# EE6412: Optimal Control
# Project: Numerical and Experimental Implementation of Leapfrog Algorithm for Optimal Control of a Differential Drive Mobile Robot

Project Submission
R. Gautham, ED14B015
Guru Koushik, ED14B017
Akhil Sathuluri, ED14B037

May 5, 2017


## 1 Abstract

A kinematic model of a mobile robot is used to find optimal control law to go from an initial point and heading angle to a final destination point and a heading angle in a 2-D plane. This project uses Leapfrog Algorithm and Pontryagins Maximum Principle to solve for the optimal trajectory when an initial feasible trajectory from the source to destination is given. Since this is a problem which requires solving of a two point boundary value problem because the initial and terminal conditions are known, single shooting method is employed to solve the same.


## 2 Introduction

The cost that is to be minimised is the total control energy during the motion between the given two points. It physically means that the energy required is minimised between the points of interest. Also the final time is an unknown. The most important part is the initial guess of the costates which was found out using the initial conditions of the states and the fact that the costates are affinely related.

## 3 Leapfrog Algorithm

The Leapfrog Algorithm takes an initial feasible trajectory and divides it into n partitions and the optimisation objective and constraints for the full trajectory is formulated with

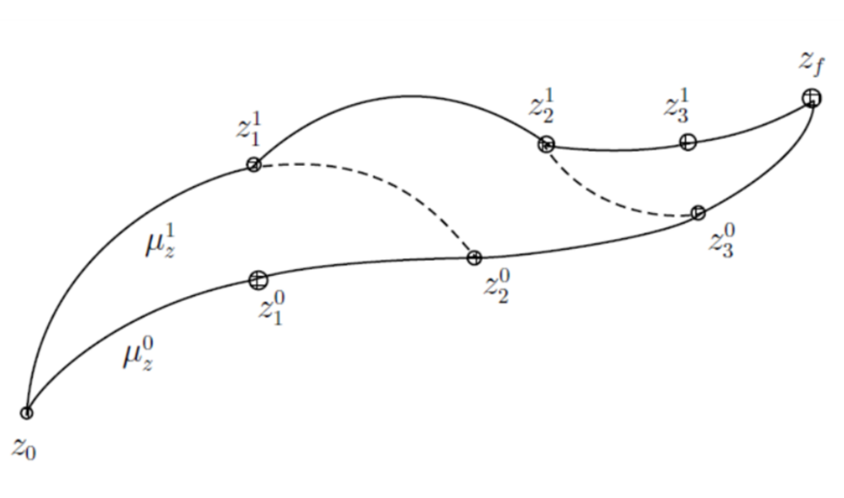


Figure 1: Method of update of leapfrog algorithm

the objective function as the quadratic function of the inputs and the constraints being the kinematic model of the mobile robot. Also a sub-problem is formulated which solves the optimisation problem from (i-1)th state to the (i+1)th state initial and final conditions as the values of the corresponding points in the feasible trajectory. This alogorithm gives us second order accuracy.

Also this algorithm is preferred as it is a symplectic algorithm, implies that the alorithm is area preserving. That is, the integration can be considered as an area map of the variables. Since area preservation is an exact feature of an equation it is preferred that the numerical approximation preserves it and such approximations are called symplectic. Hence this is an advantage of the Leapfrom algorithm[3]. The sub-problem Pi is solved from (i-1)th partition

$$P = \begin{cases} \text{minimize } \int_{t_0}^{t_f} f_0(x(t), u(t))dt \\ \text{subject to } \dot{x}(t) = f(x(t), u(t)), \\ \qquad x(t_0) = x_0,\ x(t_f) = x_f \end{cases}$$

Figure 2: The posed problem

point to (i+1)th partition point. The resulting trajectory obtained from solving Pi is used to update the value of ith partition point from that corresponding to the initial feasible trajectory. This process is successively followed till we reach the final state and get a set of updated state values at the partition points. These local optimal curves are concatenated and is considered as the initial feasible trajectory and the whole process is followed until the updated state values at the partition points is not changing more than a pre-set threshold value from the previous iterations state values. When this situation is encountered, the loop is terminated saying that the optimal trajectory has been found.

