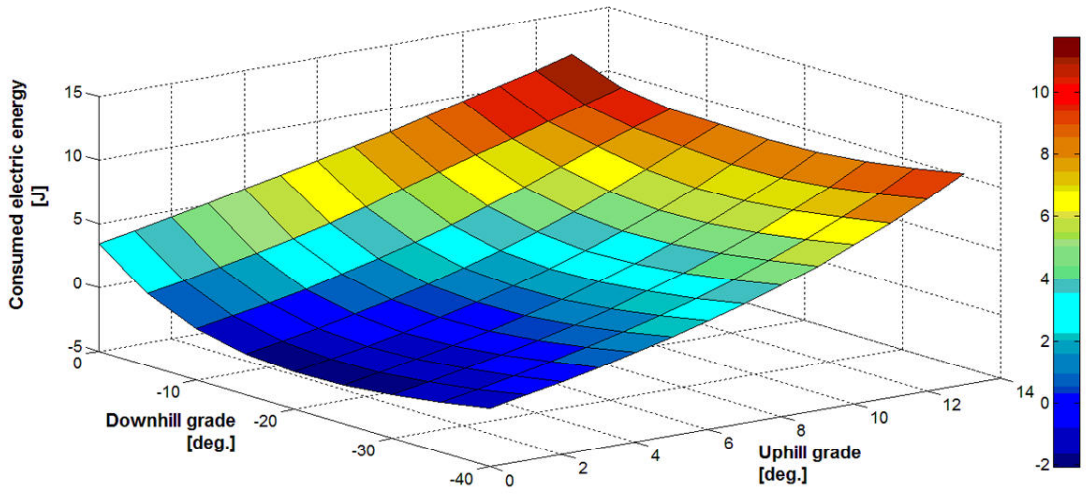


Figure 5- 2: Effect of downhill and uphill grades on the consumed electric energy by the DC motors in case of MPC control with 4 sec. prediction horizon.

the increase of the prediction horizon.

It is essential to mention here that after a certain prediction horizon value, the consumed electric energy cannot be reduced more. This value is the time required to cross the ditch which is about 14s in this case study. Figure 5- 7 shows the consumed electric energy by the DC motors when using MPC control with a prediction horizon of 18s. If compared to Figure 5- 6, the consumed electric energy did not change.

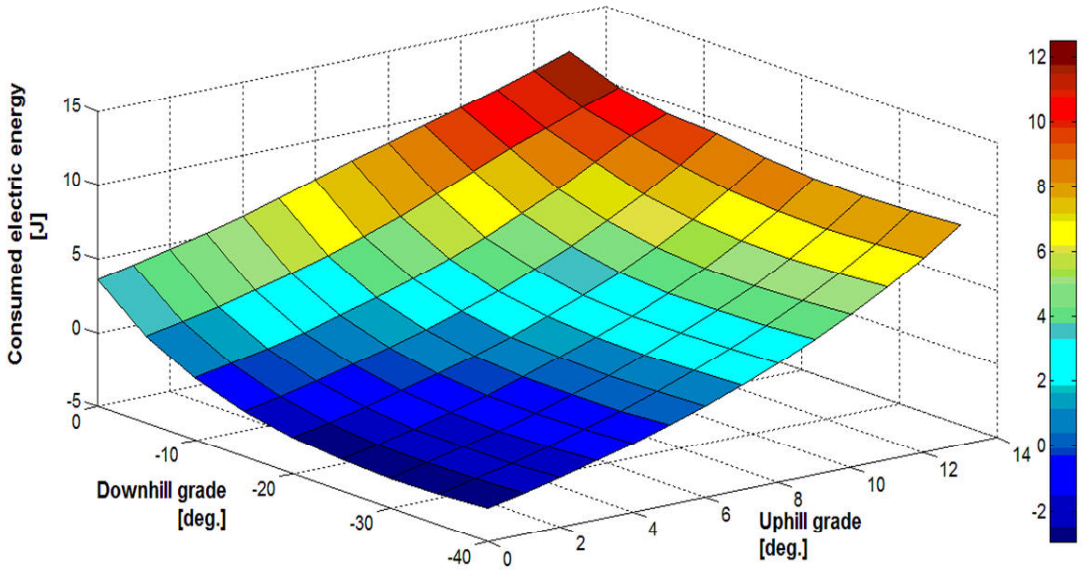


Figure 5- 3: Effect of downhill and uphill grades on the consumed electric energy by the DC motors in case of MPC control with 6 sec. prediction horizon.

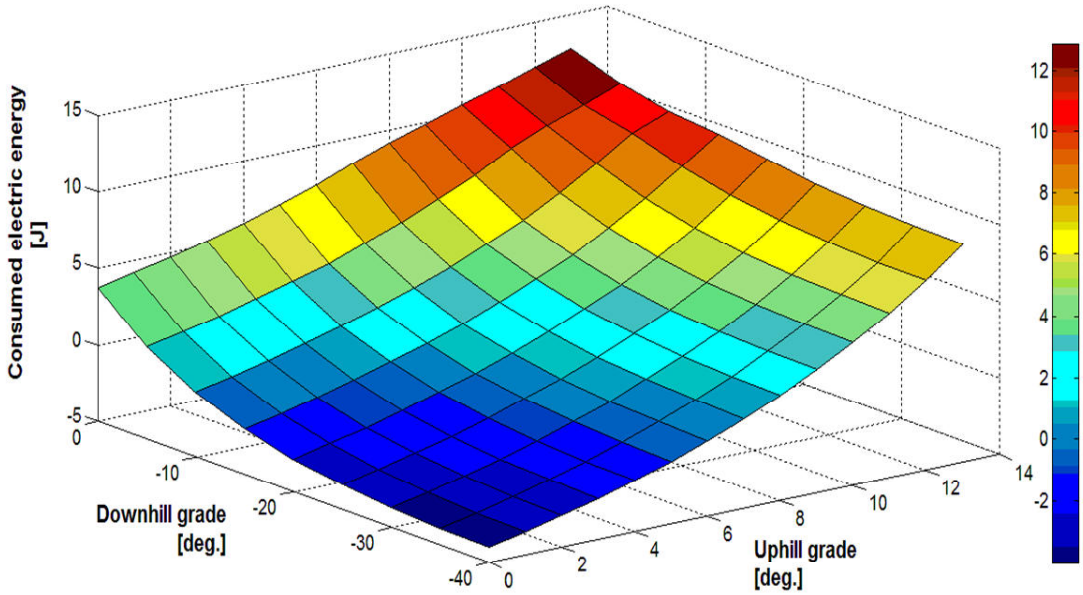


Figure 5- 4: Effect of downhill and uphill grades on the consumed electric energy by the DC motors in case of MPC control with 8 sec. prediction horizon.

