Dynamic Path Planning using Stream function, Gaussian Approach and MPC

Mohammed Tousif Zaman*, Mitesh Agrawal† and Dr. Jie Fu‡
Robotics Engineering, Worcester Polytechnic Institute
Worcester, MA, USA
Email: *mzaman@wpi.edu, †msagrawal@wpi.edu, ‡jfu2@wpi.edu

***Abstract*—Safe navigation is one the biggest concerns in the research of autonomous vehicles. Most imortant sub-parts of navigation problem are obstacle path prediction, robot path planning and trajectory tracking. Path planning using holomorphic function for obstacle avoidance is an alternative overcoming the drawbacks of conventional potential field methods. Model Predictive Control (MPC) method is highly effective in tracking the desired path whilst respecting the vehicle dynamics. This paper discusses stream function based path planning, obstacle path prediction using gaussian approach and implementation of a model predictive controller on a Turtlebot robot. Experimental results are available from simulation.**

***Index Terms*—Path planning, Stream Function, Gaussian Process, Receding Horizon Control, Model Predictive Control**

I. INTRODUCTION

Today we are closer than ever to seeing the autonomous car become a reality. Internet giants like Google to automaker Nissan are working on autonomous technology, the streets may soon be filled with driverless cars [1]. A study conducted by Columbia University reports that replacing a fleet of 13,000 yellow cabs with 9,000 driverless cars in New York could cut costs per mile by nearly 88% and wait times down from 5 minutes to just 36 seconds during rush hour [2]. One of the biggest challenge for an autonomous vehicle is the planning of a reference trajectory with accuracy to guarantee the safety of passengers as well as the surrounding environment comprising other vehicles on the road, pedestrians and objects on the path.

We see that the general problem of safe navigation could be subdivided into robot path planning, obstacle path prediction and robot path tracking. Robot path planning is essential to solve the problem of generating a path from start to the goal avoiding any obstacles on the path. The desired trajectory must be designed to avoid collisions with obstacles on the path of the robot and guarantee a safe vehicle behavior. We seek to achieve obstacle avoidance in the robot path planning stage further enhanced probabilistically by predicting the obstacle trajectory in advance. Finally, the desired path from the planning stage can be tracked using a predictive control method such as Model Predictive Control (MPC).

There are several methods to solve the path planning and tracking problem while considering the uncertainty associated with the obstacles. The idea of Timed Elastic Bands (TEB) presented in [3] as a method to deform and optimize the robot trajectory generated by the global planner in order to get around the obstacles on the path. This method is not always able to move across obstacles and can get stuck in a local optimal trajectory.

Potential field based control is another method to reach a goal while avoiding the obstacles. [4] solves the problem of a moving target tracking and avoidance of moving obstacles using potential field functions considering the goal, velocity information of the robot and obstacles. The attractive force of the target brings the robot towards the goal while the repelling force of obstacles aid in collision avoidance. These forces are computed as negative gradient of the respective potentials defined in terms of both position and velocity of robot relative to obstacles. This method suffers from the local minima problem and the robot could get stuck at the local minima in the field instead of reaching the goal.

Inspired by fluid dynamics, Akishita et al. [5] describe a local minima free approach by expressing the potential field as a harmonic function. Waydo and Murray [6] further extend this approach using the Circle Theorem [7] to show how multiple circular obstacles could be added to the flow without distorting the streamlines. This method generates smooth paths that are suitable for dynamics of second order systems. They show results with obstacle avoidance achieved for both static and moving obstacles. For such systems where the obstacle trajectories change at every time step depending on the changing conditions of the road, predictive control methods like MPC would be the best option for tracking the desired robot trajectory. The essence of MPC is to optimise forecasts of process behaviors [8]. A Linear MPC controller is employed to converge to the goal considering the vehicle dynamics.Our paper builds on this idea by incorporating obstacle trajectory prediction and including a tracking control based on MPC in addition to the stream function based path planning.