$$P_i = \begin{cases} \text{minimize } \int_{t_{i-1}}^{t_{i+1}} f_0(x(t), u(t))dt \\ \text{subject to } \dot{x}(t) = f(x(t), u(t)), \\ x(t_{i-1}) = z_{i-1},\ x(t_{i+1}) = z_{i+1}. \end{cases}$$

Figure 3: Subproblems solved in each iteration

## 4 Pontryagin's Maximum Principle (PMP)

PMP gives a set of necessary conditions to be satisfied to solve the optimal control problem which gives rise to a set of differential equations and boundary conditions of the states and co-states of the system. Solving these will give the optimal solution for minimising the cost function for the given constraints and boundary conditions. In the formulation of PMP, there is an abnormal multiplier which is taken to be -1 without loss of generality. The equations of PMP are as shown in Fig.4, Fig.5, Fig.6.

## 5 Single Shooting Method

Single shooting method is an alternate method to solve for the optimal trajectory by solving the two point boundary value problem instead of using the leapfrog algorithm. We have six variables and six equations which include three state vectors and three co-state vectors. This system needs six initial/boundary conditions for having a unique solution. What is available in this case is three initial conditions on the states and three final conditions on the states. This translates into solving a two point boundary value problem.

To solve this, we employ the single shooting method by making an initial guess for the co-state vectors and compute the difference between the desired final states and the final

$$\boxed{H(x^*(t), u^*(t), \lambda^*(t), t) \leq H(x^*(t), u, \lambda^*(t), t)}$$

Figure 4: Inequality satisfying the Hamilton at optimality

$$-\dot{\lambda}^{\mathrm{T}}(t) = H_x(x^*(t), u^*(t), \lambda(t), t) = \lambda^{\mathrm{T}}(t) f_x(x^*(t), u^*(t)) + L_x(x^*(t), u^*(t))$$

Figure 5: Canonical equation

$$\Psi_T(x(T)) + H(T) = 0$$

Figure 6: Costate final condition

states required from the guessing of initial co-state vectors. We add a perturbation to the initial co-state vector guess and find the Jacobian matrix for the variation of co-states. Using this Jacobian matrix, we update the co-states for every iteration with the difference in the co-states by the Newton-Raphson method. When the final state arrived from guessing and updating the co-states is very close to the desired final states, we terminate this loop and plot the optimal trajectory.

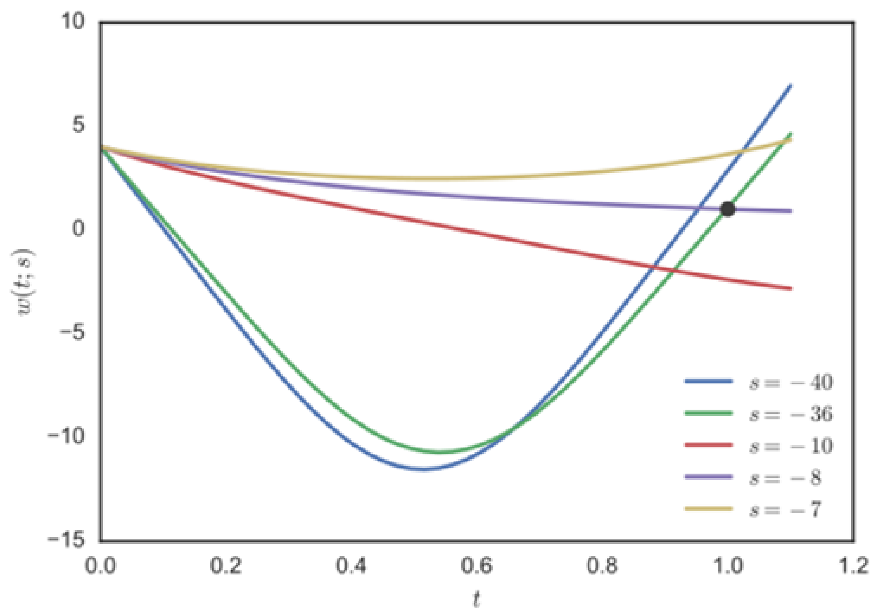


Figure 7: Plots for various 's'