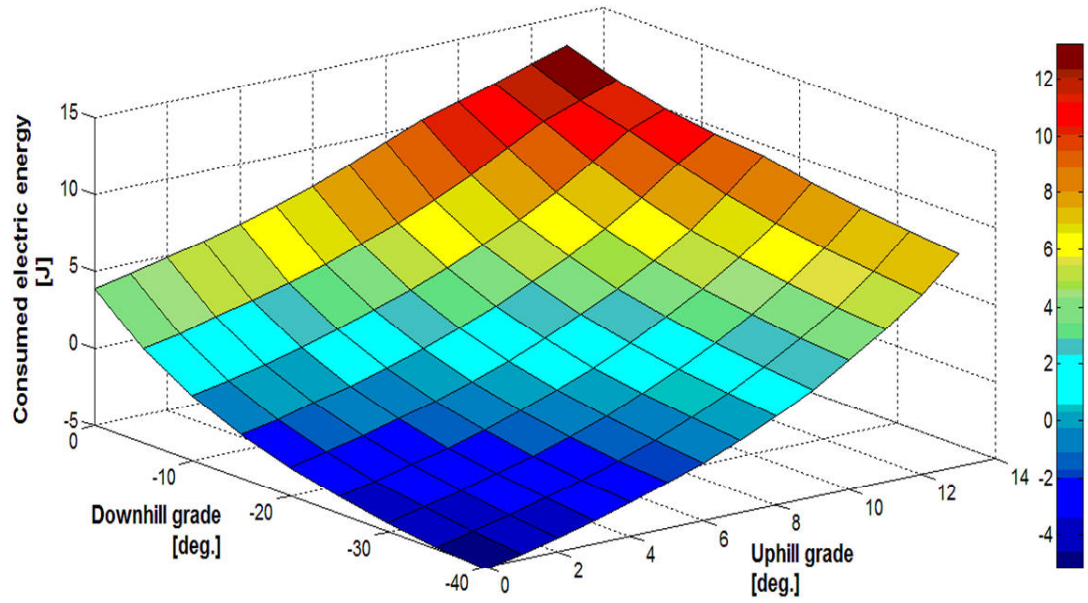


Figure 5- 5: Effect of downhill and uphill grades on the consumed electric energy by the DC motors in case of MPC control with 12 sec. prediction horizon.

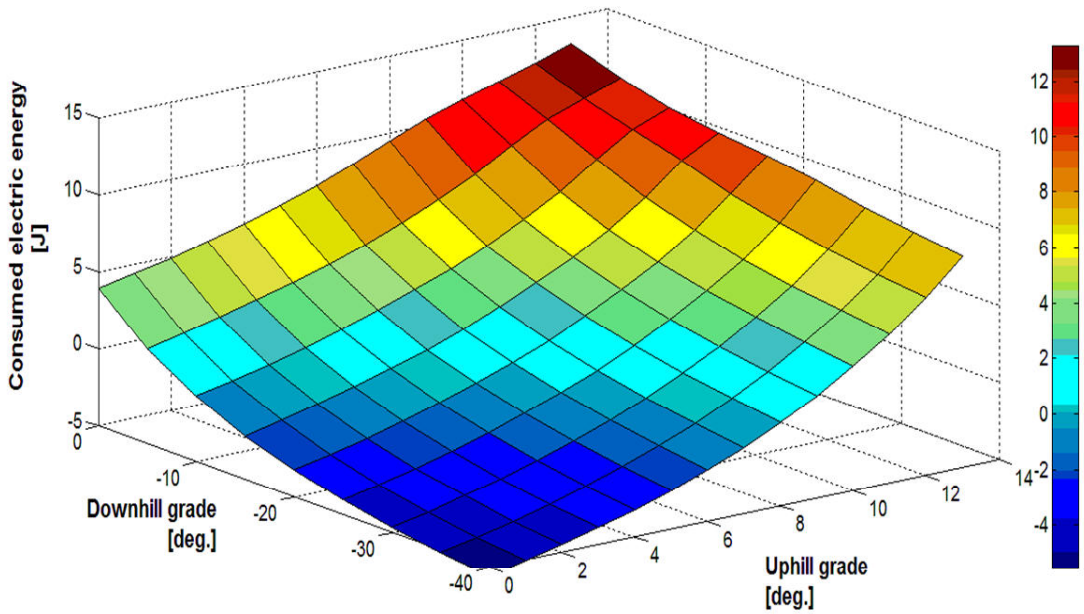


Figure 5- 6: Effect of downhill and uphill grades on the consumed electric energy by the DC motors in case of MPC control with 16 sec. prediction horizon.

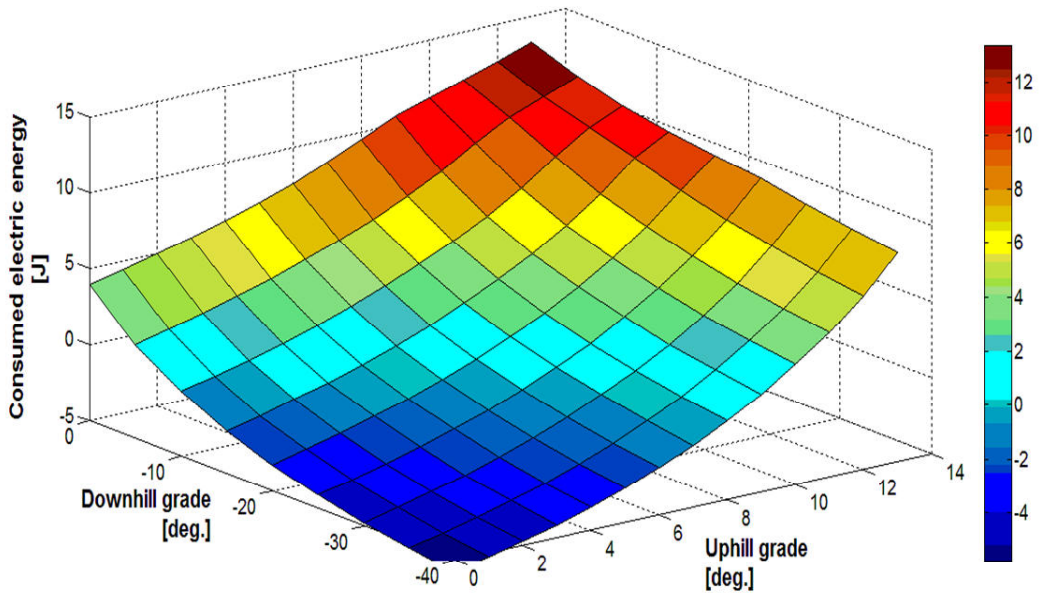


Figure 5- 7: Effect of downhill and uphill grades on the consumed electric energy by the DC motors in case of MPC control with 18 sec. prediction horizon.

5.5 Curve fitting

In this section, curve fitting of the obtained simulation results will be done to more address the effect of changing the road slopes and the prediction horizon on the consumed electric energy of the DC motors of the mobile robot.

The curve fitting will be considered here is a bivariate quintic polynomial as it is the maximum available curve fitting degree provided by the MATLAB® Curve Fitting Toolbox®. The consumed electric energy could be expressed in polynomial form as

$$E(\beta,\gamma) = \sum_{i=0}^{5}\sum_{j=0}^{5} a_{i,j}\beta^i\gamma^j$$

(5-7)

where, β and γ are the uphill and downhill angles in degrees, respectively, and $a_{i,j}$ are the polynomial coefficients. The curve fitting results obtained are summarized in Table 5- 1.

There are four statistics used in the MATLAB® Curve Fitting Toolbox® to examine the goodness-of-fit. These statistics are described as follows as found in [48] and [49]:

Sum of Squares due to Errors (SSE)

This statistic measures the total deviation of the response values from the fit to the response values. It is also called the summed square of residuals and is usually labeled as SSE.

$$SSE = \sum_{i=1}^{n} w_i(y_i - \hat{y}_i)^2$$

A value closer to 0 indicates that the model has a smaller random error component, and that the fit will be more useful for prediction.

R-Square

This statistic measures how successful the fit is in explaining the variation of the data. Put another way, R-square is the square of the correlation between the response values and the predicted response values. It is also called the square of the multiple correlation coefficient and the coefficient of multiple determination.

