Control of a Helicopter Under Wind

Rishi Narayan Chakraborty, Sharvit Dabir, Vaibhav Shete, Advaith Sethuraman

***Abstract*—VTOL aircrafts prove to be of great utility in relatively confined environments for surveillance, inspection, package delivery, search-and-rescue, cinematography and other. This paper explores Linear Quadratic Regulator (LQR) and Model Predicitive Control (MPC) of a small scale helicopter following a non-aggressive trajectory, subjected to wind. The simulations reveal the necessity of a Kalman Observer to avoid steady state error while tracking a straight-line trajectory with wind.**

I. Introduction

Vertical Take-Off and Landing (VTOL) air-crafts have been used for search and rescue operations and package delivery in dangerous terrains where fixed winged air-crafts cannot operate. Other areas where they are deployed are cinematography, inspection and surveillance. Quadrotors, given their symmetrical design and elegant dynamics, are easier to control. However, quadrotors are widely manufactured on an RC scale unlike helicopters, which are manufactured on large scale as well as RC scale. The non-symmetrical orientation of rotors on traditional design of a helicopter, giving rise to complex and coupled dynamics, poses a difficult control problem. Exploring controller strategies that are not only able to handle the coupled dynamics of a helicopter but are robust to external disturbances such as wind (which is an inevitable reality of the operational space of a UAV) is essential.

Two of the notable works on system identification and parameterization of a small scale helicopter are [14] and [4]. In 1994, H_2 and H_∞ based controller was employed on a small-scale helicopter with the dual aim of smooth hover control under disturbance and achieving high-bandwidth of the closed loop controller [18]. A cascaded Single-Input-Single-Output (SISO) Proportional-Controller was explored in [5] for slow hover-like flight conditions. On similar lines, [15] implemented PD controller with rotor-stabilizer-fuselage coupled dynamics. An attempt at aggressive maneuvers in simulation was made [17] in 2000, inspired by the concept of intuitive control, where the input reference trajectory extracted from an expert human RC pilot was fed independently into a PID control loop. In 2003, moderate to high-speed flight maneuvers were realized on Yamaha R-50 robotic helicopter at Carnegie Mellon University, using gain scheduled H_∞ loop shaping controller [12].

In parallel, autonomous control of helicopter using Reinforcement Learning has been discussed in papers like [3] and [11], along with [16] focusing on learning inverted hovering. Another unique piece of work is [2] where, after learning the dynamics model from an expert RC pilot's data, Differential Dynamic Programming was used to design a controller for autonomous "autorotation" of a helicopter, which is a maneuver that can be used to 'parachute' the helicopter down safely in case of engine failure.

In this paper, we explore Optimal Control strategies with Kalman Filter for achieving robust control of the helicopter under wind, while tracking non-acrobatic trajectories.

Fig. 1. X-Cell 60 SE helicopter [10]

TABLE I
HELICOPTER PARAMETERS

Mass	8.2 Kg
Total Length	0.91 m
Total Height	0.10 m
Main Rotor Radius	0.78 m
Tail Rotor Radius	0.13 m

Section II discusses cockpit controls constituting the input space, equations of motions and searching hover equilibrium conditions. Section III talks about the results of implementation of LQR and MPC on hover (for validation) and simple trajectory tracking, followed by Section IV and Section V, which detail the behaviour of the control strategy under wind (steady-state and variable), with and without the Kalman Filter algorithm.

The link to the source code is in Section VII.

II. Model

For this paper, we referred [4] for the dynamics model of an X-Cell 60 E robotic helicopter instrumented at MIT (see Table I). For the purpose of convenience, we have ignored the effect of rotor-blade flapping dynamics, horizontal fin, vertical fin, and stabilizer bars. Another assumption worth noting is that the centre of pressure has been assumed to be coincident with the centre of gravity. We used Julia programming language for simulation and control ([6] [7] [8] [1]).