## 6 Methodology

The kinematic model of the differential drive mobile robot is as shown in the figure. 8. The objective is to minimise the control energy given the end time. So the objective function is taken as: The Hamiltonian equation for this system is: Applying PMP, we get the following equations: Now the equation we get is a two point boundary value problem. To solve this, we use two methods: 1. Leapfrog algorithm 2. Single Shooting algorithm If Leapfrog algorithm is used, then an initial reference trajectory is taken and then iteratively an optimisation problem is solved and the trajectory is updated until the threshold is crossed. If the single shooting method is used, an initial guess is given and the guess is updated using Newton-Raphson method successively until the error is lower than a threshold value. After the two point boundary value problem is solved, the value of the initial co-states corresponding to the optimal trajectory is given as an initial value problem to the system model and then the state values are evaluated and then plotted.

## 7 Model Predictive Control

The control action derived using theory of optimal control is generally open loop in nature,i.e there is no feedback. Such control action are not preferred since they are not robust in nature. In order to make it more robust, a nonlinear MPC algorithm is coded. AN MPC algorithm was written and is attempted to implement but the parameters are yet to be tuned to achieve the final stabilising effect.

$$\dot{x} = v\cos\theta, \quad \dot{y} = v\sin\theta, \quad \dot{\theta} = \omega$$

Figure 8: Kinematic Model of the robot

Model predictive control is an control algorithm wherein the control actions are calculated based only on the present state value and nothing else. The basic algorithm is that it predicts the state over certain horizon and manipulates the control values such that the error obtained is minimised. Once we get a control value, this is used to propagate the system forward in time and this is repeated again and again.

# 8 Results

## 8.1 Initial Feasible Path

Explain how it is generated

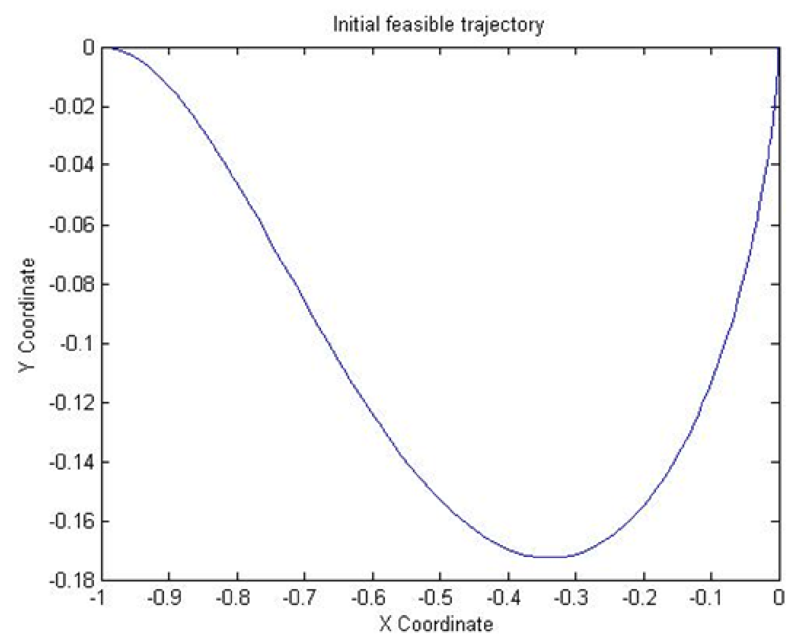


Figure 9: Feasible path generated

## 8.2 Optimal Path Generated

## 8.3 Variation of states

Interpret the stuff here

# 9 Future Scope

1. Current algorithm works only for a fixed source and destination values. The inputs are found for the initial feasible trajectory for a given source and destination and generalising this algorithm to find the trajectory for any given initial and final states should be done.