To obtain probabilistic confidence on avoidance of obstacles is not a new concept. [9] shows stochastic model of robot and obstacle used to avoid obstacles using Model Predictive Control. The paper uses Bayesian Policy Optimisation and claims to achieve real-time obstacle avoidance. On similar lines, [10] explains how chance constraints can be utilized along with MPC for obstacle avoidance. They calculate the overlap between the gaussians representing uncertainities of obstacle and robot using Bhattacharya distance and use it to

The authors are with the Robotics Engineering Program, Department of Electrical and Computer Engineering, Worcester Polytechnic Institute, Worcester, MA, 01609, USA

avoide collision. In [11], Wang shows gaussian processes are fast and accurate enough to predict movements of humans. He compares various probablistic techniques such as Gaussian Process regression, Local Gaussian Process and Support Vector Regression for uncertainity prediction.

The organization of this paper is as follows: The preliminaries are introduced in section II. Path planning using stream function is discussed in section III. In section IV, we discuss obstacle trajectory prediction using gaussian process approach. The linear MPC control design has been described in section V. The experimental results in section VI show that the system converges well in simulation but we await to see the performance in real-time on a Turtlebot.

II. Preliminaries

This section describes some preliminary concepts and lists the equations which are useful to understand the subsequent sections. The proofs for the fluid mechanics concepts can be found in [7].

A. Potential Flow

In fluid mechanics, incompressible and irrotational fluid flow is described in terms of a velocity potential. The flow velocity is given by the gradient of the potential field that describes the flow. In two dimensions, the flow in x-y plane is described using complex analysis.

B. Complex Function

A complex variable is given by the below equation

$$z = x + iy \tag{1}$$

where, z represents the complex variable in terms of the cartesian coordinates x and y. The spatial coordinates x and y are real-valued, they define the two-dimensional space in which the robot and obstacles are located. A function of this complex variable z is called a complex function $F(z)$.

C. Velocity Potential and Stream Function

The definition of complex variable is used in the complex function to get the real-valued 2D functions ϕ and ψ. For instance, consider the complex function below,

$$F(z) = z^2 \tag{2}$$

$$F(x+iy) = (x+iy)^2 = (x^2 - y^2) + i(2xy) \tag{3}$$

$$F(x+iy) = \phi(x,y) + i\psi(x,y) \tag{4}$$

$$\phi(x,y) = x^2 - y^2 \tag{5}$$

$$\psi(x,y) = 2xy \tag{6}$$

The functions ϕ and ψ form a conjugate pair for a given complex function. The real part of the complex function, ϕ represents the potential field also called the velocity potential while the imaginary part, ψ represents the stream function. The lines of constant ϕ are called equipotential lines which can also be obtained by taking the contours of the velocity potential ϕ. The lines of constant ψ are called streamlines which are also the contours of the stream function ψ. The equipotential lines are orthogonal to the streamlines.

D. Cauchy-Riemann Relation

The complex function $F(z)$ is holomorphic and taking the derivative of 4 by z shows that it satisfies the Cauchy-Riemann equations.

$$\frac{dF}{dz} = \lim_{|\delta z| \to 0} \frac{F(z + \delta z) - F(z)}{\delta z} \tag{7}$$

Substituting $\delta z = \delta x$ in 7 gives the relation,

$$\frac{dF}{dz} = \frac{\partial \phi}{\partial x} + i \frac{\partial \psi}{\partial x} \tag{8}$$

Substituting $\delta z = i\delta y$ in 7 gives the relation,

$$\frac{dF}{dz} = -i \frac{\partial \phi}{\partial y} + \frac{\partial \psi}{\partial y} \tag{9}$$

This results in the below relations,

$$\frac{\partial \phi}{\partial x} = \frac{\partial \psi}{\partial y} \tag{10}$$

and,

$$\frac{\partial \psi}{\partial x} = -\frac{\partial \phi}{\partial y} \tag{11}$$

E. Laplace Equation

By taking the derivative of the Cauchy-Riemann relation in 10, we get a new relation,

$$\frac{\partial^2 \phi}{\partial x^2} = \frac{\partial}{\partial x}\frac{\partial \psi}{\partial y} = \frac{\partial}{\partial y}\frac{\partial \psi}{\partial x} = \frac{\partial}{\partial y}\Big(-\frac{\partial \phi}{\partial y}\Big) = -\frac{\partial^2 \phi}{\partial y^2} \tag{12}$$

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = 0 \tag{13}$$

Similarly, taking derivative of 11 and following the steps in 12 gives,

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = 0 \tag{14}$$

The new relations in 13 and 14 show that ϕ and ψ satisfy the Laplace equations in 2-D space.