R-square is defined as the ratio of the sum of squares of the regression (SSR) and the total sum of squares (SST). SSR is defined as

$$SSR = \sum_{i=1}^{n} w_i(\hat{y}_i - \bar{y})^2$$

SST is also called the sum of squares about the mean, and is defined as

$$SST = \sum_{i=1}^{n} w_i(y_i - \bar{y})^2$$

where SST = SSR + SSE. Given these definitions, R-square is expressed as

$$R - square = \frac{SSR}{SST} = 1 - \frac{SSE}{SST}$$

R-square can take on any value between 0 and 1, with a value closer to 1 indicating that a greater proportion of variance is accounted for by the model.

For example, an R-square value of 0.8234 means that the fit explains 82.34% of the total variation in the data about the average.

If you increase the number of fitted coefficients in your model, R-square will increase although the fit may not improve in a practical sense. To avoid this situation, you should use the degrees of freedom adjusted R-square statistic described below.

Note that it is possible to get a negative R-square for equations that do not contain a constant term. Because R-square is defined as the proportion of variance explained by the fit, if the fit is actually worse than just fitting a horizontal line then R-square is negative. In this case, R-square cannot be interpreted as the square of a correlation. Such situations indicate that a constant term should be added to the model.

Degrees of Freedom Adjusted R-Square

This statistic uses the R-square statistic defined above, and adjusts it based on the residual degrees of freedom. The residual degrees of freedom is defined as the number of response values n minus the number of fitted coefficients m estimated from the response values.

$$v = n - m$$

v indicates the number of independent pieces of information involving the *n* data points that are required to calculate the sum of squares. Note that if parameters are bounded and one or more of the estimates are at their bounds, then those estimates are regarded as fixed. The degrees of freedom are increased by the number of such parameters.

The adjusted R-square statistic is generally the best indicator of the fit quality when you compare two models that are nested —that is, a series of models each of which adds additional coefficients to the previous model.

$$adjusted\ R-square = 1 - \frac{SSE(n-1)}{SST(v)}$$

The adjusted R-square statistic can take on any value less than or equal to 1, with a value closer to 1 indicating a better fit. Negative values can occur when the model contains terms that do not help to predict the response.

Root Mean Squared Error

This statistic is also known as the fit standard error and the standard error of the regression. It is an estimate of the standard deviation of the random component in the data, and is defined as

$$RMSE = \sqrt{MSE}$$

where MSE is the mean square error or the residual mean square

$$MSE = \frac{SSE}{v}$$

Just as with SSE, an MSE value closer to 0 indicates a fit that is more useful for prediction.

Table 5- 1: Effect of prediction horizon and control type on polynomial coefficients of curve-fitted consumed energy.

Polynomial coefficient name	**Value of the polynomial coefficient**						
	PID	**MPC**					
		Prediction horizon					
		4 sec.	**6 sec.**	**8 sec.**	**12 sec.**	**16 sec.**	**18 sec.**
$a_{0,0}$	44.78	3.461	3.628	3.763	3.902	3.957	3.992
$a_{1,0}$	2.56	0.4829	0.4687	0.4588	0.4672	0.4679	0.4687
$a_{0,1}$	0.7394	0.6883	0.6662	0.6498	0.649	0.6565	0.6623
$a_{2,0}$	0.2857	0.01541	0.01991	0.01458	0.005723	0.005797	0.006441
$a_{1,1}$	-0.05069	-0.003806	0.0007798	-0.001617	-0.005673	-0.00758	-0.008086
$a_{0,2}$	0.03306	0.0301	0.02842	0.02512	0.02272	0.02276	0.02304
$a_{3,0}$	-0.04394	$5.394x10^{-5}$	0.001168	0.002965	0.003983	0.003185	0.002432
$a_{2,1}$	0.0004583	-0.001333	-0.0003945	0.0003437	0.0003288	$8.188x10^{-6}$	-0.0003321
$a_{1,2}$	-0.004261	-0.0005691	-0.0001042	$2.124x10^{-5}$	-0.0002229	-0.0004702	-0.0005962
$a_{0,3}$	0.001833	0.0006106	0.0006973	0.0006129	0.0005099	0.0005045	0.0005165
$a_{4,0}$	0.003503	$5.923x10^{-6}$	$-8.627x10^{-5}$	-0.0001776	-0.0001046	$5.901x10^{-5}$	0.0001912
$a_{3,1}$	$5.555x10^{-6}$	$-2.9x10^{-5}$	$-8.637x10^{-5}$	$-7.625x10^{-5}$	$2.53x10^{-5}$	$9.357x10^{-5}$	0.000143
$a_{2,2}$	$4.457x10^{-5}$	$-4.965x10^{-5}$	$-6.575x10^{-5}$	$-3.655x10^{-5}$	$1.53x10^{-6}$	$1.264x10^{-5}$	$1.422x10^{-5}$
$a_{1,3}$	-0.0001364	$-1.81x10^{-5}$	$-1.674x10^{-5}$	$-7.443x10^{-6}$	$-7.112x10^{-6}$	$-1.319x10^{-5}$	$-1.679x10^{-5}$
$a_{0,4}$	$5.08x10^{-5}$	$7.844x10^{-6}$	$9.825x10^{-6}$	$8.827x10^{-6}$	$6.478x10^{-6}$	$5.965x10^{-6}$	$6.041x10^{-6}$
$a_{5,0}$	$-9.949x10^{-5}$	$-2.842x10^{-6}$	$-2.479x10^{-6}$	$-2.56x10^{-6}$	$-9.941x10^{-6}$	$-1.776x10^{-5}$	$-2.376x10^{-5}$
$a_{4,1}$	$1.595x10^{-5}$	$-3.06x10^{-6}$	$-8.229x10^{-6}$	$-1.19x10^{-5}$	$-1.78x10^{-5}$	$-2.062x10^{-5}$	$-2.241x10^{-5}$
$a_{3,2}$	$1.369x10^{-5}$	$-9.83x10^{-7}$	$-5.135x10^{-6}$	$-6.667x10^{-6}$	$-7.001x10^{-6}$	$-6.617x10^{-6}$	$-6.201x10^{-6}$
$a_{2,3}$	$5.049x10^{-6}$	$-4.575x10^{-7}$	$-1.941x10^{-6}$	$-1.94x10^{-6}$	$-1.384x10^{-6}$	$-1.077x10^{-6}$	$-9.266x10^{-7}$
$a_{1,4}$	$-1.014x10^{-6}$	$-1.363x10^{-7}$	$-3.676x10^{-7}$	$-2.671x10^{-7}$	$-1.846x10^{-7}$	$-2.16x10^{-7}$	$-2.373x10^{-7}$
$a_{0,5}$	$5.063x10^{-7}$	$5.339x10^{-8}$	$5.521x10^{-8}$	$5.136x10^{-8}$	$3.233x10^{-8}$	$2.516x10^{-8}$	$2.428x10^{-8}$
SSE	42.7002	0.3704	0.4040	0.4633	0.5382	0.5749	0.6172
R^2	0.9987	0.9998	0.9998	0.9998	0.9998	0.9998	0.9998
Adjusted R^2	0.9984	0.9997	0.9998	0.9998	0.9998	0.9998	0.9998
RMSE	0.6377	0.0594	0.0620	0.0664	0.0716	0.0740	0.0767

Figure 5- 8 helps in visual examination of curve fitting of the obtained consumed electric energy data for MPC control. As it is seen from the figure, the surface fits the obtained data satisfactorily. As it is seen from Table 5- 1, the fit of the PID controller has less capability of predicting the response compared with the MPC controllers fits, as seen from the high SSE and RMSE, however it is still acceptable, as seen from the R^2 and the adjusted R^2. This is caused by the rough planner nature of the energy surface in the PID case, see Figure 5- 1.