A. Controls

The pitch (or angle of attack) of the rotor blades decides the magnitude of the force acting on the blades. 'Collective-Pitch' control on a helicopter changes the pitch of the blades equally on all sides of the rotor-shaft (”all sides” radially). This generates a net thrust force on the rotor. Whereas, ’Cyclic-Pitch’ refers to the case where the pitch of a blade is different when the blade is on one side of the rotor-shaft, and the pitch changes as the blade rotates on its journey around the rotor-shaft to the other side. This creates a torque that tends to re-orient the rotor-shaft and hence tilt the thrust vector.

The collective and cyclic pitch can be changed using a ’Swash Plate’ mechanism (see [13]).

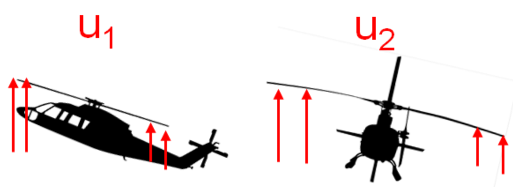

Fig. 2. Cyclic-Pitch of Main-Rotor Blades Causes Body Pitching (left) and Rolling (right)

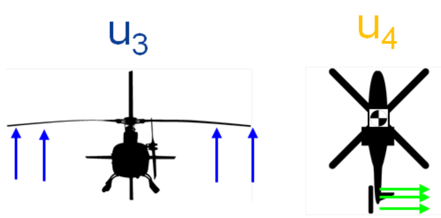

Fig. 3. Change in Collective-Pitch of Rotor Blades Causes Rise/Drop in Altitude (left) and Yawing (right)

u_1 and u_2 are representative of the cyclic pitch, where u_1 tilts the main-rotor thrust vector in the ’fore-aft’ direction thus pitching the helicopter and u_2 tilts the main-rotor thrust vector in the ’port-starboard’ direction causing the helicopter to roll. In the context of this paper, u_1 and u_2 are the projected angles that the net thrust vector makes with the body-yaw axis as viewed perpendicular to body-pitch axis and body-roll axis respectively (see Fig 2).

u_3 and u_4 represent the collective pitch of, and hence the thrust generated by, the main-rotor and tail-rotor respectively (see Fig 3).

B. Equations of Motion

The helicopter dynamics model represents a typical 6 Degrees of Freedom aerodynamic vehicle. The state is composed of position, orientation, body velocity, and body angular velocity. To avoid issues like gimbal lock while constraining space used, the orientation is represented using a quaternion [9] q. The full state is r, q, v and ω. The derivative of the state is then calculated in continuous time; since this is not a state-space representation, the dynamics are not linearized here.

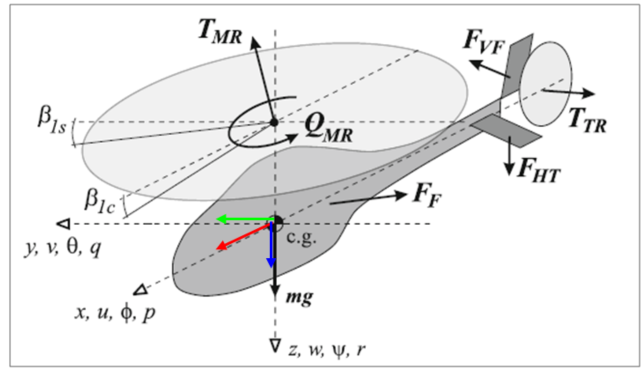

Fig. 4. Equations of Motion Derived using Newton-Euler Equations in Body Frame [4]

$$
\begin{aligned}
\dot{r} &= Qv \\
\dot{q} &= \frac{1}{2}LH\omega \\
\dot{v} &= \frac{1}{m}(Q^T \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + \begin{bmatrix} -\mathrm{T}_{mr}a_1 \\ \mathrm{T}_{mr}b_1 + T_{tr} \\ \text{-}\mathrm{T}_{tr} \end{bmatrix}) - \omega \times v \qquad (1) \\
\dot{\omega} &= J^{-1}(\begin{bmatrix} \mathrm{h}_{mr}T_{mr}b_1 + h_{tr}T_{tr} \\ \mathrm{h}_{mr}T_{mr}a_1 \\ \text{-}\mathrm{l}_{tr}T_{tr} \end{bmatrix} - \omega \times J\omega)
\end{aligned}
$$

a_1, b_1, T_{mr} and T_{tr} are analagous to u_1, u_2, u_3 and u_4. Q, L, and H come from typical quaternion mathematics. m and J represent inertia properties of the vehicle, and g is $9.81 m/s^2$. The remaining unaccounted terms, h_{mr}, h_{tr}, and l_{tr}, represent geometric properties of the vehicle measuring distances between the rotors and the vehicle center of mass.