F. Velocity Components in 2-D plane

The Cauchy-Riemann relations represent the velocity components of the flow velocity field $V = (u, v)$ in the x-y plane.

$$V_x = u = \frac{\partial \phi}{\partial x} = \frac{\partial \psi}{\partial y} \tag{15}$$

$$V_y = v = \frac{\partial \phi}{\partial y} = -\frac{\partial \psi}{\partial x} \tag{16}$$

G. Complex Potential and Circle Theorem

Some useful primitives to describe the field are source/sink and vortex. The respective complex potentials are denoted by $f_s(z)$ and $f_v(z)$ respectively,

$$f_s(z) = C \ln(z) \tag{17}$$

$$f_v(z) = Ci \ln(z) \tag{18}$$

C represents the strength of source/sink potential. C is positive for a source and negative for a sink. The real and imaginary parts of the source/sink potential can be swapped to get the vortex potential. Obstacles can be represented either by a vortex or by doublets. A doublet is formed by placing both source and sink very close to each other. These primitives could be combined to generate potential fields and hence streamlines in the flow. Circle Theorem described by Waydo and Murray in [6] allows addition of circular shaped obstacles without distoring the existing streamlines in the flow. The below condition must be satisfied for such a flow,

$$f_{total}(z) = f(z) + \overline{f(\overline{z})} \tag{19}$$

III. Path Planning using Stream Functions

A. Motivation

Stream function on a 2D space can be visualized as a stream of water flowing to a given point, say, the sink, cutting across large stones on the path. The flow is due to the attraction force generated by artificial potential field of the sink while the repelling force generated by the field of obstacles is attributed to the collision avoidance behavior. The resulting streamlines follow the gradient of the artificial potential field. Stream function has the below properties which makes it better suitable for path planning than most potential field based methods.

- These functions are harmonic (twice continuously differentiable) i.e. they satisfy Laplace equations. Due to this, they do not suffer from local minima problem.
- The paths generated are smoother and preserve nonlinear dynamics of 2nd order systems.

Given these properties, a robot following the trajectory generated over the gradient of the potential field will end up reaching the goal from any starting position and also avoiding the obstacles. The field is formed by a sink (goal) and multiple obstacles which could be stationary or moving at constant velocities in the x-y plane. Similar to [6], the following subsections describe the equations necessary to model the stream function for circular shaped robot and obstacles.

B. Stationary Obstacles

Complex potential in this case is given by Circle Theorem as the sum of the potentials due to the sink and the stationary obstacles. The obstacle is centered at (bx, by) and has radius equal to a.

$$w = -C \ln z - C \ln \frac{a^2}{z - b} + b \tag{20}$$

The stream function is given by the imaginary component of the complex potential.

$$\psi = -C \tan^{-1}\left(\frac{y}{x}\right) + C \tan^{-1}\left(\frac{\frac{a^2(y-b_y)}{(x-b_x)^2+(y-b_y)^2} + b_y}{\frac{a^2(x-b_x)}{(x-b_x)^2+(y-b_y)^2} + b_x}\right)$$

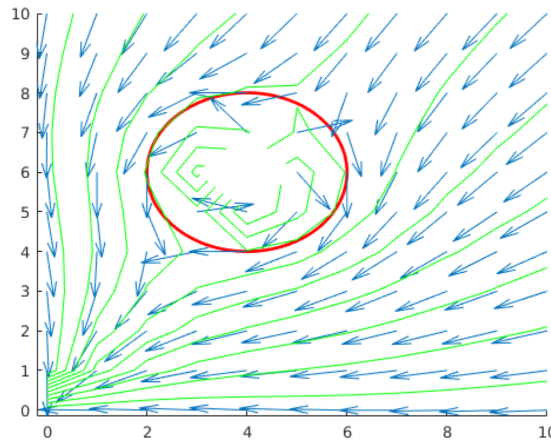

Fig. 1. Streamlines and velocity vector field for a single stationary obstacle

C. Moving Obstacles

Complex potential $w(z)$ in this case is given by the sum of the potentials due to the sink and the moving obstacles. The moving obstacle is centered at (bx, by) with radius equal to a and is moving at a constant velocity $v = (v_x + iv_y)$.

$$w(z) = w_s(z) - v_x\left(\frac{a^2}{z-b} + \bar{b}\right) - iv_y\left(\frac{a^2}{z-b} + \bar{b}\right) \tag{21}$$

where $w_s(z)$ is the complex potential for the stationary case. The stream function is given by the imaginary component of $w(z)$.