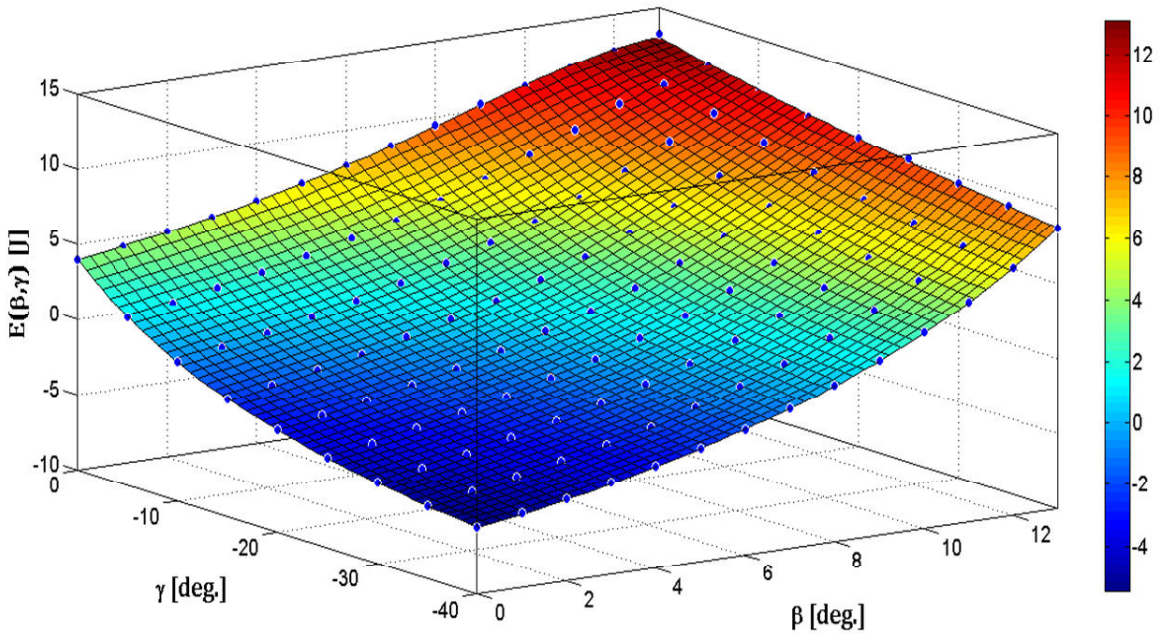


Figure 5- 8: Curve fitting of simulated consumed energy data of MPC control.

The dots represent the energy data obtained from simulation.

For the MPC control, the following polynomial coefficients are found sensitive to the change in prediction horizon

$a_{0,0}$ increases by increasing the prediction horizon,

$a_{1,1}$, $a_{2,1,}$ $a_{4,1}$ and $a_{0,5}$ decrease by increasing the prediction horizon. In the same sense, there are some polynomial coefficients that are inconsistently changing with the change of the prediction horizon, $a_{2,0}$, $a_{0,2}$, $a_{1,2}$, $a_{2,2}$, $a_{5,0}$ and $a_{2,3}$. On the other hand, all the remaining polynomial coefficients are uncorrelated to the change of the prediction horizon. The change of the first mentioned coefficients with the prediction horizon needs more investigation to address the degree of change within those coefficients.

Chapter 6

Conclusions and recommendations for future work

6.1 Conclusions

Predictive control is applied to control the drive of electrically driven mobile robot moving in off-road conditions.

The off-road condition is utilized such that benefits are obtained with regard to energy consumption and torque saturation. The predictive control is used to control the DC motors of a mobile robot while crossing a ditch, given a prior knowledge of the road profile. The aim of the study is to investigate the capability of the predictive control to prevent the motors from reaching torque saturation. Additionally, the effect of using the predictive control on the energy consumption is investigated.

A comparison between the predictive control and the classical control is accomplished. The classical control used in the current research is the PID control in addition to the open-loop control.

The results of the study are summarized as follows:

In the simulation, the predictive control succeeded to overcome a ditch with slope angles beyond the gradeability of the mobile robot. It also showed a noticeable reduction in the energy consumption compared with the PID control and the open-loop control, which actually failed to overcome the ditch.

Using the armature current instead of the armature supply voltage as a control variable in the predictive control simulation showed more efficient control as well as reduced energy consumption.

Experimental work using a mobile robot moving on ramps of various slopes was conducted to verify the simulation results of the predictive control compared with the PID control and the open-loop control. The experimental results agreed to a great extend with the simulation results.

A parametric study was conducted for the developed predictive control model. The consumed electric energy by the DC motors was examined by observing the effect of the road downhill and uphill grades. The effect of the prediction horizon on the consumed energy was also investigated. The simulation results showed that the consumed energy is reduced when using the predictive control compared with using the PID control. Additionally, when using the predictive control, the energy consumption is reduced by increasing

the prediction horizon until certain limit is reached. This limit is dominated by the width of the ditch in front of the vehicle and the vehicle speed.

In general, using the predictive control for mobile robots in off-road environment enhances its gradeability by eliminating torque saturation occurrence at the original gradeability. In addition, the energy consumption is reduced compared with using the PID speed control.

6.2 Recommendations for future work

Practically, the predictive control needs to know the 3D map ahead of time before reaching the obstacle. It is advisable to use one of the practical techniques to build the 3D map while moving in the off-road environment like using 3D range finders, sonar sensors or machine vision.

More real off-road features need to be considered such as irregular holes or irregular hills.

A more generic strategy needs to be proposed and evaluated for moving a wheeled vehicle in an off-road environment in order to optimize energy consumption by disabling the predictive control whenever it is not needed and activating the artificial velocity fields for example. Figure 6- 1 shows a proposed strategy for an unmanned vehicle moving in an off-road environment.

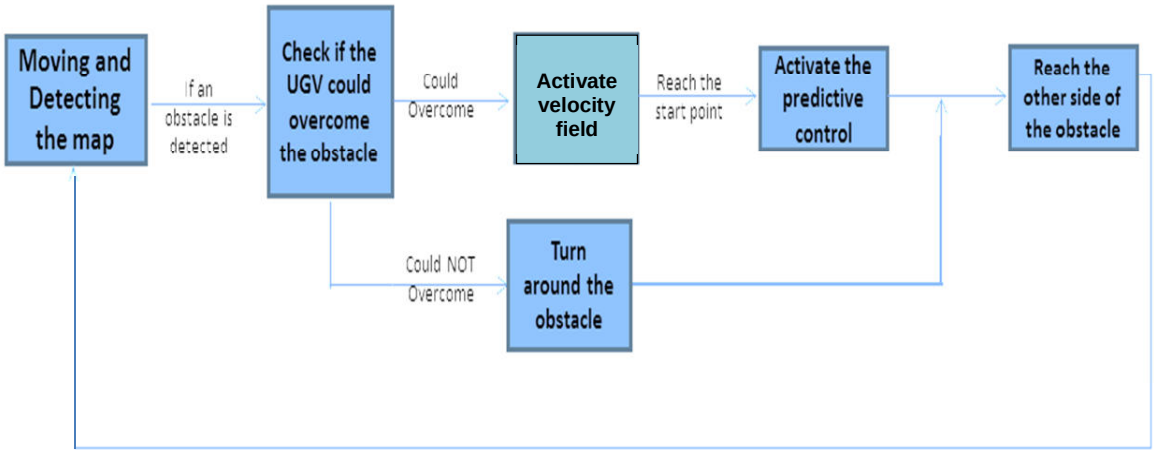


Figure 6- 1: A proposed strategy for an unmanned wheeled vehicle moving in an off-road environment.

