Stochastic MPC for Quadrotor Navigation

Siddhik Reddy Kurapati
Mechanical Engineering

Jayaprakash Harshavardhan
Mechanical Engineering

Atharva Navsalkar
Aerospace Engineering

***Abstract*—This project aims to develop a controller that enables a quadcopter to navigate complex and uncertain environments. As quadcopters are getting better and cheaper as time passes so is the accessibility to them. With increasing number of applications for the drones, autonomy becomes an essential aspect for the drones that need to travel larger distances without human monitoring. Often the environments in such cases might not be known accurately apriori or have inherent uncertainties. In such cases, a stochastic MPC is proposed ensures that the drone can navigate these non-deterministic environments. We particularly focus on the problem of obstacles avoidance for randomly moving obstacles. We consider that the mean position of the obstacles is known, and their position changes randomly according a known probability distribution. This may not be the case for actual deployment as it senses the obstacles using the sensor in real-time for deployment beyond simulations, where the probability distributions of the uncertainties is not known perfectly. In this project, we utilize the MPC framework to implement stochastic constraints for cylindrical perturbing obstacles. This field is an interesting area of research that is fast evolving with new approaches to handle the uncertainties more accurately.**

***Index Terms*—Stochastic MPC, quadcopter, optimization, Probabilistic constraints**

I. Introduction

In real-world applications, quadcopters are frequently utilized UAVs (Unmanned Aerial Vehicles). Popular consumer goods for individual consumers looking for entertainment are quadcopters. The use of quadcopters is swiftly growing beyond private individuals to include a variety of industries, agricultural, shooting, etc. The quadcopters, for instance, can keep an eye on the status of the agricultural goods that are growing and carry out fertilizing, watering, and pesticide spraying. The job of the quadcopters raises the intelligence level in agriculture, enhancing its effectiveness and profitability [1] – [3]. The majority of autonomous quadcopters' tasks necessitate that they a certain course, such as spraying, mapping, monitoring, etc. Drones are used today for a wide range of purposes, including food delivery [4], public security [5], journalism [6], robotics education [7], biodiversity protection [8], precision agriculture for pest monitoring [9], and the targeted application of pesticide [10]. However, certain potential applications are still unexplored because of the costs associated with purchasing and maintaining these devices.

Quadcopters are getting cheaper as time progresses, and the accessibility for use cases is also increasing with it. The range of a drone is increasing with time hence creating a need for a method of reliable transportation when the drone is not directly in the line of sight. Most drones are today used in urban settings where there are multiple obstacles every few meters but one can argue that this problem will be much harder if a drone were to traverse in a forest under tree cover. At least in an urban setting, it would be easier to collect data, and maintaining an eye on drone navigation is much easier, the same cannot be said while traversing forest cover.

Until some time ago PID was the only method to control a quadcopter but, MPC is now evolving as a much more robust method to control the drone in ways where a PID couldn't cope. An MPC can work with multiple nonlinear constraints which is just not possible in a PID. In the case of navigating through one has to account for disturbances, To cope with these conditions a robust MPC was initially introduced and is still commonly used for my cases involving as proposed in [11]. However, the robust MPC described in [11], always considers the worst possible case scenario, to ensure that this strategy is effective to possible uncertainties. Designing a controller for the worst-case uncertainty can result in severe conservatism and even the impossibility of solving the optimization problem, since the worst-case uncertainty can be arbitrarily big and does not necessarily occur in practice. Moreover, these control algorithms often ignore the probabilistic information about disturbances to which the quadrotor is susceptible, such as wind-gust disturbances, measurement noises, and the uncertainty arising from the attitude tracking error.

The stochastic MPC is proposed as a solution to the aforementioned problems. This method designs less conservative controllers using prior probabilistic information about perturbations. For example, wind-gust disturbance can be treated as a stochastic process with Gaussian probability distribution. The wind-gust disturbance can be represented in power spectral density using Dryden or von Kármán turbulence models, where random processes are colored [12].

Similarly, we propose in this paper to design a stochastic MPC as a controller for a quadcopter that will be able to navigate through obstacles in an uncertain environment such as navigation under forest cover to avoid detection. We can think of multiple applications for such technology both in military and civilian aspects.