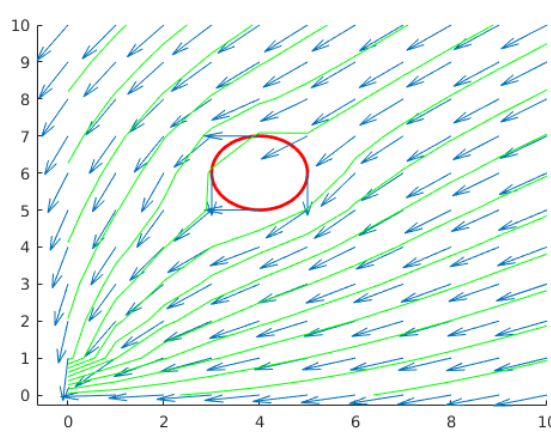

Fig. 2. Streamlines and velocity vector field for a single moving obstacle

IV. Trajectory Prediction using Gaussian Process

Gaussian process regression (GPR) models are nonparametric kernel-based probabilistic models [12]. An instance of response y (output train) from a Gaussian process regression (GPR) model can be modeled as

$$P(y_i|f(x_i), x_i) = N(y_i|h(x_i)^T\beta + f(x_i), \sigma^2) \tag{22}$$

where x_i and y_i are the input and output data train for the GPR, respectively. In order to calculate coefficient vector β, the kernel parameters or hyperparameters θ and the noise variance σ^2, we use the standard GPR method as mentioned below.

$$\hat{\beta}, \hat{\theta}, \hat{\sigma^2} = argmax_{\beta,\theta,\sigma^2} logP(y|X, \beta, \theta, \sigma^2) \tag{23}$$

Then we can predict future values of y-train for given values of x-train using

$$P(y_{new}|f(x_{new}), x_{new}) = N(y_{new}|h(x_{new})^T\beta + f(x_{new}), \sigma^2) \tag{24}$$

In our model, we use four GPR models - positions x, y, velocity in x and velocity in y of the obstacle to predict the obstacle trajectory in future for given planning horizon. We use the moving horizon GPR i.e. we keep movig the window of train data to GPR at each step to avoid over fit problems. The training and prediction of trajectory is done at each step of the planning.

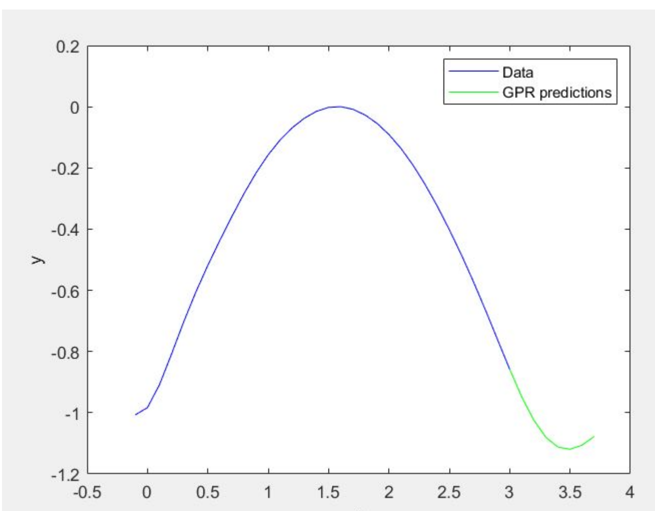

Fig. 3. Prediction of the obstacle trajectory from historical data using GPR

V. Linear Model Predictive Controller Design

Obstacles in the environment are unpredictable and they could deviate from the predicted trajectory at any step so we need to plan the trajectory at each step of the horizon. The controller must be able to accomodate such changing trajectory at each step. Model predictive control (MPC) is an advanced technique of process control that takes care of constraints along with model dynamics and past control signals while generating the new control input for the system. The main advantage of MPC is that it provides the current output by optimizing the future outputs that can generate the best result in that particular system.