References

[1] Douglas W. Gage, “Unmanned Ground Vehicle (UGV) development efforts”, Unmanned Systems Magazine, vol. 13, no. 3, pp. 9-16, summer 1995.

[2] Nilsson, Nils J., “A mobile automaton: An application of artificial intelligence techniques”, International Joint Conference on Artificial Intelligence, Washington DC, May 1969.

[3] Flynn, Anita A., “Redundant sensors for mobile robot navigation”, MIT AI Lab Technical Report, pp. 11-15, Cambridge, MA, 1985.

[4] Moravec, H.P. “Obstacle avoidance and navigation in the real world by a seeing robot rover”, PhD thesis, Stanford University, Pittsburgh, PA, September 1980.

[5] Moravec, H.P., “The Stanford cart and CMU rover”, Proceedings of IEEE, vol. 71, no. 7, pp. 872–884, 1983.

[6] Thrope, C., “Vision and navigation of the Carnegie-Mellon Navlab”, IEEE transactions on pattern analysis and machine intelligence, vol 10, no. 3, pp. 362-373, 1988.

[7] Seida, Steven, Morgenthaler, G., Podlaseck, M., Douglass, B., McSwain, J., Knourek, R., Thanas, M., “Vision-based road following in the autonomous land vehicle”, Proceedings of the 26th IEEE conference on decision and control, vol 11, pp. 1814-1819, Los Angeles, CA, September 1987.

[8] Gleichman, K., Harmon, L., Miciek, T., Miller, C., Zuk, D., “Sensor support for the DARPA autonomous land vehicle program”, Final Report, Defense Technical Information Center, pp. 8-49, Fort Belvoir, VA, October 1988.

[9] Shoemaker, C. M., and Bomstein, J. A. " The DEMO III UGV program: A testbed for autonomous navigation research", Proc. of IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA), Intelligent Systems and Semiotics (ISAS) Joint Conference, pp. 644-651, Gaithersburg, MD, USA, September 1998.

[10] Kurtz, J., "DEMO III experimental unmanned vehicle autonomous mobility system overview", Proc. of IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA), Intelligent Systems and Semiotics (ISAS) Joint Conference, pp. 640-643, Gaithersburg, MD, USA, September 1998.

[11] Ron Kurjanowicz, "Special Issue on the DARPA Grand Challenge", Foreword of Journal of Field Robotics, vol. 23, no. 8, pp. 461-462, August 2006.

[12] Sebastian Thrun *et al.*, "Stanley: the robot that won the DARPA Grand Challenge", Journal of Field Robotics, vol. 23, no. 9, pp. 661-692, September 2006.

[13] Inge Spangelo, Olav Egeland, "Trajectory planning and collision avoidance for underwater vehicles using optimal control", IEEE Journal of Oceanic Engineering, vol. 19, no. 4, pp. 502-511, October 1994.

[14] Chuy, O., Collins E., Yu, W., Ordonez, C., "Power modeling of a skid steered wheeled robotic ground vehicle", IEEE International Conference on Robotics and Automation, Kobe, May 2009.

[15] Zhang, W., Lu, Y., Hu, J., "Optimal solutions to a class of power management problems in Mobile Robots", Automatica (Journal of IFAC), vol. 45, no. 4, pp. 989-996, Tarrytown, NY, April 2009.

[16] Jito Vanualailai, Bibhya Sharma, and Shin-ichi Nakagiri, “An asymptotically stable collision-avoidance system”, International Journal of Nonlinear Mechanics, vol. 43, no. 9, pp. 925-932, November 2008.

[17] R. Saravanan, S. Ramabalan, C. Balamurugan, “Evolutionary multi-criteria trajectory modeling of industrial robots in the presence of obstacles”, Engineering Application of Artificial Intelligence, vol. 22, no. 2, pp. 329-342, March 2009.

[18] S.F.P. Saramago, Steffen Jr., “Trajectory modeling of robot manipulators in the presence of obstacles”, Journal of Optimization Theory and Applications, vol. 110, no. 1, pp. 17-34, July 2001.

[19] Jae-Hoon Jang, Seung-Ho Bae, Myung-Wook Park , and Jung-Ha Kim, “Research of velocity control on the slope road for unmanned ground vehicle”, International Conference on Control, Automation and Systems, pp. 1085-1088, Gyeonggi-do, South Korea, October 2010.

[20] Jesus Morales, Jorge L. Martinez, Anthony Mandow, Alejandro Pequeno-Boter, Alfonso Garcia-Cerezo, “Simplified power consumption modeling and identification for wheeled skid-steer robotic vehicles on hard horizontal ground”, IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 4769-4774, Taipei, Taiwan, October 2010.

[21] J. Morales, J. L. Mart´ınez, A. Mandow, A. Garc´ıa-Cerezo, and S. Pedraza, “Power consumption modeling of skid-steer tracked mobile robots on rigid terrain,” IEEE Transactions on Robotics, vol. 25, no. 5, pp. 1098–1108, October 2009.

[22] W. Press, S. Teukolsky, W. Vetterling, B. Flannery, “Downhill simplex method in multidimensions”, Numerical Recipes in C, 2nd ed., Ch. 10.4, pp. 408-412, Cambridge University Press, Cambridge, UK, 1992.

[23] J. A. Nelder and R.Mead, “A simplex method for function minimization,” Computer Journal., vol. 7, no. 4, pp. 308–313, January 1965.

[24] Trung Dung Ngo, Hector Raposo, Henrik Schiøler, “Potentially distributable energy: Towards energy autonomy in large population of mobile robots”, Proceedings of the 2007 IEEE International Symposium on, Computational Intelligence in Robotics and Automation, pp. 206-211, Jacksonville, FL, June 2007.

[25] He Xu, Dawei Tan, Zhenyu Zhang, Yulin Ma and Gaoliang Peng, “An innovative reconfigurable mobile robot with multi-maneuver modes”, Proceedings of the 2008 IEEE, International Conference on Robotics and Biomimetics, pp. 1659-1664, Bangkok, Thailand, February 2009.

[26] P. Suntharalingam, J.T. Economou, and K. Knowles, “Gear locking mechanism to extend the consistent power operating region of the electric motor to enhance acceleration and regenerative braking efficiency in hybrid electric vehicles”, Vehicle Power and Propulsion Conference ‘09 IEEE, pp. 103-108, Dearborn, MI, September 2009.

[27] Wei Yu, Oscar Ylaya Chuy, Emmanuel G. Collins, and Patrick Hollis, “Analysis and experimental verification for dynamic modeling of a skid-steered wheeled vehicle”, IEEE Transactions on Robotics, vol. 26, no. 2, pp. 340-353, April 2010.

[28] Amir Sadrpour, J. Jin, A. G. Alsoy, “Mission energy prediction for Unmanned Ground Vehicles”, IEEE International Conference on Robotics and Automation, pp. 2229-2234, Saint Paul, MN, May 2012.

[29] Qin, S.J., Badgwell, T. A., “A survey of industrial model predictive control technology”, Control Engineering Practice, vol. 11, no. 7, pp. 733–764, July 2003.

[30] Rawlings, J. B., “Tutorial overview of model predictive control”, IEEE Control Systems Magazine, vol. 20, no. 3, pp. 38–52, June 2000.