II. System Dynamics

While the formulation of the constarints is applicable for a wide variety of autonomous robots, we focus on quadrotors for this project. This section describes the dynamics of the model (quadrotor), that we consider for the MPC formulation. As seen in Figure 1, two reference frames are considered: (*i*) inertial frame $\mathcal{A}$ with axes $\mathbf{a}_1$, $\mathbf{a}_2$, $\mathbf{a}_3$ and (*ii*) body frame $\mathcal{B}$

with axes $\mathbf{b}_1$, $\mathbf{b}_2$, $\mathbf{b}_3$. The position $\mathbf{r}$ in $\mathcal{A}$ is represented by $\begin{bmatrix} x & y & z \end{bmatrix}^\top$. The $[Z-X-Y]$ Euler angle notation is used to represent the orientation of the drone. The orientation angles are roll ϕ (along X-axis), pitch θ (along Y-axis), and yaw ψ (along Z-axis). The angular velocity in the body-fixed frame is given by $\boldsymbol{\omega} = p\mathbf{b}_1 + q\mathbf{b}_2 + r\mathbf{b}_3$. The state of the system is represented by the position and velocity, the Euler Angles (in the $[Z-X-Y]$ sequence), and the angular velocities. The control inputs are the thrust and the moments about the three axes.

The state and control inputs are denoted by

$$\mathbf{x} = \begin{bmatrix} x & y & z & \dot{x} & \dot{y} & \dot{z} & \phi & \theta & \psi & p \\ q & r & & & & & & & & \end{bmatrix}^\top,$$

$$\mathbf{u} = \begin{bmatrix} u_1 & u_2 & u_3 & u_4 \end{bmatrix}^\top.$$

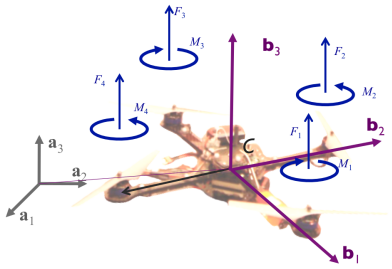

Fig. 1. The inertial and body-fixed reference frames, and forces and moments by each of the rotors. Image obtained from 'Aerial Robotics' on Coursera by Prof. Vijay Kumar.

The dynamic equations for translational motion of the drone are

$$m\ddot{\mathbf{r}} = m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + R \begin{bmatrix} 0 \\ 0 \\ u_1 \end{bmatrix}, \tag{1}$$

where the components are denoted in the inertial frame along $\mathbf{a}_1$, $\mathbf{a}_2$ and $\mathbf{a}_3$; m and g represent mass and gravitational acceleration respectively, and R is the rotation matrix from $\mathcal{B}$ to $\mathcal{A}$. The input

$$u_1 = F_1 + F_2 + F_3 + F_4,$$

is the combined thrust obtained, where F_i is the thrust produced by i^{th} propeller. The equations for the rotation are:

$$I \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} u_2 \\ u_3 \\ u_4 \end{bmatrix} - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times I \begin{bmatrix} p \\ q \\ r \end{bmatrix}, \tag{2}$$

where the components are along the body-fixed principal axes $\mathbf{b}_1$, $\mathbf{b}_2$ and $\mathbf{b}_3$; I is the inertia matrix and L is the distance between the rotor and the center of mass of the drone; the rotational control inputs are

$$u_2 = L(F_2 - F_4),$$
$$u_3 = L(F_3 - F_1),$$
$$u_4 = M_1 - M_2 + M_3 - M_4,$$

where u_2, u_3, and u_4 are the moments along the body axes and M_i is the moment produced by i^{th} propeller.