C. Hover Conditions

To solve for the control policy, an initial reference trajectory is needed. This requires finding a state and set of inputs such that the helicopter is able to hover. Helicopter hover is different from hover in other aerodynamical systems, such as a quadcopter; here, the rotors cannot be the only controls used, as this would induce a rotation. As a bare minimum, the pitch cyclic control must also be used. In practice, stabilization with only three controls is near-impossible, and the roll cyclic control must also be used.

The hover state and control were found using the Gauss-Newton method such that the dynamics around hover would have zero velocity and zero acceleration. The system is linearized about the current guess to generate A and B matrices, the gradient and hessian of the Lagrangian are calculated, and the state and control guesses and dual variables are updated until the guess converges below a specified tolerance.

D. Approach

The control policy is designed to minimize the objective function which is based on the deviation between current and reference trajectory and magnitude of control inputs. The objective is expressed as a Quadratic Problem (QP) with linear dynamics and quadratic cost as follows:

$$\min_{u_{1:N}} \qquad x_N^T Q_N x_N + \sum_{k=1}^{N-1} x_k^T Q x_k + u_k^T R u_k \tag{2}$$

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

Where x_k and u_k are state and control at timestep k, x_N is final state and N is the number of timesteps. Q, R and Q_n are the cost matrices for state, control and final state respectively. LQR and MPC are the two controllers implemented in this paper.

The problem is solved directly on error state ($\Delta x \ \epsilon \ \mathbb{R}^{12}$) which has attitude represented by Rodrigues Parameters rather than quaternion.

$$\Delta x_k = \begin{bmatrix} r_k - \bar{r}_k \\ \varphi^{-1}(\bar{\mathbf{q}}_\mathbf{k}^{-1} \otimes \bar{\mathbf{q}}_\mathbf{k}) \\ v_k - \bar{v}_k \\ \omega_k - \bar{\omega}_k \end{bmatrix} \qquad \text{from [9]}$$

The system is assumed to be time-invariant and A and B are calculated at hover state. Then, $\tilde{A}$ and $\tilde{B}$ are modified matrices using error state jacobian ($E(x) \ \epsilon \ \mathbb{R}^{12\times 13}$) to accommodate changes caused by introduction of error state.

$$E(x) = \begin{bmatrix} I_3 & & & \\ & G(q) & & \\ & & I_3 & \\ & & & I_3 \end{bmatrix} \qquad \text{from [9]}$$

$$\tilde{A} = E(x_{hover}) \ A \ E(x_{hover})^T$$
$$\tilde{B} = E(x_{hover}) \ B$$

The first approach, LQR, is regarded as the cornerstone of controls. The Riccati equations are solved backwards from final time step to initial timestep to calculate optimal gains at each time. The Riccati equations are as follows:

$$\begin{aligned} P_N &= Q_N \\ K_k &= (R + \tilde{B}^T P_{k+1} \tilde{B})^{-1} \tilde{B}^T P_{k+1} \tilde{A} \\ P_k &= Q + \tilde{A}^T P_{k+1} (\tilde{A} - \tilde{B} K_k) \end{aligned} \tag{3}$$

Infinite Horizon LQR was implemented and the Riccati recursion was carried out until the gains K and P converged. These gains will be used for all the LQR based implementation in upcoming sections. A forward rollout in discrete time using RK4 with timestep of 0.05s is carried out on the helicopter model to find the control input needed and states it reaches for a time of 5s.

To validate the helicopter dynamics, we first disturbed the helicopter around hover state and checked whether it was able to return to equilibrium point. The infinite horizon gains and forward rollout mentioned above is used to obtain the results.