[31] F. Berlin and P.M. Frank, “Design and realization of a MIMO predictive controller for a 3-tank system”, Advances in Model-Based Predictive Control, Oxford University Press, pp.446-457, 1994.

[32] Wei Jiang , “DC Machines”, Dynamic Simulations of Electric Machinery: Using MATLAB/SIMULINK, ch. 8, pp. 370-399, Prentice Hall, Upper Saddle River, NJ, February 2006.

[33] Jo Yung Wong, “Performance characteristics of road vehicles”, Theory of Ground Vehicles, ch. 3, pp. 203-294, John Wiley & Sons, New York, NY, 2001.

[34] Dr Robot Inc. (2012, May) X80: An industrial grade robot for research and development applications. [Online]. http://www.drrobot.com/products/item_downloads/x80_1.pdf

[35] Canon Inc. (2012, May) DC micro motors. [Online]. http://usa.canon.com/C-USA/assets/app/pdf/motors/DC_Micro_Catalog.pdf

[36] Venom Group (2013 March) Venom 7.2v 3600mAh 6-Cell NiMH Battery with Universal Plug System. [Online]. http://www.venom-group.com/7-2v-3600mah-NiMH-Pack-UNI.html

[37] Maciejowski, Jan (2012 May) Predictive Control: Introduction to predictive control. [Online]. http://www.control.eng.cam.ac.uk/jmm/4f3/predictive1.pdf

[38] Alberto Bemporad, Manfred Morari, and N. Lawrence Ricker, “Model Predictive Control Toolbox user’s guide”, MATLAB R2011b, The MathWorks Inc., 2011.

[39] Liuping Wang, “Discrete-time MPC for beginners”, Model Predictive Control System Design and Implementation Using MATLAB®, ch. 1, pp. 1-42, Springer Verlag London Lt., London, UK, 2009.

[40] J. M. Maciejowski, “A basic formulation of predictive control”, Predictive Control with Constraints, ch. 2, pp. 36-73, Prentice Hall, Essex, England, 2002.

[41] T. Kailath, “State-space and matrix-fractions description”, Linear Systems, ch 6, pp. 345-498, Prentice Hall Inc., Englewood Cliffs, N.J., 1980.

[42] J. S. Bay, “Fundamentals of Linear State Space Systems”, McGraw-Hill, 1999.

[43] Katsuhiko Ogata, “Analysis of control systems in state space”, Modern Control Engineering, 3rd ed., ch 11, pp. 710-785, Prentice Hall Inc., Upper Saddle River, NJ, 1997.

[44] B. D. O. Anderson and J. M. Moore, “Kalman filter properties”, Optimal Filtering, ch 5, pp. 90-128, Prentice Hall Inc., Englewood Cliffs, NJ, 1979.

[45] Zhengming Wang, Dongyun Yi, Xiaojun Duan, Jing Yao, Defeng Gu, "Discrete-time Kalman filter”, Measurement Data Modeling and Parameter Estimation, ch 5, pp. 299-346, CRC Press, Boca Raton, FL, December 2011.

[46] B. Porter and A. Bradshaw, “Multivariable time-invariant linear systems with input-derivative control: state controllability and eigenvalue assignability”, International Journal of Control, vol. 16, no. 1, pp. 101-104, 1972.

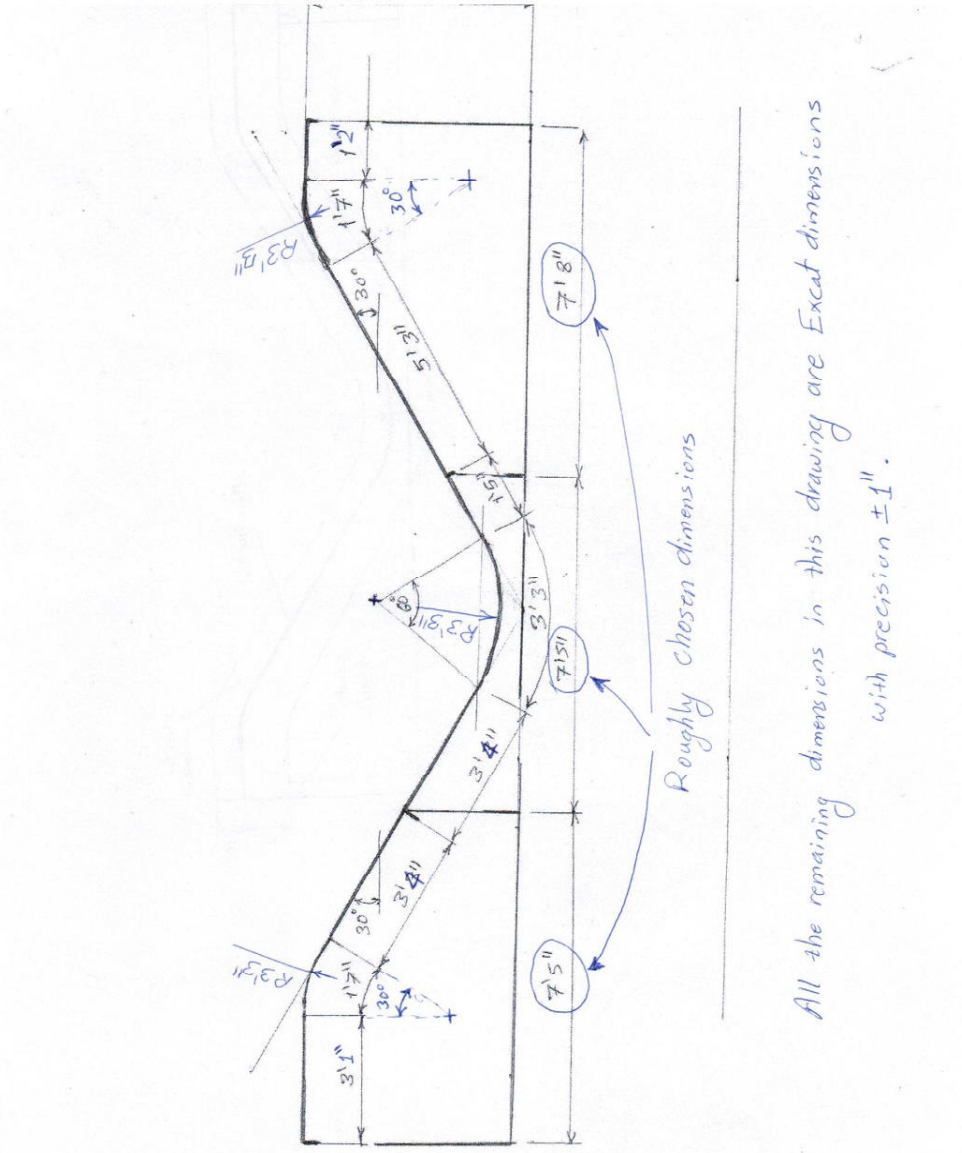
[47] K. Astrom, T. Hagglund, “PID control”, PID Controllers; Theory, Design, and Tuning, 2nd ed., ch 3, pp. 59-119, Instrument Society of America, Research Triangle Park, NC, 1995.

[48] Marques de Sa, Joaquim P., “Data regression”, Applied Statistics: Using SPSS, STATISTICA, MATLAB and R, 2nd ed., ch 7, pp. 275-328, Springer, Heidelberg, DEU, January 2007.