The dynamics of the multirotor can be represented in the standard $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u})$ as represented in form as follows

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \ddot{x} \\ \ddot{y} \\ \ddot{z} \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \frac{1}{m}(c\psi s\theta + c\theta s\phi s\psi)u_1 \\ \frac{1}{m}(s\psi s\theta - c\theta s\phi c\psi)u_1 \\ \frac{1}{m}(c\phi c\theta)u_1 - g \\ p(c\theta) + r(s\theta) \\ p(s\theta s\phi/c\phi) + q - r(c\theta s\phi/c\phi) \\ -p(s\theta/c\phi) + r(c\theta/c\phi) \\ \frac{1}{I_{xx}}(u_2 - (I_{zz} - I_{yy})qr) \\ \frac{1}{I_{yy}}(u_3 - (I_{xx} - I_{zz})pr) \\ \frac{1}{I_{zz}}(u_4 - (I_{yy} - I_{xx})pq) \end{bmatrix}, \tag{3}$$

where I has only the diagonal components I_{xx}, I_{yy}, I_{zz} as $\mathbf{b}_1$ along $\mathbf{b}_2$, $\mathbf{b}_3$ principal axes, and I denote $c\phi := \cos(\phi)$ and $s\phi := \sin(\phi)$ and so on, for better readability.

Specific limits on states and inputs are applied for all agent at all times as deterministic constraints $\mathbf{x} \in \mathcal{X}$ and $\mathbf{u} \in \mathcal{U}$. These include environment boundaries, quadrotor orientation bounds, actuators limit, among others. Full state dynamics (3) are linearized around the hovering condition for faster computation.

III. Collision Avoidance Constraints

For this project, we model the obstacle as cylindrical objects with radius r_{obs}. Additionally, the drone is modelled as a spherical object with radius r_{safe}, for safety requirements as a distance between the robot centre of mass and obstacle. Due to the cylindrical shape of the obstacles, the collision avoidance problem is effectively reduced to a plane with robot and obstacles as circles of r_{safe} and r_{obs}, respectively, as seen in figure 2.

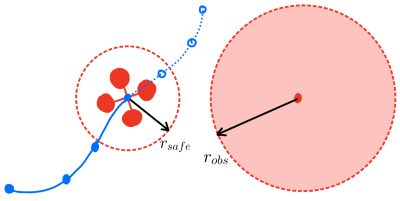

Fig. 2. Planar collision avoidance problem for drone (sphere) and obstacle (cylinder)

Let the position of i^{th} obstacle given by $\mathbf{p}_i$, the position of drone at current time t, be given by $\mathbf{p}_0^t$. At any time $t+k$, i.e., k time steps in future, the collision condition for the i^{th} obstacle is given by,

$$C_i^k := \{\mathbf{x}_k \mid ||\mathbf{x}_k - \mathbf{p}_i|| \leq r_{obs} + r_{safe}\},$$

where the free variable $\mathbf{x}_k$ is the position of the drone for the k^{th} time step [13]. The constraint for collision avoidance is effectively given by

$$\mathbf{x}_k \notin C_i^k \quad \forall i \in \mathcal{I},$$

where $\mathcal{I}$ is the set of all obstacles. This is a nonlinear constraint with an uncertain parameter $\mathbf{p}_i$. Here, we consider randomly moving obstacles where $\mathbf{p}_i \sim \mathcal{N}(\hat{\mathbf{p}}_i, \Sigma_i)$. This represents a randomly moving obstacle, with mean position $\hat{\mathbf{p}}_i$ with covariance matrix for the position being Σ_i. Figure 3 shows how the obstacle perturbs randomly with Gaussian uncertainty (drawn from a normal distribution).

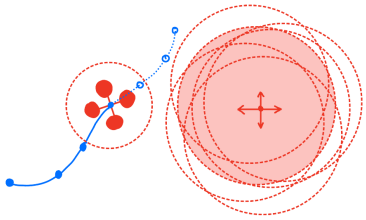

Fig. 3. Randomly moving obstacle with Gaussian uncertainty

IV. Linearized Chance Constraints

We implement the stochastic version of the collision avoidance constraint which is given by

$$\Pr(\mathbf{x}_k \notin C_i^k) \geq \delta, \quad \forall i \in \mathcal{I}, \quad \forall k,$$

where δ is the collision probability threshold. The probabilistic constraint must be converted to a deterministic form to get computationally tractable constraints. We refer to [13], where the authors solve the linear chance constraints in the deterministic form.