We are considering the non-linear kinematic bicycle model. The bicycle model being a simple one, allows for steering of only the front wheel and as a result, the front steering angle δ and the velocity v are the only control inputs.

$$\dot{\mathbf{z}} = \begin{cases} \dot{x} = v\cos\theta \\ \dot{y} = v\sin\theta \\ \dot{\theta} = \dfrac{v}{L}\tan\delta \end{cases} \tag{25}$$

or, in a compact form as,

$$\dot{\mathbf{z}} = f(\mathbf{z}, \mathbf{u}) \tag{26}$$

Here,z is the state vector that comprises of x and y represent the center of mass of the vehicle, δ gives us the steering angle of the car which is one of the control inputs.

In the numerous Non-linear MPC (NMPC) schemes proposed in the literature [13], one important point to note is that the computation effort necessary is much higher than the linear MPC version. NMPC is a non-linear, non-convex problem which needs to be solved online and it is difficult to find the solution in general. In the linear MPC approach, the fundamental idea employed is the successive linearization approach which yields a time varying system. The decision variables considered here are the inputs, which are transformed into an optimization problem to be solved at each sampling time in a quadratic programming (QP) problem. Since these are convex in nature, QP problems can be solved easily by algorithms which are numerically robust that lead to globally optimal solutions.

We can obtain a linear model of the system dynamics by computing an error model with respect to a reference trajectory. We can expand the state model in the taylor series form around a desired reference point (z_r,u_r) and discard the higher order terms that follow. This is also known as "Jacobian Linearization".

$$\dot{\mathbf{z}} = f(\mathbf{z_r}, \mathbf{u_r}) + \left.\frac{\partial f(\mathbf{z}, \mathbf{u})}{\partial \mathbf{z}}\right| \tag{27}$$

or,

$$\dot{\mathbf{z}} = f(\mathbf{z_r}, \mathbf{u_r}) + f_{\mathbf{z},\mathbf{r}}(\mathbf{z} - \mathbf{z_r}) + f_{\mathbf{u},\mathbf{r}}(\mathbf{u} - \mathbf{u_r}) \tag{28}$$

where $f_{\mathbf{z},r}$ and $f_{\mathbf{u},r}$ are the jacobians of f with respect to $\mathbf{z}$ and $\mathbf{u}$, respectively. These jacobians are evaluated around the reference point $(\mathbf{z_r}, \mathbf{u_r})$. When we subtract 27 from 28 we get,

$$\dot{\tilde{z}} = f_{\mathbf{z},\mathbf{r}}\tilde{z} + f_{\mathbf{u},\mathbf{r}}\tilde{\mathbf{u}} \tag{29}$$

where, $\tilde{\mathbf{z}} = \mathbf{z} - \mathbf{z_r}$ is the error with respect to the reference and $\tilde{\mathbf{u}} = \mathbf{u} - \mathbf{u_r}$ is the error in the control input. The discrete-time system model is given by the approximation of $\dot{\mathbf{z}}$ using the forward differences:

$$\tilde{\mathbf{z}}(k+1) = \mathbf{A}(k)\tilde{\mathbf{z}}(k) + \mathbf{B}(k)\tilde{\mathbf{u}}(k) \tag{30}$$

with

$$\mathbf{A}(k) = \begin{bmatrix} 1 & 0 & -v_r(k)\sin\theta_r(k)T \\ 0 & 1 & v_r(k)\cos\theta_r(k)T \\ 0 & 0 & 0 \end{bmatrix} \tag{31}$$

$$\mathbf{B}(k) = \begin{bmatrix} \cos\theta_r(k)T & 0 \\ \sin\theta_r(k)T & 0 \\ \dfrac{\tan\delta(k)T}{L} & \dfrac{v_r \sec^2\delta(k)T}{L} \end{bmatrix} \tag{32}$$

Given this information, we can now formulate the cost function and constraints for MPC. MPC considers the trajectory tracking as an optimization problem which could be achieved in steps over a finite time horizon T subject to constraints on the system state and control input.