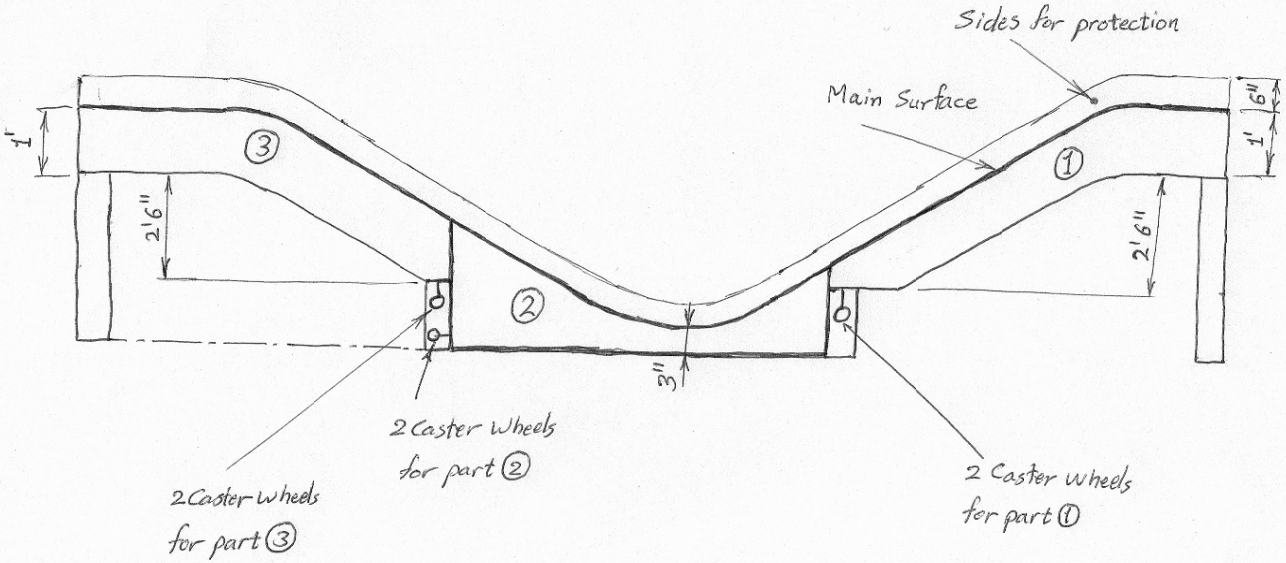
[49] The MathWorks® Inc. (2013 March) Evaluating goodness-of-fit. [Online]. http://www.mathworks.com/help/curvefit/evaluating-goodness-of-fit.html

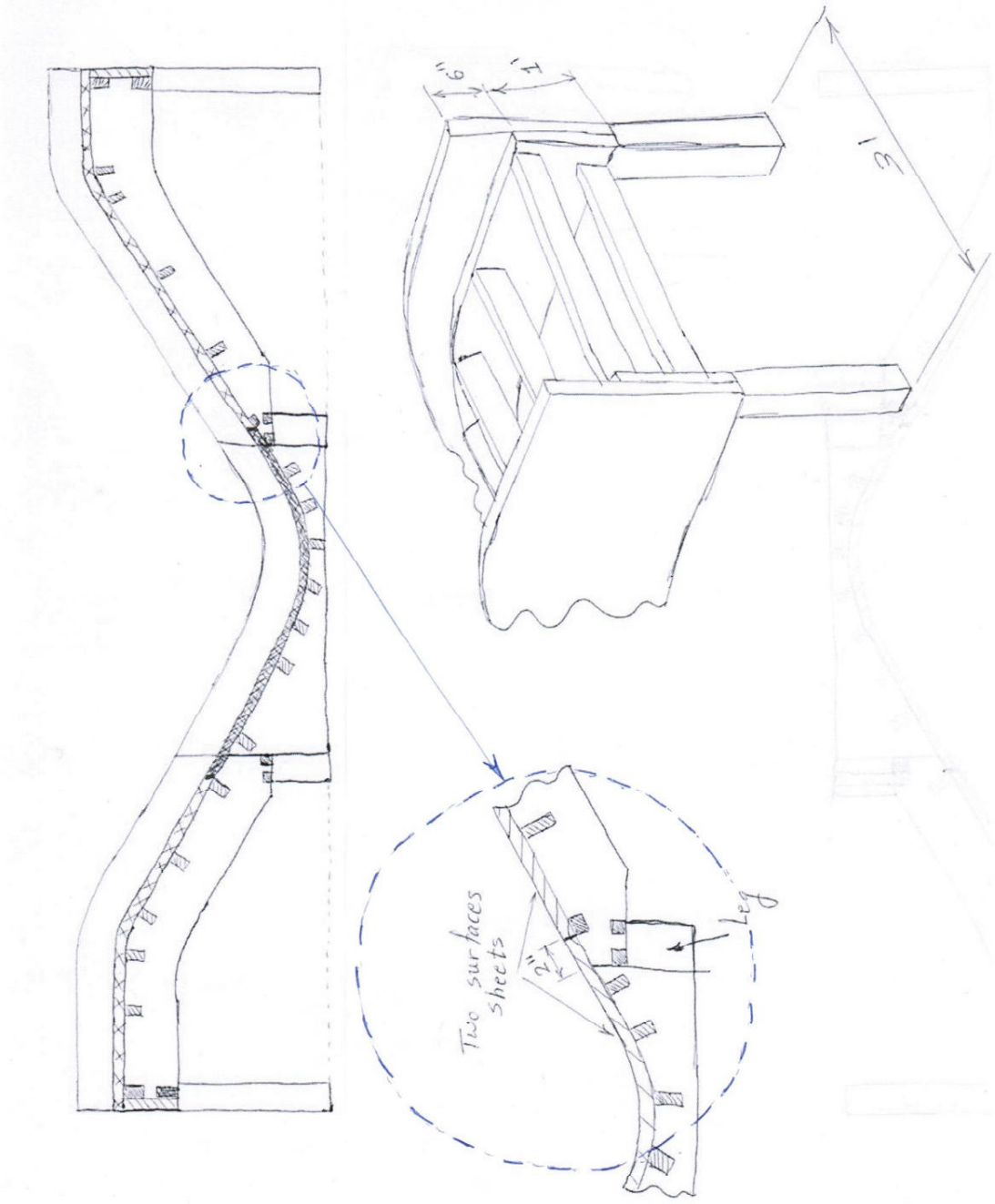
Appendix

Construction drawings of the Ramps:

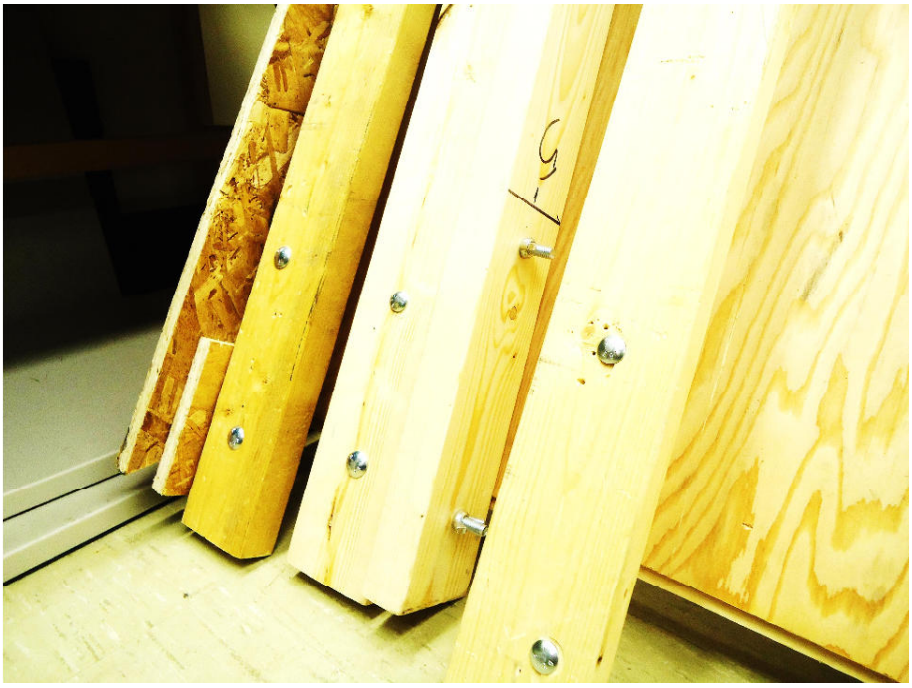


The total width of this ramp
is (3')