Consider a linear chance constraint in the form $\Pr(\mathbf{a}^T\mathbf{x} \leq b) \leq \delta$, where $\mathbf{x} \in \mathbb{R}^n$ is a random variable such that $\mathbf{x} \sim \mathcal{N}(\hat{\mathbf{x}}, \Sigma)$, $\mathbf{a} \in \mathbb{R}^n$, and $b \in \mathbb{R}$. Authors in [13] show that

$$\Pr(\mathbf{a}^T\mathbf{x} \leq b) \leq \delta \iff \mathbf{a}^T\hat{\mathbf{x}} - b \geq c, \tag{4}$$

where $c = \text{erf}^{-1}(1-2\delta)\sqrt{2\mathbf{a}^T\Sigma\mathbf{a}}$. The error function is given by $\text{erf}(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}dt$.

Our original collision avoidance constraint can be linearized with respect to the current drone position $\mathbf{p}_0^t$. This can be seen as a linear (half-space) constraint that is tangential to the (new) circular ball of $R = r_{safe} + r_{obs}$. Figure 4 shows how the nonlinear distance constraint is linearized based on current drone position and obstacle position.

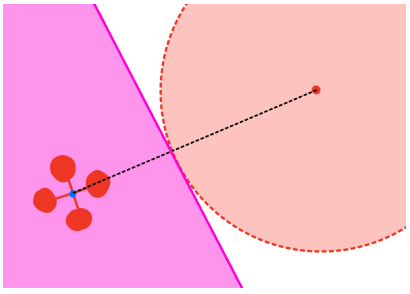

Fig. 4. Linearization of the collision avoidance constraints

Mathematically, the new collision region $\hat{C}_i^k$ is given by

$$\hat{C}_i^k := \{\mathbf{x}_k | \mathbf{a}_i^T(\mathbf{x}_k - \mathbf{p}_i) \leq b_i\}, \tag{5}$$

where $\mathbf{a}_i = (\mathbf{p}_0^t - \hat{\mathbf{p}}_i)/||(\mathbf{p}_0^t - \hat{\mathbf{p}}_i)||$ and $b_i = R = r_{safe} + r_{obs}$.

The deterministic form of these constraints, as discussed above, will then be given by,

$$\mathbf{a}_i^T(\mathbf{x}_k - \mathbf{p}_i) \geq b_i + \text{erf}^{-1}(1-2\delta)\sqrt{2\mathbf{a}_i^T\Sigma\mathbf{a}_i}. \tag{6}$$

This constraint can now be plugged into the MPC formulation, as described in the next section.

V. MPC formulation

We are formulating a collision avoidance problem. To perform this task using trajectory tracking. For simplicity, the reference trajectory is considered to be a straight line increasing only in one dimension. We slightly modify the 3rd Model Predictive Control formulation from Module 5 of our class notes which considers integral control to combat steady-state error. This method is sometimes helpful to tackle uncertainties due to model mismatch or process noise.

$$x_{k+1} = Ax_k + Bu_k \tag{7}$$

$$i_k + 1 = i_k + (Cx_k - r) \tag{8}$$

$$\begin{bmatrix} x_{k+1} \\ i_{k+1} \\ r_{k+1} \end{bmatrix} = \begin{bmatrix} A & 0 & 0 \\ C & I_{rXr} & -I_{rXr} \\ 0 & 0 & I_{rXr} \end{bmatrix} \begin{bmatrix} x_k \\ i_k \\ r_k \end{bmatrix} + \begin{bmatrix} B \\ 0 \\ 0 \end{bmatrix} u_k \tag{9}$$

We know that 12 states represent the dynamics of a quadcopter, a 3 states are for integral control for states x,y, and z. We also need additional reference states for x,y, and z. Thus the total number of states to 18 for MPC formulation and inputs stay at 4 representing different moments. Slight modification included a varying set point tracking over the prediction horizon instead of using a constant reference tracking for the prediction horizon. Varying set point is evaluated using a polynomial function function. For solving the MPC problem, CasADI and MPCtools were used.

$$e_k = C_{x_k} - r = \begin{bmatrix} C & 0 & -I_{rXr} \end{bmatrix} x_k \tag{10}$$

$$i_k = \begin{bmatrix} 0 & I_{rXr} & 0 \end{bmatrix} x_k \tag{11}$$

$$x_k^{ext} = \begin{bmatrix} x_k & i_k & r_k \end{bmatrix}^T \tag{12}$$

We calculate the cost of formulation in the following manner

$$J = \sum_{k=0}^{N-1}((x_k^{ext})^T(E^TQ_eE + L^TQ_iL)x_k^{ext} + u_K^TRu_K \rightarrow \min_{u_0,\ldots u_{N-1}} \tag{13}$$

$$\begin{aligned} st \quad & x^k = Ax^{k-1} + Bx^{k-1} \\ & x_{min}^{ext} \leq x_k^{ext} \leq x_{max}^{ext} \\ & u_{min} \leq u_k \leq u_{max} \\ & Pr(x^k \notin C_o^k) \leq \delta, \forall o \in I \\ & \forall k \in 1, \ldots N \end{aligned} \tag{14}$$

Terminal penalty P for the above formulation evaluated using dlyap and dlqr functions in Matlab. Feedback gain was evaluated using (Q,R,Ad,Bd) for dlqr, subsequently Terminal Penalty was calculated.

VI. Results and Discussions

This section performs numerical simulations to evaluate proposed Stochastic Model Predictive Control(SMPC). Simulations were done in two environments with obstacles at different position.The obstacles are considered to be cylindrical in shape and with a radius of 5 metres and are centered according the figure 5. The first set of results will show that proposed controller was able to successfully to navigate to the goal position avoiding obstacles for two different scenarios. The second set of results will compare number of collisions with increase in covariance and tolerance.

A. Collision Avoidance

For simulations, the obstacles were considered to be cylinders and two environments were created with obstacles at different positions. The Q,Qi,R matrix were easily tuned considering various trajectories like straight line and helix path. The SMPC successfully allowed the drone to reach the goal location without hitting any obstacles. Fig 5,6 shows

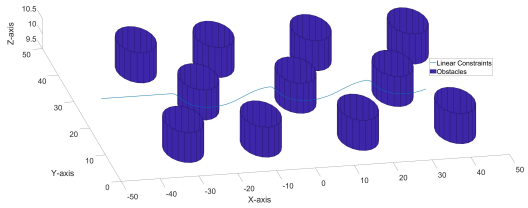

Fig. 5. Simulation Results of Quad-copter avoiding multiple obstacles in 3d view for environment 1.

a simulation of SMPC using linear constraints as shown in Section III. A covariance of 0.1 for the obstacle uncertainties and tolerance of 15% for collision was considered to solve for linearized constraints. In Fig 6, the path taken by quadrotor using linearized constraints is not same as the non-linear constraints. The non-linear constraints MPC have high computational time and don't account for uncertainties of the obstacles.

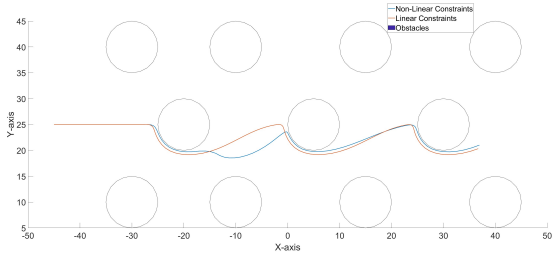

Fig. 6. 2-d trajectory of SMPC for Linearized and Non-linear constraints for environment 1.

Fig 7 and 8 represents the drone navigation in the second environment where the path taken using the linear constraints and non-linear constraints are same. Upon a closer look, the path taken by later one is the much closer to obstacles. Even with slight stochastic uncertainty(covariance 0.1), the

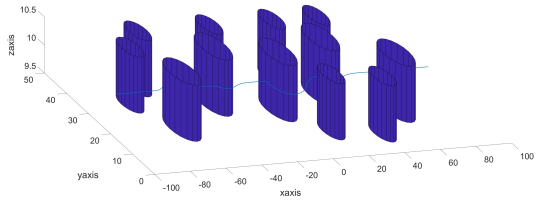

Fig. 7. Simulation Results of Quad-copter avoiding multiple obstacles in 3d view for environment 2.

drone collides with the obstacles multiple times. Whereas with proposed controller, drone successfully navigates and reaches the goal position with an uncertainty much greater than 0.1. SMPC allows us to navigate through uncertainty as it accounts for safety factor(not too low nor too high). The parameters used to define SMPC will be discussed in the next subsection.

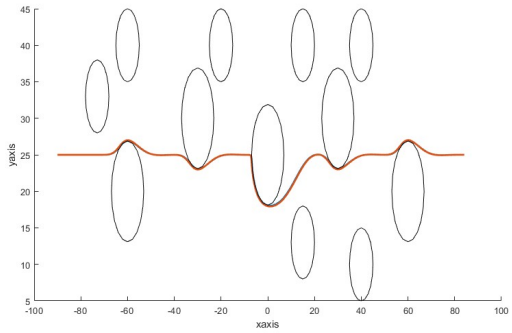

Fig. 8. 2-d trajectory of SMPC for Linearized and Non-linear constraints environment 2.

B. SMPC parameters and efficiency comparisons

In this section, we demonstrate how the proposed Stochastic Model Predictive Control (SMPC) system accounts for the uncertainties of obstacles by examining the number of collisions under varying tolerances for collision and covariances for obstacle uncertainties. For environment 1, as depicted in Figure 5, we specifically focus on obstacles located at positions [-20, 25], [5, 25], and [30, 25]. These obstacles are strategically positioned along the central trajectory path for detailed comparison, while the influence of other obstacles on the trajectory is minimal. As the covariance value increases, the constraints within the system are tightened, resulting in lesser number of collisions. Figures 9, 10, and 11 illustrate the distance between the obstacles and the drone, demonstrating that an increase in the covariance matrix enhances the effectiveness of the linearized chance constraints, as defined in Section IV, to appropriately account for obstacle uncertainty. When SMPC covariance parameter equal to 0, number of collisions are around 19 and whereas number of collisions for two covariance is just two.

From Figures 9, 10 and 11, it is observed that as the covariance matrix is reduced, the frequency with which the

distance from the obstacles to the quadcopter crosses the collision limit (which is equivalent to the radius of the obstacles) also increases. Consequently, simulations with a covariance of zero result in the highest number of collisions, as they fail to account for the stochastic nature of the obstacles. Increasing covariance parameter of SMPC also increase a chance of not tracking the reference signal. For simulation with very high value of the covariance matrix diagonals ($\sim 4m$), the drone does not reach the goal position as safety constraints become too large.

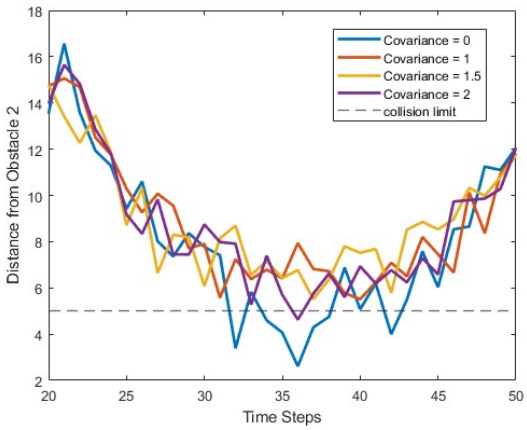

Fig. 9. Distance between the obstacle(positioned at [-20,25]) to Quadcopter current position between 20 to 50 time steps with tolerance of 0.15 and obstacle uncertainty about covariance of 1(simulated) for varying covariance parameter(SMPC).

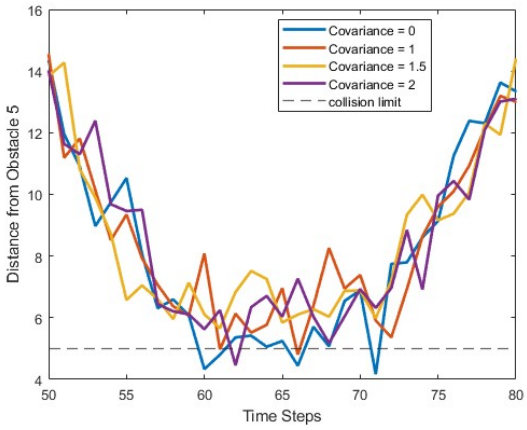

Fig. 10. Distance between the obstacle(positioned at [5,25]) to Quadcopter current position between 50 to 80 time steps with tolerance of 0.15 and obstacle uncertainty about covariance of 1(simulated) for varying covariance parameter(SMPC).

Figure 12, 13, and 14 illustrate how varying the tolerance for collisions affects the number of collisions encountered by the SMPC controller. As the tolerance decreases, the controller becomes less susceptible to collisions with obstacles. This decrease in tolerance results in tighter constraints, providing stricter collision avoidance measures. Specifically, when the

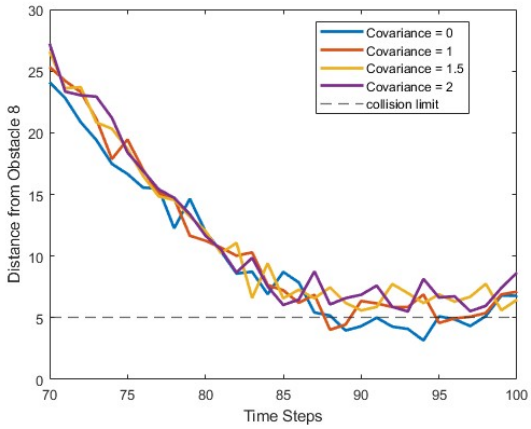

Fig. 11. Distance between the obstacle(positioned at [30,25]) to Quadcopter current position between 70 to 100 time steps with tolerance of 0.15 and obstacle uncertainty about covariance of 1(simulated) for varying covariance parameter(SMPC).

tolerance is set at 0.35, there are 14 collisions; at a tolerance of 0.25, the collisions reduce to 6; and further lowering the tolerance to 0.05 results in only 1 collision, with all cases maintaining the same covariance of 1 (diagonal elements).

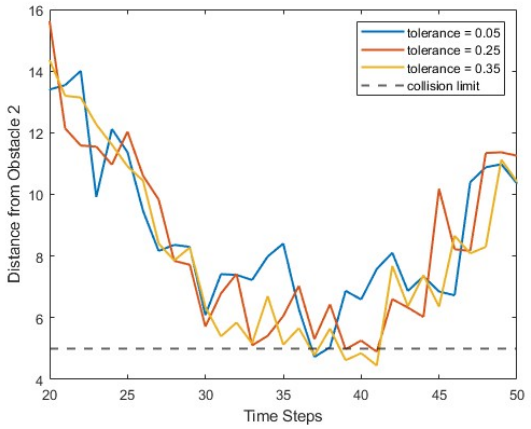

Fig. 12. Distance between the obstacle(positioned at [-20,25]) to Quadcopter current position between 20 to 50 time steps with covariance parameter(SMPC) of 1 and obstacle uncertainty about covariance of 1(obstacle-simulation) for varying tolerance.

VII. Conclusion

In this project, we have successfully proposed and implemented Stochastic MPC for dealing with Obstacles uncertainties. Proposed controller was successfully navigating through difficult and different environment with multiple obstacles. We also investigated about the various collision avoidance constraints and converting them into stochastic version of it. The efficiencies of the covariance and tolerance values were also evaluated and discussed.

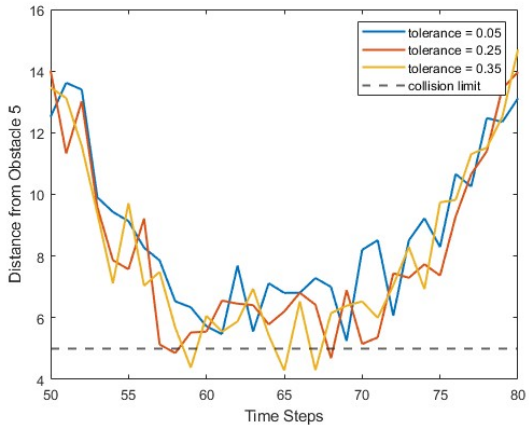

Fig. 13. Distance between the obstacle(positioned at [5,25]) to Quadcopter current position between 50 to 80 time steps with covariance parameter(SMPC) of 1 and obstacle uncertainty about covariance of 1(obstacle-simulation) for varying tolerance.