Assume $z(0)$ and $z(T)$ to be the initial and final states respectively, of the car and given a set of predicted trajectories, we obtain for each trajectory, a cost J that is a quadratic function of the error in the system state z and the control input u. The optimization problem is solved by minimizing the cost J . The equations of the linearized system, linear constraints and the quadratic cost function in case of Linear MPC as explained in [14] are:

$$\begin{aligned}
\min_u & \Bigg[\sum_{i=0}^{N-1} (z_i - z_{ref,i})^2 Q + \\
& \sum_{i=0}^{N-1} [(u_i - u_{ref})^2 R + (z_N - z_{ref,N})^2 Q_f \Bigg] \\
\text{s.t. } & z_0 = z_t, u_{-1} = u(t - t_s) \\
& z_{i+1} = Az_i + Bu_i, \;\; i = 0, ..., N-1 \\
& u_{min,i} \le u_i \le u_{max,i} \;\; \forall i \\
& z_{min,i} \le z_i \le z_{max,i} \;\; \forall i
\end{aligned}$$

VI. Experiments and Simulation Results

This section includes results from simulations demonstrating the effectiveness of stream function in obstacle avoidance with obstacle path prediction and further when tracking with MPC is included. The simulations consider a single obstacle centered at a given point in x-y plane and moving with a constant velocity in the x and y directions, respectively. The results include the predicted trajectory of the obstacle using Gaussian process and the desired path tracking using the Linear MPC controller with time step $t_s = 0.1$s, horizon length $= 7$.

A. Stream function with Deterministic GPR

The below figure shows the robot avoiding collision with a single moving obstacle. The obstacle (in red) is moving from $(5, 5)$ at a constant velocity $V_x = 0.5$ units/s and $V_y = 0$ units/s towards the robot which starts at $(9, 5)$. The predicted trajectory of the obstacle is a straight line while the robot executes the trajectory as determined by the streamline. This simulation is done considering the particle dynamics. When we apply these conditions to actual robot it is mandatory to consider the kinematic model of robot which will induce some errors while tracking the trajectory.

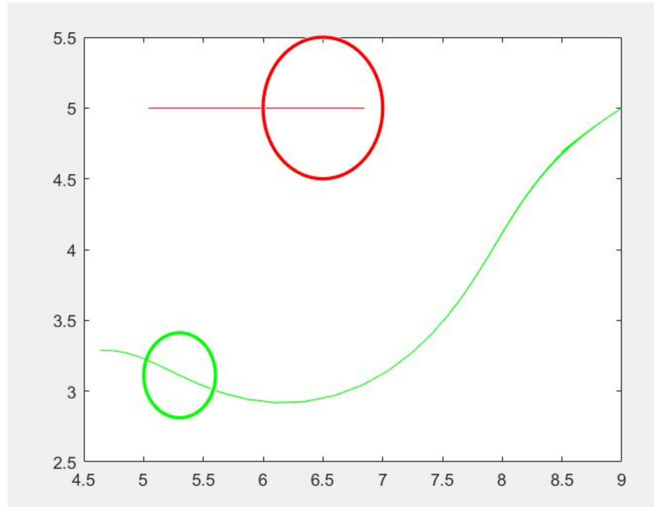

Fig. 4. Stream function with Deterministic GPR for a single stationary obstacle

B. Stream function with Deterministic GPR and Linear MPC

Initial planning is done using the stream function and the resulting velocity gradients are used to obtain the coordinates in cartesian space. The coordinates saved over successive time steps lead to the resulting trajectory for the robot and the obstacle. The below figure shows the robot avoiding a head-on collision with a single aggressive obstacle. The predicted trajectories using GPR are visible within the circles. The obstacle (in red) is moving from $(5, 5)$ at a constant velocity $V_x = 1.5$ units/s and $V_y = -0.55$ units/s towards the robot (in green) which starts at $(9, 5)$. The predicted trajectory of the obstacle is a straight line while for the robot, it keeps shifting downwards as MPC updates the control effort to track a different streamline in order to avoid collision with the obstacle. The inclusion of MPC considers the dynamics of the robot thereby resulting in a dynamically feasible robot trajectory.

The actual and predicted trajectory of the robot can be seen in the below figure. The robot ends up at the goal (origin) with the tracked path by LMPC being very close to the predicted path by GPR.

VII. Conclusion and Future Work

We see scope for future work in the below areas.

- Evaluate effect of source/sink strength on the speed and planned trajectory of the robot.
- Achieve desired velocity of the robot on the planned trajectory.
- Better prediction of obstacle trajectory using multivariate gaussian process.
- Quantify the levels of safety in obstacle avoidance with this approach.