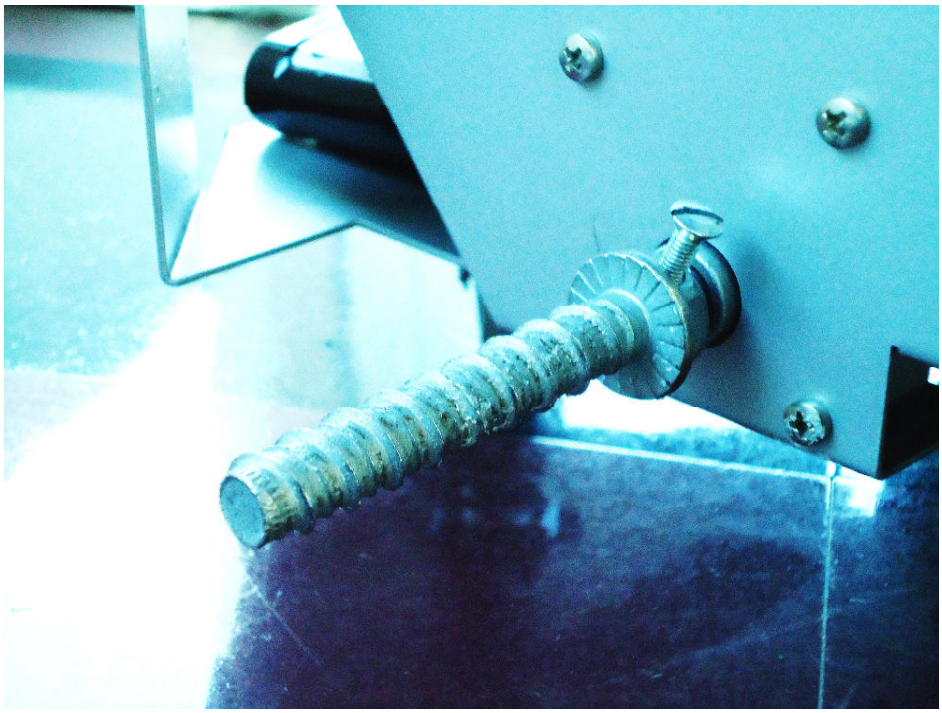


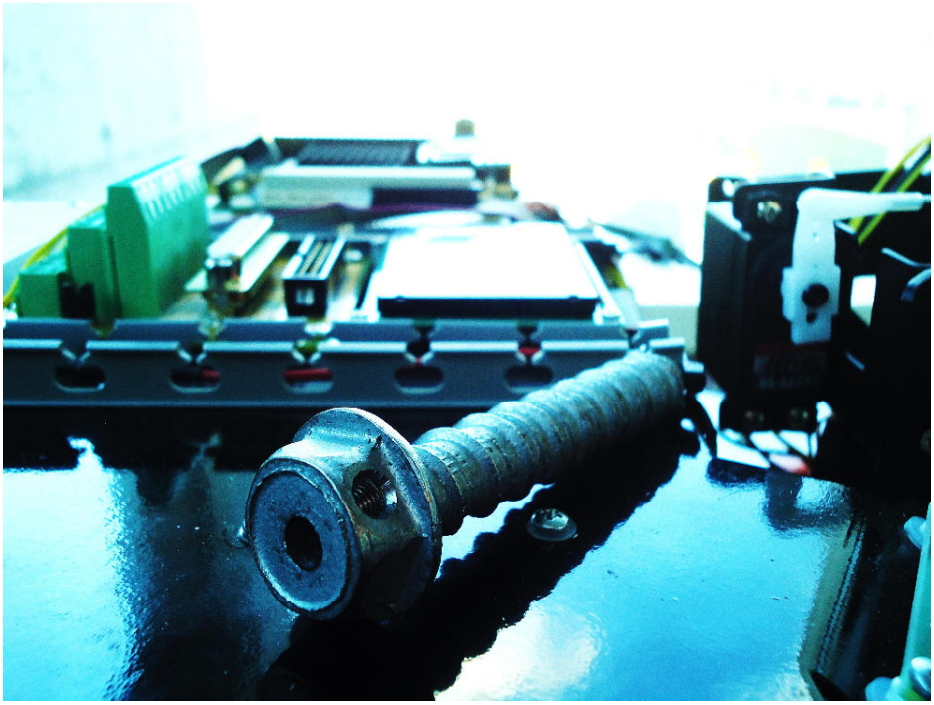

6"
1"
15"
Two surfaces sheets
2"
Leg

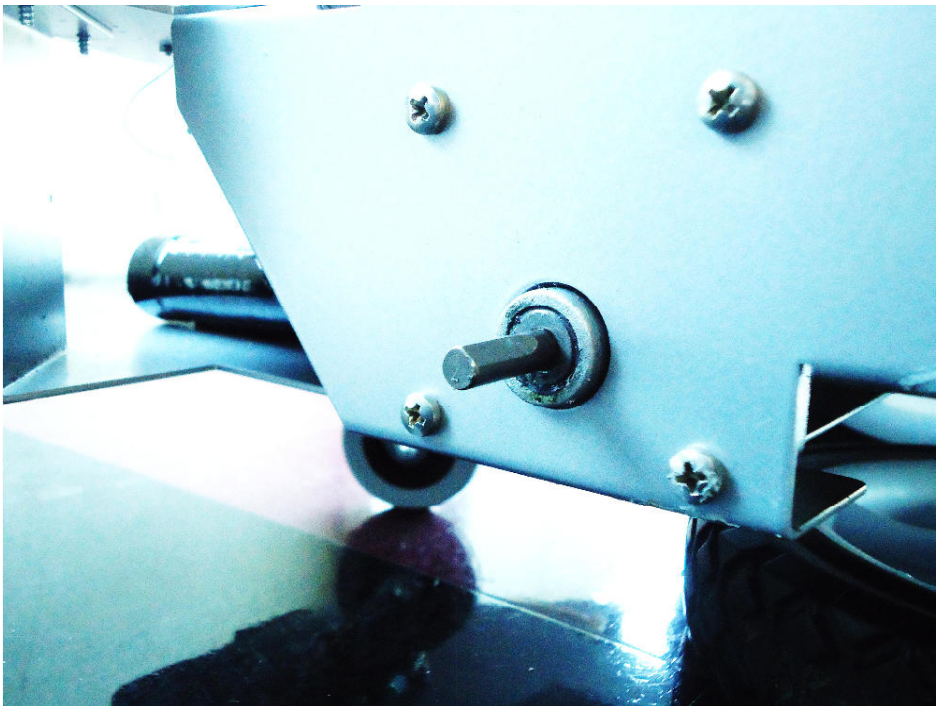
Photos show the ramps construction:



New wheel fixation:







Additional caster wheel fixation: