ISME 867

Name: Sudan Baral

WID: #_0470_

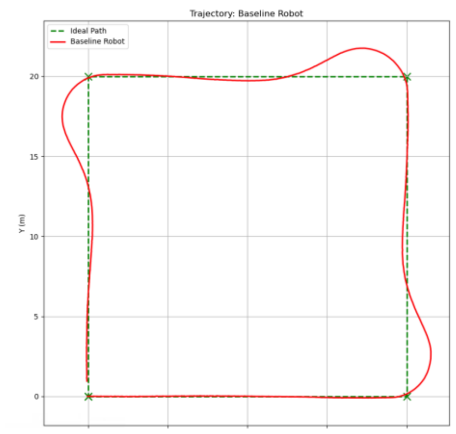
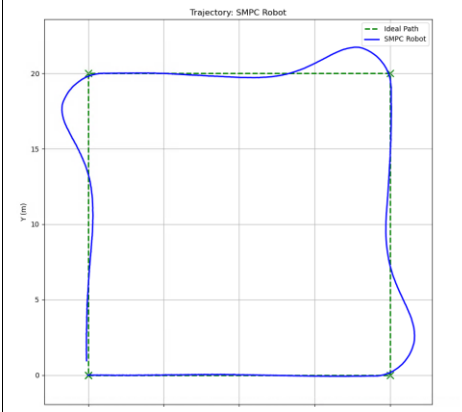
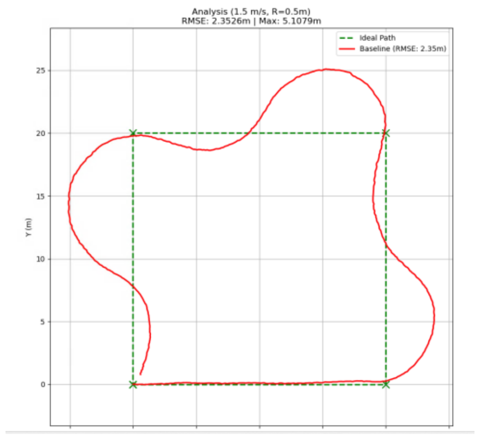
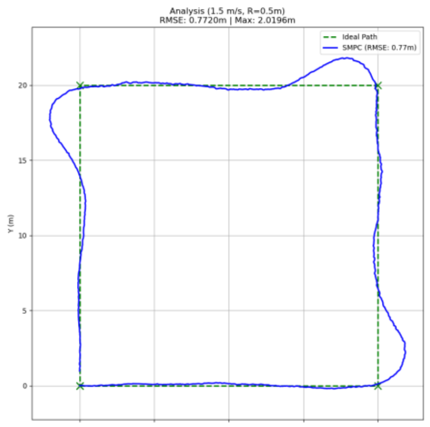
Final Project

Project: Waypoint navigation for a differential drive robot using Stochastic MPC

Overview

The objective of this project was to design a stochastic model predictive controller instead of a regular PID based controller and investigate error reduction in waypoint navigation. The whole system is based on ROS architecture and the simulation platform used is the physics engine Isaac-sim 5.0. Two different types of terrain, smooth plane and rough plane, were used to see the difference and how SMPC deals with the difference in terrain uncertainties. To test the motion, a 20m x 20m waypoint trajectory was selected with the robot velocity of 1.5m/s. The values used in this model are what are used in the robot used at Farmslab. Robot momentum and wheel slip are typically the core uncertainities kinematic models fail, which validated the use of physics engine capable of dynamic modeling and a stochastic controller to minimize such errors. We used a P-controller as a comparative baseline. From the experiment in Isaac sim, using same environment and same waypoint trajectory, reduction in root mean square error was observed with the use of SMPC by about 67% in rough terrain. Although the error difference was almost none in a perfectly smooth plane.

Standard controllers such as P-controller assume rolling without slipping which can be accurate if the ground plane is perfectly smooth and speeds are low. However, the robots in real life behave differently, they skid and slip in uneven terrain and high velocity. When standard P-controller is used for the latter case, the momentum resists the turning command and makes a larger curve during turning. Which is where the deterministic part fails. And this is where the stochastic controller aids in. Below is the trajectory traced by the robot using baseline deterministic controller and SMPC. The first row was done in low velocity 1m/s and perfectly smooth ground plane. Whereas the second row is trajectory traced on an uneven plane with robot velocity 1.5m/s.

	Baseline	SMPC
Smooth ground plane		
Rough terrain plane		

Stochastic MPC

The SMPC addresses this problem by treating the robots state as Gaussian distribution. To maintain itself in the lane for waypoint navigation, it adds uncertainty propagation in the covariance of the robots state.

kinematic state space

$$X = [x \quad y \quad \theta]^T$$

The discrete time update at step k with time-step dt is,

$$x_{k+1} = x_k + v_k \cos(\theta_k) dt$$

$$y_{k+1} = y_k + v_k \sin(\theta_k) dt$$

$$\theta_{k+1} = \theta_k + \omega_k dt$$

Uncertainity propagation
It is represented as a covariance matrix Σ which is a 3x3 matrix. To predict how it grows, we use EKF prediction step:

$$\Sigma_{k+1} = A_k \Sigma_k A_k^T + Q_k$$

where Σ_k is the current uncertainity

A_k is the jacobian of the robot position

A_k^T projects old uncertainity

Q_k is process noise (bumps due to uneven terrain)

$$A_k = \begin{bmatrix} 1 & 0 & -v\sin(\theta)dt \\ 0 & 1 & v\cos(\theta)dt \\ 0 & 0 & 1 \end{bmatrix}$$

where $-v\sin\theta$ and $v\cos\theta$ are sensitity, such that slight change in θ results in massive overall error in the robot position.

Normally Q_k is either constant or 0.

$$Q_k = Q_{base} \times (1 + \gamma \times \text{slip ratio})$$

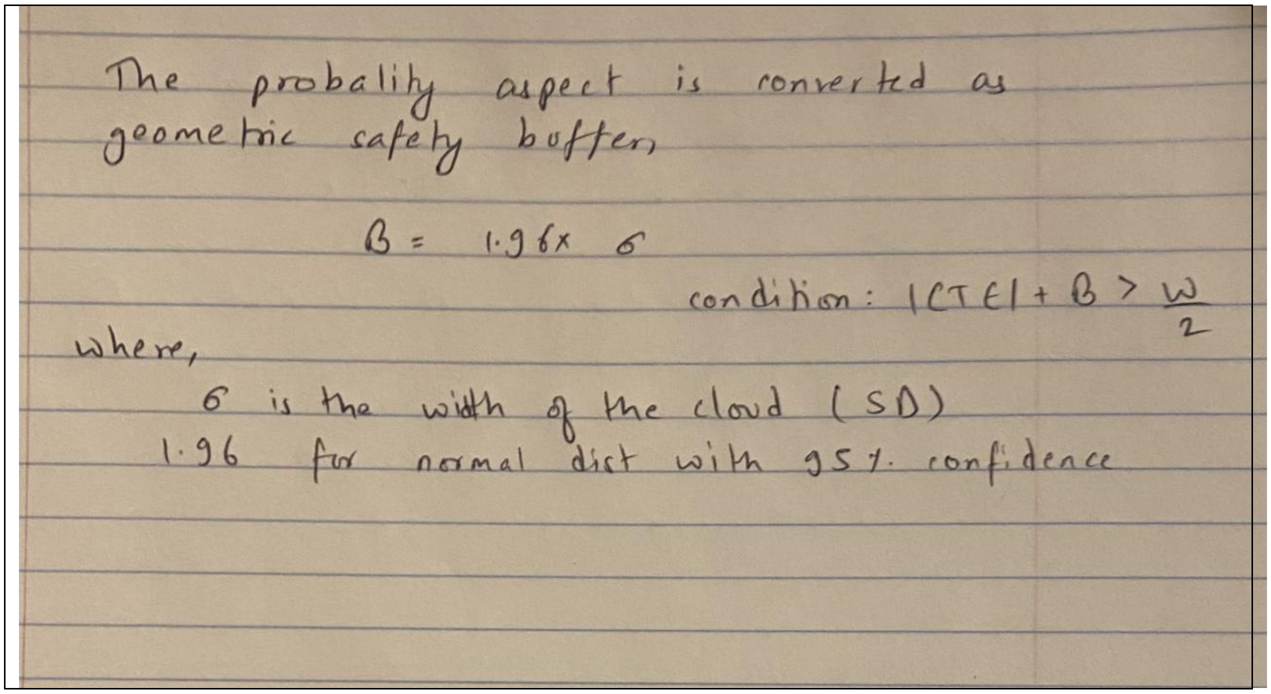
In this model, Q_k depends on the slip ratio. If the slip is small or almost 0, Q_k stays the same, if slip is high, Q_k increases too.