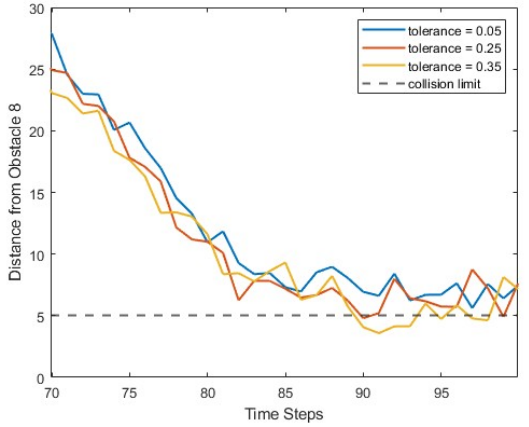

Fig. 14. Distance between the obstacle(positioned at [30,25]) to Quadcopter current position between 70 to 100 time steps with covariance parameter(SMPC) of 1 and obstacle uncertainty about covariance of 1(obstacle-simulation) for varying tolerance.

REFERENCES

[1] V. Puri, A. Nayyar, and L. Raja, "Agriculture drones: A modern breakthrough in precision agriculture," J. Statist. Manage. Syst., vol. 20, no. 4, pp. 507–518, 2017, doi: 10.1080/09720510.2017.1395171.

[2] U. R. Mogili and B. B. V. L. Deepak, "Review on application of drone sys- tems in precision agriculture," Proc. Comput. Sci., vol. 133, pp. 502–509,Jul. 2018, doi: 10.1016/j.procs.2018.07.063.

[3] J. Navia, I. Mondragon, D. Patino, and J. Colorado, "Multispectral mapping in agriculture: Terrain mosaic using an autonomous quadcopter UAV," in Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS), Jun. 2016, pp. 1351–1358.

[4] Seo SH, Won J, Bertino E, Kang Y, Choi D. A security framework for a drone delivery service. InProceedings of the 2Nd Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use 2016 Jun 26 (pp. 29-34).

[5] Alsamhi SH, Ma O, Ansari MS, Gupta SK. Collaboration of drone and internet of public safety things in smart cities: An overview of qos and network performance optimization. Drones. 2019 Jan 27;3(1):13.

[6] Holton AE, Lawson S, Love C. Unmanned Aerial Vehicles: Opportunities, barriers, and the future of "drone journalism". Journalism practice. 2015 Sep 3;9(5):634-50.

[7] Krajník T, Vonásek V, Fišer D, Faigl J. AR-drone as a platform for robotic research and education. InResearch and Education in Robotics-EUROBOT 2011: International Conference, Prague, Czech Republic, June 15-17, 2011. Proceedings 2011 (pp. 172-186). Springer Berlin Heidelberg.

[8] López JJ, Mulero-Pázmány M. Drones for conservation in protected areas: Present and future. Drones, 3 (1), 1–23.

[9] Puri V, Nayyar A, Raja L. Agriculture drones: A modern breakthrough in precision agriculture. Journal of Statistics and Management Systems. 2017 Jul 4;20(4):507-18.

[10] Mogili UR, Deepak BB. Review on application of drone systems in precision agriculture. Procedia computer science. 2018 Jan 1;133:502-9.

[11] Alexis K, Papachristos C, Siegwart R, Tzes A. Robust model predictive flight control of unmanned rotorcrafts. Journal of Intelligent Robotic Systems. 2016 Mar;81:443-69.

[12] Beal TR. Digital simulation of atmospheric turbulence for Dryden and von Karman models. Journal of Guidance, Control, and Dynamics. 1993 Jan;16(1):132-8.

[13] Zhu, Hai, and Javier Alonso-Mora. "Chance-constrained collision avoidance for mavs in dynamic environments." IEEE Robotics and Automation Letters 4.2 (2019): 776-783.