2. Implementing multiple shooting method to solve the two point boundary value problem which will give a better approximation of the trajectory and also ensure convergence more than the single shooting method.

3. Implement adaptive step size control of the descent to decrease the time of convergence to the solution.

4. Derive a closed loop optimal control to allow for real time implementation of the algorithm in robots.

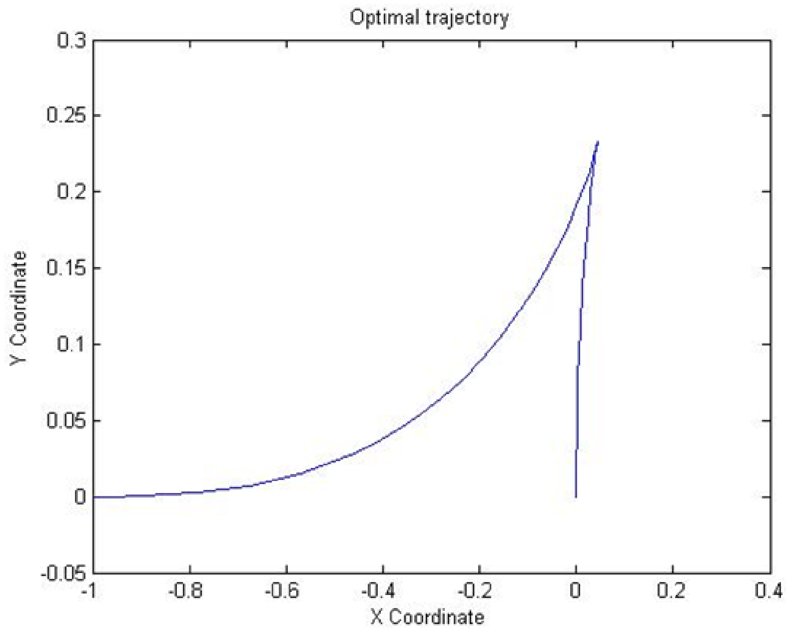


Figure 10: Optimal Path generated

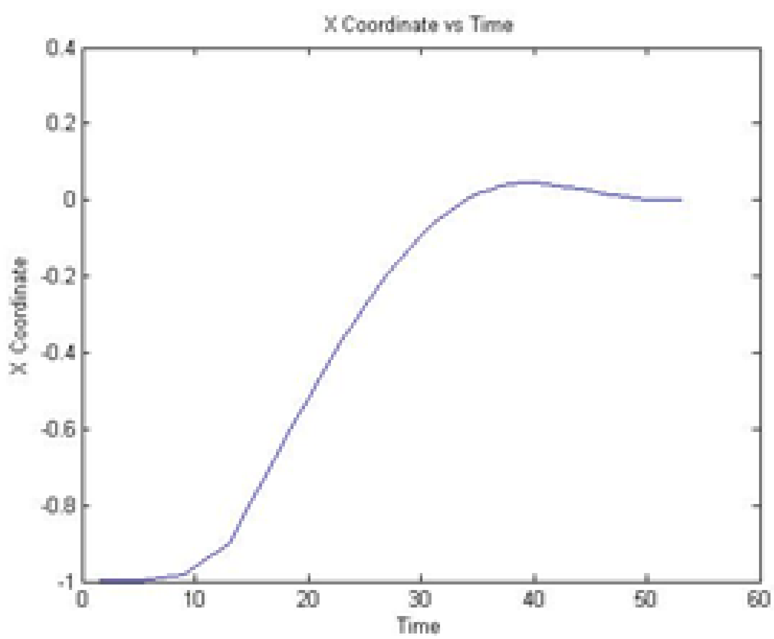


Figure 11: Variation of X-cordinate with time

# 10 References


1. Aditya Vamsikrishna, Arun D. Mahindrakar, Shaligram Tiwari, "Numerical and Experimental Implementation of Leapfrog Algorithm for Optimal Control of a Mobile Robot", Indian Control Conference, 2017

2. A. Astolfi, "Asymptotic Stabilisation of non-holonomic systems with discontinuous control", PhD thesis, Swiss Federal Institute of Technology, Zurich, 1996

3. Peter Young, "The leapfrog method and other symplectic algorithms for integrating Newtons laws of motion", Physics 115/242, 2014

4. D. Liberzon, Calculus of Variations and Optimal Control Theory, Princeton University Press, 2012

5. R.Bulirsch, "Introduction to Numerical Analysis", 3rd Edition, January 2002

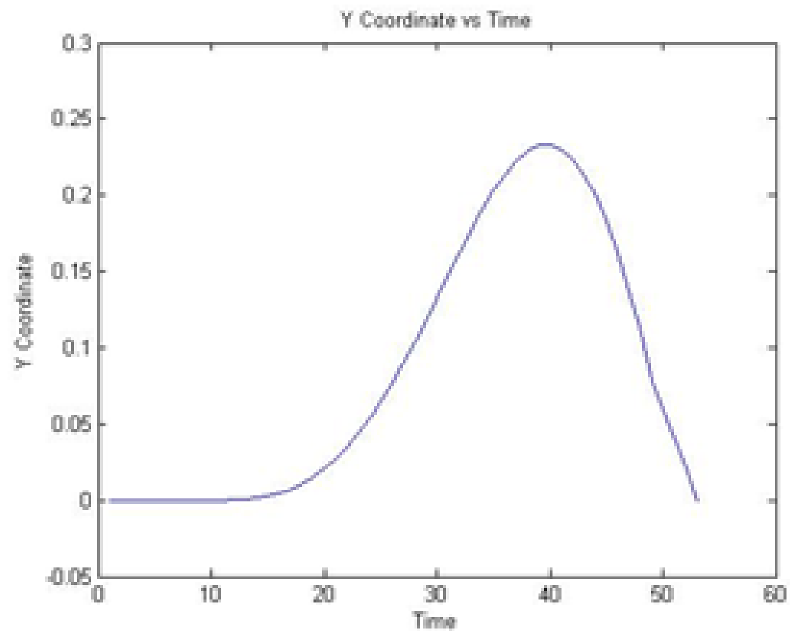


Figure 12: Variation of Y coordinate with time

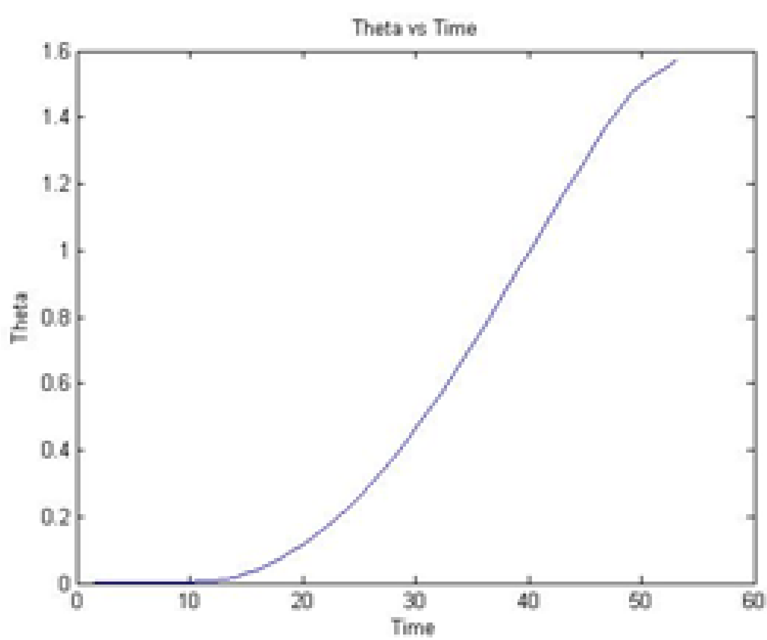


Figure 13: Variation of the angle with time