The second step was to track a straight-line trajectory from an initial position to hover position. The trajectory is generated using linear interpolation for position, SLERP for quaternion and rate of change in position for body velocity. The inifinite horizon gains and reference trajectory is used to calculate state and control at each timestep during forward rollout.

Although LQR finds a trajectory by minimizing cost, it does not account for constraints on state or control. Realistically, the helicopter is subjected to limits on thrust and cyclic pitch. To integrate constraints in the controller, Model Predictive Control approach is used; this entails generating a cost-to-go for a specified time horizon, find solution using OSQP toolbox and then update for the next time horizon. The time horizon for MPC is set at 51 timesteps in all the implementations.

The constraints are outlined in the table below. Effectively, all controls must be limited to realistic values. It may also be beneficial to account for the roll and pitch; however, testing showed this was unnecessary for the trajectory used in testing, as neither angle deviated far from stable flight to violate linearization. A more aggressive maneuver may necessitate added constraints.

TABLE II
CONSTRAINTS ON CONTROL INPUTS

Constrained parameter	Min	Max
Roll cyclic control	-0.15 rad	0.15 rad
Pitch cyclic control	-0.15 rad	0.15 rad
Main rotor thrust	0 N	200 N
Tail rotor thrust	-17 N	17 N

III. TRACKING RESULTS

A. LQR Tracking from Disturbed Position

The helicopter was able to reach hover position from random disturbed position. The error in final position of helicopter was 0.02m.

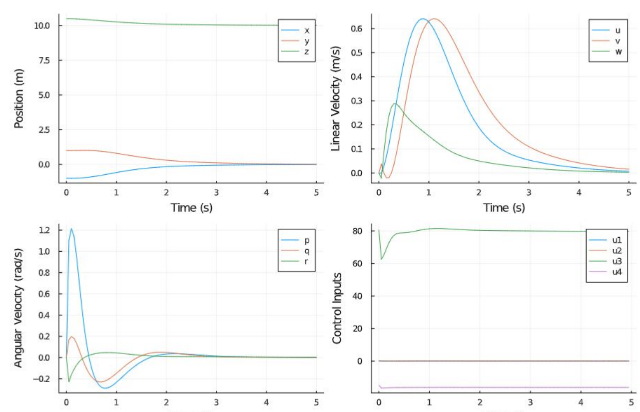

Fig. 5. LQR Performance from Disturbed Position

B. LQR Tracking for Straight Line Trajectory

LQR is able to track the reference trajectory and reaches the final position with an error of 0.27m. The control input u1 is violated while tracking the trajectory.

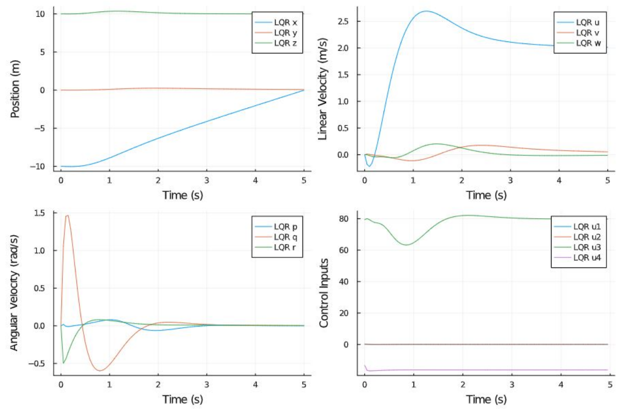

Fig. 6. LQR Performance for Straight Line Trajectory

C. MPC Tracking for Straight Line Trajectory

The MPC has constraints on control inputs as stated above. The MPC tracking is similar to LQR with less error in final position which is around 0.08m. The MPC does not violate constraint on u1 and instead stays on the limit of 0.15 rad for first few timesteps.