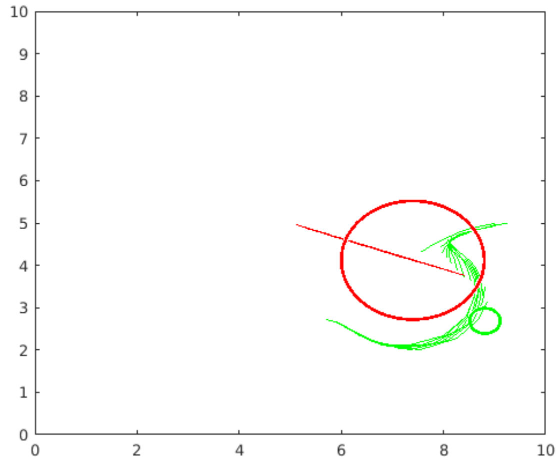

Fig. 5. Stream function with Deterministic GPR and LMPC for avoiding a single moving obstacle

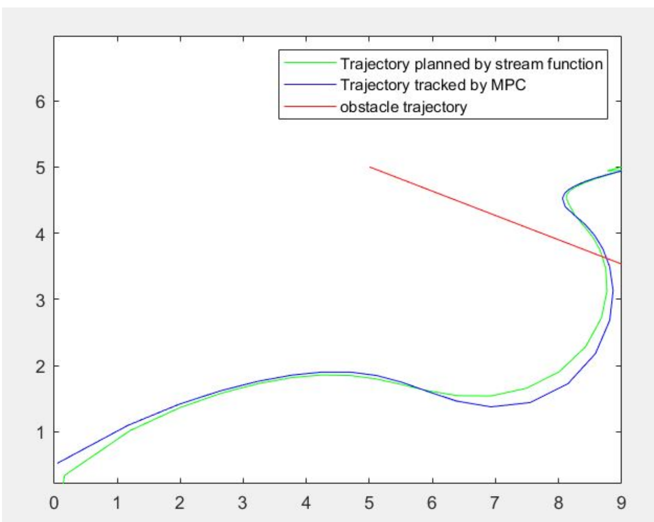

Fig. 6. Actual and planned trajectory of the robot

REFERENCES

[1] B. Zhang, "Autonomous Cars Could Save The US $1.3 Trillion Dollars A Year," Sep 2014, accessed 05-06-2019. [Online]. Available: https://www.businessinsider.com/morgan-stanley-autonomous-cars-trillion-dollars-2014-9

[2] J. Markoff, "Googles Next Phase in Driverless Cars: No Steering Wheel or Brake Pedals," May 2014, accessed 05-06-2019. [Online]. Available: https://www.nytimes.com/2014/05/28/technology/googles-next-phase-in-driverless-cars-no-brakes-or-steering-wheel.html

[3] C. Rösmann, W. Feiten, T. Wösch, F. Hoffmann, and T. Bertram, "Trajectory modification considering dynamic constraints of autonomous robots," in *ROBOTIK*, 2012.

[4] S. S. Ge and Y. J. Cui, "Dynamic motion planning for mobile robots using potential field method," *Autonomous Robots*, vol. 13, pp. 207–222, 2002.

[5] S. Akishita, S. Kawamura, and K. Hayashi, "Laplace potential for moving obstacle avoidance and approach of a mobile robot," 1990.

[6] S. Waydo and R. M. Murray, "Vehicle motion planning using stream functions," in *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, vol. 2, Sep. 2003, pp. 2484–2491 vol.2.

[7] L. M. Milne-Thomson, *Theoretical hydrodynamics*. New York: Dover Publications, 1996.

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

[9] O. Andersson, M. Wzorek, P. Rudol, and P. Doherty, "Model-predictive control with stochastic collision avoidance using bayesian policy optimization," *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4597–4604, 2016.

[10] D. Bhatt, A. Garg, B. Gopalakrishnan, and K. M. Krishna, "Chance constraints integrated mpc navigation in uncertainty amongst dynamic obstacles: An overlap of gaussians approach," 2018.

[11] Z. Wang, "Human movements prediction using on-line gaussian processes," 2016.

[12] C. E. Rasmussen and C. K. I. Williams, "Gaussian processes for machine learning," *the MIT Press*, 2006.

[13] H. Chen and F. Allgöwer, "A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability," *1997 European Control Conference (ECC)*, pp. 1421–1426, 1997.

[14] A. Bemporad and M. Morari, "Robust model predictive control: A survey," 1999.