Thus, more slip results in larger covariance.

Finally, we use the covariance to make decision. For traversing, we define a lane of width w. In regular controller, the model just checks if the center of the robot is in the center of the lane. ~~A~~ The SMPC checks if the entirity of the gaussian is in the lane. For that a chance constraint is enforced,

$$P(|CTE| > W/2) \leq e$$

Where CTE is the cross track error or distanc from the center and $e = 0.05$ for 95% confidence.



The probality aspect is converted as geometric safety buffer,

$$B = 1.96 \times \sigma$$

condition: $|CTE| + B > \frac{w}{2}$

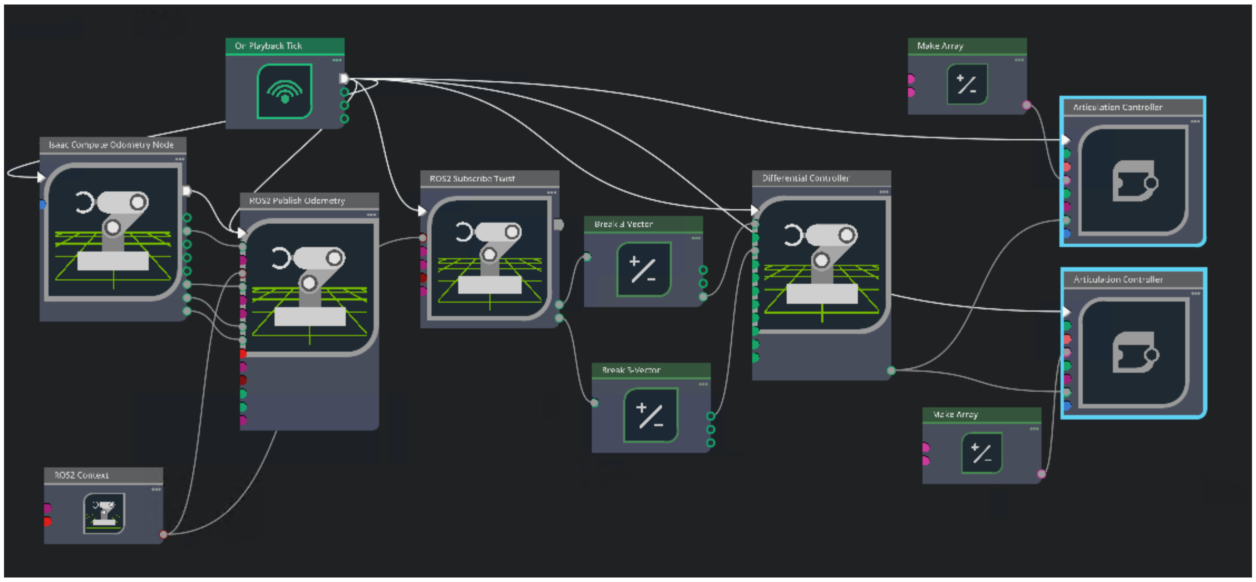
where,

σ is the width of the cloud (SD)

1.96 for normal dist with 95% confidence

ROS architecture

The Isaac sim model is populated with a ROS publisher which publishes real-time odometry data, a ros subscriber which subscribes command velocity and a differential drive articulation controller. The published odometry data is intercepted by our python node which runs in ROS environment and uses it to publish command velocity data in real time. The subscriber node in the Isaac-sim intercepts the command velocity to move the robot. The layout looks like this:



Finally, root means square error(RSME) was calculated for regular baseline controller and the SMPC and is tabulated as follows:

Iteration	Baseline model(P-controller)	SMPC	Improvement
1	RMSE (Average Drift): 2.3526 meters Max Error (Worst Case): 5.1079 meters	RMSE (Average Drift): 0.7720 meters Max Error (Worst Case): 2.0196 meters	SMPC: RMSE reduced by 67.18% SMPC: Max error reduced by 60.46%
2	RMSE (Average Drift): 2.4478 meters Max Error (Worst Case): 5.7390 meters	RMSE (Average Drift): 0.7975 meters Max Error (Worst Case): 2.2157 meters	SMPC: RMSE reduced by 67.42% SMPC: Max error reduced by 61.39%