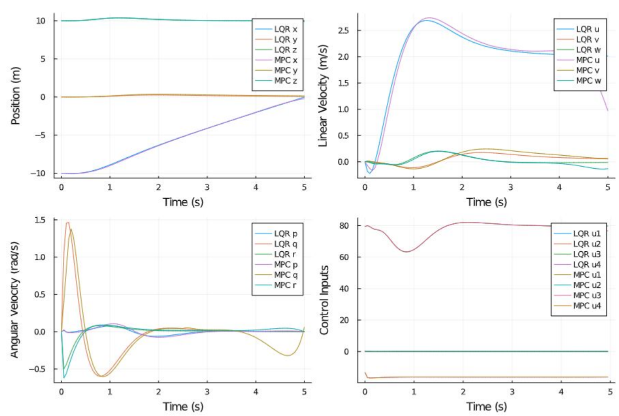

Fig. 7. MPC Performance for Straight Line Trajectory

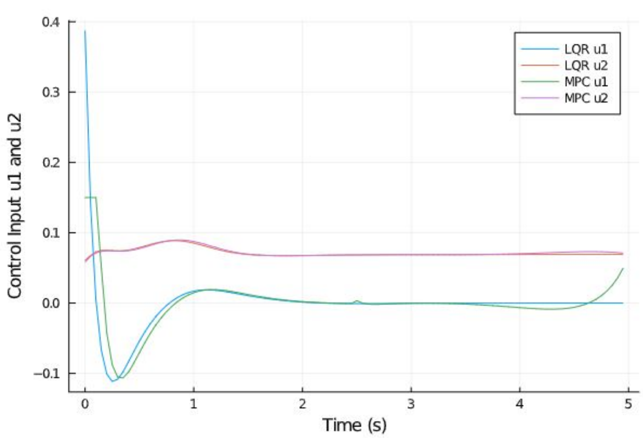

Fig. 8. Control Inputs Comparison

IV. Introduction of Steady-State Wind

A. LQR under steady-state wind

The LQR controller was given a straight line trajectory along the x axis to follow. The system was subjected to steady-state wind of 50N along the y-axis. Notable results include the error in the y trajectory. It is clear that the helicopter drifts off, and is unable to recover from the initial disturbance. The RMSE for this scenario was 3.84m.

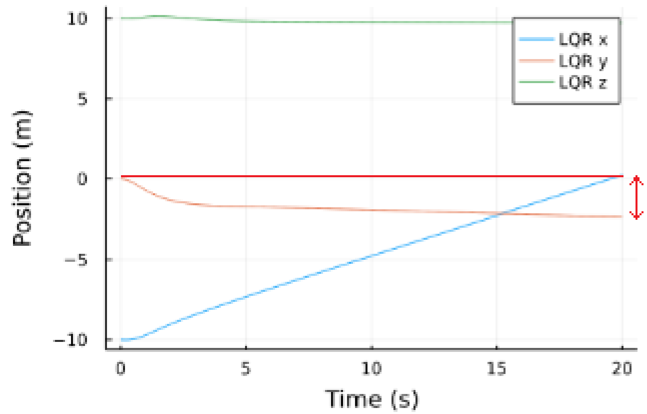

Fig. 9. LQR Position under Steady Wind

B. Kalman Observer Model

An Extended Kalman Filter is used to estimate the wind force and its rate of change. The following equations govern the prediction step:

$$x = \begin{bmatrix} s \\ f_x \\ f_y \\ f_z \\ \dot{f}_x \\ \dot{f}_y \\ \dot{f}_z \end{bmatrix} \quad s = \begin{bmatrix} r \\ q \\ v \\ \omega \end{bmatrix} \tag{4}$$

$$\hat{x}_{k+1}^{-} = A_k x_k + B_k u_k \tag{5}$$

$$P_{k|k-1}^{-} = A_k P_{k|k-1} A_k^T + Q \tag{6}$$

The following equations govern the update step:

$$H_k = \begin{bmatrix} I_{6x6} & 0_{6x6} \end{bmatrix} \tag{7}$$

$$z_k = s_k \tag{8}$$

$$\tilde{y_k} = z_k - H_k \hat{x}_{k+1}^{-} \tag{9}$$

$$K_k = P_{k|k-1}^{-} H_k^T (R + H_k P_{k|k-1}^{-} H_k^T)^{-1} \tag{10}$$

$$\hat{x}_{k+1}^{+} = \hat{x}_{k+1}^{-} + K_k \tilde{y_k} \tag{11}$$

$$P_{k|k-1}^{+} = P_{k|k-1}^{-} - K_k H_k P_{k|k-1}^{-} \tag{12}$$

Where s is the state of the helicopter according to the dynamics. f_x, f_y & f_z are wind-force components. $P^-_{k|k-1}$ and $P^+_{k|k-1}$ are the prior and posterior Covariance Matrices. K_k is the Kalman Gain. The Kalman Filter uses Jacobians A_k, B_k linearized about its current estimate. The measurement z_k is the part of the helicopter state that does not include any estimated wind. The filter uses the discrepancy between its dynamics and its actual state to estimate the amount of wind that the helicopter is subjected to.

The values for Q, P_0, R were tuned by hand until convergence.

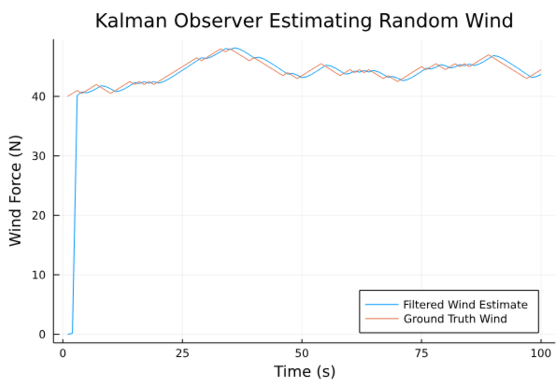

Fig. 10. Wind Estimation using Helicopter State

C. MPC with Kalman Observer under Steady-State Wind

After incorporating the Kalman Filter into the MPC controller, we reran the steady state wind test using 50N along the y-axis. The MPC controller is now able use the information from the observer model to inform the plan for the horizon. The MPC controller is also able to handle the wind much better compared to the LQR controller (also pictured in Fig. 11). The RMSE for the MPC controller under steady state wind is 0.67m.

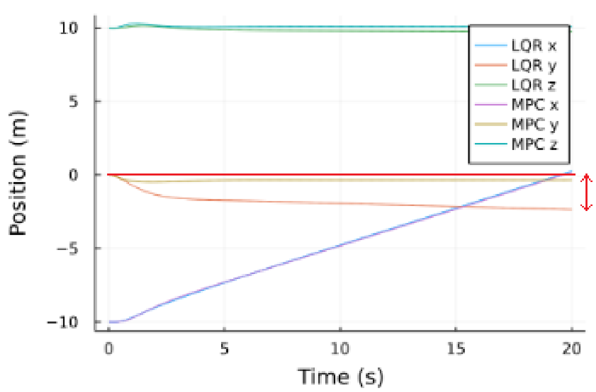

Fig. 11. LQR and MPC Position under Steady Wind and Kalman Observer

V. Introduction of Variable Wind

Although the steady-state wind model provides valuable insight into the relative performance of the LQR and MPC controllers, it does not represent a realistic scenario. A more accurate wind model was desired; to be successful, it should be continuous with random changes.

A. Wind Model

The selected wind model is based on a random walk algorithm. At each time step, a forward step or a negative step is added to the previous value for wind in both the x and y directions. Wind in the z direction has been ignored for this analysis. Although it would be possible to also vary the amount of change, this was deemed unnecessary as the small time step and large number of steps allowed the wind to change by variable amounts over larger windows.

A plot of the random wind in the x and y directions along with the estimated wind from the Kalman Observer is shown below. The true wind is C^0 and changes randomly, as desired. The Kalman Observer is able to make an accurate estimate of both the wind value and its general trajectory.

B. Modifications to the Kalman Observer

The only modification that was done to accommodate varying wind was to add an additional velocity for the wind so that the change in the wind could be better predicted.

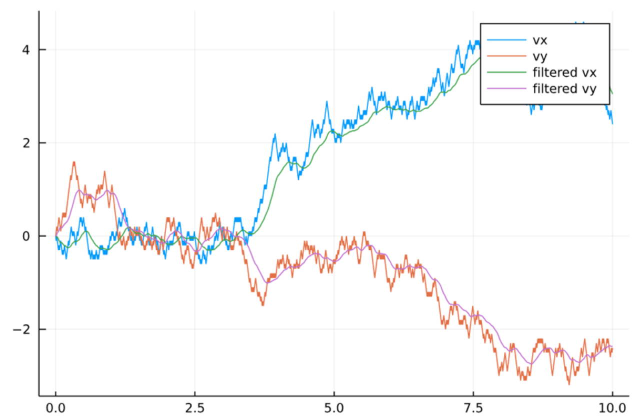

Fig. 12. True random wind and Kalman Observer-based estimation

C. MPC with Kalman Observer under variable wind

We reran the tests using the MPC controller and the LQR controller under variable wind. The conditions for this test was random wind from $Unif(40, 0.83)$. The variance was chosen from experimentation, which led to a reasonable amount of wind that was challenging for the helicopter.

The LQR and MPC controllers without observer struggle to follow the trajectory in these circumstances. The MPC controller with observer is effective at following the trajectory, under varying wind. The LQR and MPC controllers without observer achieve an RMSE of 6.85m and 7.48m respectively, whereas the MPC controller with observer achieves an RMSE of 1.57m.

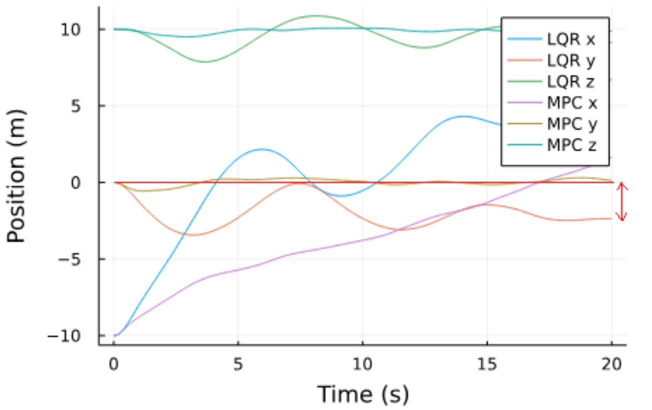

Fig. 13. MPC and LQR Position under Variable Wind

TABLE III
CONTROLLER ERROR UNDER WIND

Controller	Constant Wind (50N)	Varying Wind (Unif(40, 0.83))
LQR no Observer	3.84m	6.85m
MPC no Observer	4.44m	7.48m
MPC with Observer	**0.67m**	**1.57m**

VI. CONCLUSION

LQR and MPC are able to track the straight line trajectory without wind, but LQR is not able to constrain the control inputs. For steady-state wind, both LQR and MPC drift away in Y direction. Although the LQR error appears lower than the MPC error, this is compromised by the unrealistic control inputs in the LQR case. By introducing a Kalman Observer, the MPC controller is able to reject both a constant wind vector and a randomly-generated time-varying wind vector. The combination of MPC with Kalman Observer offers a robust controller for helicopters to perform stable flight under windy conditions, for trajectories within the linearization assumption.

VII. CODE

The Source Code is available at: https://github.com/SharvitDabir/HelicopterControl21/tree/main

REFERENCES

[1] URL: https://github.com/RoboticExplorationLab/TrajOptPlots.jl.

[2] P. Abbeel et al. "Autonomous Autorotation of an RC Helicopter". In: *Khatib O., Kumar V., Pappas G.J. (eds) Experimental Robotics. Springer Tracts in Advanced Robotics, vol 54. Springer, Berlin, Heidelberg*. 2009. DOI: https://doi.org/10.1007/978-3-642-00196-3_45.

[3] J.A. Bagnell and J.G. Schneider. "Autonomous helicopter control using reinforcement learning policy search methods". In: *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*. Vol. 2. 2001, 1615–1620 vol.2. DOI: 10.1109/ROBOT.2001.932842.

[4] Vladislav Gavrilets. "Dynamic Model for a Miniature Aerobatic Helicopter". In: *Handbook of Unmanned Aerial Vehicles*. Ed. by Kimon P. Valavanis and George J. Vachtsevanos. Dordrecht: Springer Netherlands, 2015, pp. 279–306. ISBN: 978-90-481-9707-1. DOI: 10.1007/978-90-481-9707-1_54. URL: https://doi.org/10.1007/978-90-481-9707-1_54.

[5] D. Hyunchul Shim, Hyoun Jin Kim, and S. Sastry. "Control system design for rotorcraft-based unmanned aerial vehicles using time-domain system identification". In: *Proceedings of the 2000. IEEE International Conference on Control Applications. Conference Proceedings (Cat. No.00CH37162)*. 2000, pp. 808–813. DOI: 10.1109/CCA.2000.897539.

[6] Brian E. Jackson. URL: https://github.com/RoboticExplorationLab/TrajOptPlots.jl.

[7] Brian E. Jackson. URL: https://github.com/RoboticExplorationLab/RobotZoo.jl.

[8] Brian E. Jackson. URL: https://github.com/RoboticExplorationLab/RobotDynamics.jl.

[9] Brian E. Jackson, Kevin Tracy, and Zachary Manchester. "Planning With Attitude". In: *IEEE Robotics and Automation Letters* (2021). DOI: 10.1109/LRA.2021.3052431.

[10] Paulin Kantue. *Online Parameter Estimation of a Miniature Unmanned Helicopter Using Neural Network Techniques*. 2011. URL: https://www.researchgate.net/publication/313879911_Online_Parameter_Estimation_of_a_Miniature_Unmanned_Helicopter_Using_Neural_Network_Techniques/figures?lo=1.

[11] H. Kim et al. "Autonomous Helicopter Flight via Reinforcement Learning". In: *Advances in Neural Information Processing Systems*. Ed. by S. Thrun, L. Saul, and B. Schölkopf. Vol. 16. MIT Press, 2004. URL: https://proceedings.neurips.cc/paper/2003/file/b427426b8acd2c2e53827970f2c2f526-Paper.pdf.

[12] M. La Civita et al. "Design and flight testing of a gain-scheduled H/sub /spl infin// loop shaping controller for wide-envelope flight of a robotic helicopter". In: *Proceedings of the 2003 American Control Conference, 2003.* Vol. 5. 2003, 4195–4200 vol.5. DOI: 10.1109/ACC.2003.1240494.

[13] J. Liu et al. "A Novel Comprehensive Kinematic and Inverse Dynamic Model for the Flybar-Less Swashplate Mechanism: Application on a Small-Scale Unmanned Helicopter". In: *Symmetry 2020, 12, 1849* (2020), pp. 1–4. DOI: https://doi.org/10.3390/sym12111849.

[14] Bernard Mettler, Mark B. Tischler, and Takeo Kanade. "System identification modeling of a small-scale unmanned rotorcraft for flight control design". English (US). In: *Journal of the American Helicopter Society* 47.1 (Jan. 2002), pp. 50–63. ISSN: 0002-8711.

[15] Bernard Mettler et al. "Attitude control optimization for a small-scale unmanned helicopter". In: *AIAA Guidance, Navigation and Control Conference*. 2000.

[16] A.Y. Ng et al. "Autonomous Inverted Helicopter Flight via Reinforcement Learning". In: *Ang M.H., Khatib O. (eds) Experimental Robotics IX. Springer Tracts in Advanced Robotics, vol 21. Springer, Berlin, Heidelberg*. 2006. DOI: https://doi.org/10.1007/11552246_35.

[17] M. Piedmonte and E. Feron. "Aggressive Maneuvering of Autonomous Aerial Vehicles: A Human-Centered Approach". In: *Hollerbach J.M., Koditschek D.E. (eds) Robotics Research. Springer, London*. 2000. DOI: https://doi.org/10.1007/978-1-4471-0765-1_50.

[18] M.F. Weilenmann, U. Christen, and H.P. Geering. "Robust helicopter position control at hover". In: *Proceedings of 1994 American Control Conference - ACC '94*. Vol. 3. 1994, 2491–2495 vol.3. DOI: 10.1109/ACC.1994.735007.