

MPC for 2D Quadcopter Landing on a Moving Platform

Samuel Mankoff, Roshan Joshua Thomas, Daniel Morales

Department of Mechanical Engineering, University of California, Berkeley

ME C231A: Experiential Advanced Control I, Fall 2025

Abstract—This paper addresses the challenge of autonomous drone delivery by developing a control framework for landing a 2D quadcopter on a moving truck platform. Two control strategies were implemented and compared: a Linear Model Predictive Controller (MPC) linearized around hover and a Nonlinear Model Predictive Controller (NMPC) modeling full system dynamics. The problem was formulated as a Constrained Finite Time Optimal Control (CFTOC) problem, enforcing strict actuator limits and safety constraints. Simulation results demonstrate that while Linear MPC provides adequate tracking in nominal conditions, it suffers from vertical overshoot and constraint saturation under wind disturbances. Conversely, the NMPC formulation, implemented using Pyomo and IPOPT, successfully uses aggressive maneuvering to relax pitch constraints from 25° to 45° to achieve disturbance rejection and simultaneous convergence in both longitudinal and vertical axes. We conclude that despite higher computational costs, the nonlinear approach is essential for the safety and reliability required in real-world logistic applications.

Presentation: <https://youtu.be/B-km3JaBA3M>

I. INTRODUCTION

The purpose of this project was to successfully conceptualize and implement a Model Predictive Control (MPC) real-world application utilizing methods learned in Dr. Borelli's MEC231a course at the University of California Berkeley.

Our team decided to develop an effective MPC control algorithm that would successfully land a 2-dimensional quadcopter on a moving truck platform on the road. The environment and problem is meant to simulate the modern application of delivery quadcopters that would deliver packages to customers of their parent company and return to a docking station for charging and to receive a new package for their next mission. Our goal was to develop a creative solution to a growing problem; these delivery quadcopters currently cost a staggering amount, with their current revenue projections not outpacing their expense (Kim, Long 2022).

Hence, in this project, we surmised that developing a successful quadcopter docking methodology on a moving truck traveling on a road would eliminate the potential heavy costs with needing to rent/purchase real estate for docking station. Hence, solving our MPC problem would help validate this novel methodology, and is something we hope could point to future work that could help scale the technology of autonomous quadcopter delivery systems.

II. METHODOLOGY

A. System Dynamics

The quadcopter motion is modeled in the 2D longitudinal plane. We define the state vector $x(t) \in \mathbb{R}^6$ and input vector $u(t) \in \mathbb{R}^2$ as:

$$x(t) = [x \ z \ \theta \ \dot{x} \ \dot{z} \ \dot{\theta}]^T, \quad u(t) = [u_f \ u_b]^T \quad (1)$$

where (x, z) is the inertial position, θ is the pitch angle, and u_f, u_b are the front and rear motor thrusts.

The continuous-time nonlinear dynamics $\dot{x}(t) = f(x(t), u(t))$, including external wind disturbances $F_{wind} = [F_x, F_z]^T$, are given by:

$$\begin{bmatrix} \dot{x} \\ \dot{z} \\ \dot{\theta} \\ \ddot{x} \\ \ddot{z} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{z} \\ \dot{\theta} \\ -\frac{1}{m}(u_f + u_b) \sin \theta + \frac{F_x}{m} \\ \frac{1}{m}(u_f + u_b) \cos \theta - g + \frac{F_z}{m} \\ \frac{l}{2I_{yy}}(u_f - u_b) \end{bmatrix} \quad (2)$$

For the Linear MPC formulation, this system is linearized around the hover equilibrium ($x_{eq} = \mathbf{0}, u_{eq} = mg/2$) and discretized with sampling time $T_s = 0.05s$ to form the prediction model $x_{k+1} = A_d x_k + B_d u_k$.

B. Linearization and Discretization

The nonlinear dynamics were linearized around the hover equilibrium ($x_{eq} = \mathbf{0}, u_{eq} = mg/2$) using Jacobian linearization. The resulting Continuous LTI system was discretized with a sampling time of $T_s = 0.05s$ using the Zero-Order Hold method to obtain the prediction model matrices A_d and B_d (see Appendix for derivation).

The discrete state-space matrices result in the following equation which yields the final prediction model used in the Linear MPC formulation:

$$x_{k+1} = A_d x_k + B_d u_k \quad (3)$$

C. Control Hierarchy

The control architecture for the autonomous landing task is structured to process the dynamic nature of the moving platform and wind disturbances while stabilizing the 2D quadcopter. The system operates in a closed-loop where the controller (MPC or NMPC) receives state estimates and computes optimal thrust commands.

1) *Strategy and Reference Generation*: The first part of the control architecture uses the estimates of the moving platform and the quadcopter to determine the immediate objective. Implementing our controller with industry task/methods in mind we built out the reference trajectory over the horizon through FSM.

- **Finite State Machine (FSM)**: The FSM consists of four states: CLIMB (ascend to safety altitude), APPROACH (track horizontal position), SYNC (match velocity), and LAND (descend). Transitions are triggered by spatial thresholds (e.g., LAND when position error < 0.1m)
- **Reference Output**: The layer outputs a time-varying reference trajectory $\mathbf{x}_{ref}(k)$ over the prediction horizon N . This trajectory predicts the platform's future kinematics based on its current velocity and adjusts the target altitude to account for the platform's variable grade (slope angle ϕ).

2) *Model Predictive Control*: This is our primary feedback controller. It minimizes a cost function J penalizing state error (\mathbf{Q}), control effort (\mathbf{R}), and constraint violations (\mathbf{Q}_s). Two solver methods were developed:

Linear MPC (CVXPY)

- **Model**: Utilizes a discrete-time LTI model ($\mathbf{A}_d, \mathbf{B}_d$) linearized around the hover equilibrium ($\theta \approx 0$).
- **Optimization**: Formulated as a convex Quadratic Program (QP) and solved using OSQP. To ensure stability, an Infinite Horizon LQR cost matrix (\mathbf{P}_{LQR}) is applied as the terminal cost.
- **Constraints**: Includes hard bounds on motor thrust and soft constraints on the pitch angle θ (using slack variables s_θ) to prevent infeasibility during strong wind disturbances.

Nonlinear MPC (Pyomo)

- **Model**: Incorporates the full nonlinear dynamics, specifically retaining trigonometric terms ($\sin \theta, \cos \theta$) essential for aggressive maneuvering.
- **Optimization**: Formulated as a Non-Linear Programming (NLP) problem and solved using IPOPT. The system dynamics are enforced via equality constraints using Explicit Euler integration at each step of the horizon.

3) *Plant and Environment*: This represents the "true" physical evolution of the system.

- **Dynamics**: The system integrates the continuous-time nonlinear equations of motion given the optimal control inputs \mathbf{u}^* .
- **Disturbances**: A wind model injects external forces (\mathbf{F}_{wind}) which are calculated as a steady mean component plus Gaussian turbulence, directly into the acceleration equations with parameters specified in Appendix A.

D. Optimal Control Formulation (MPC)

The control problem is modeled as a Constrained Finite Time Optimal Control (CFTOC) problem. In the linear MPC implementation the CFTOC is formulated as a convex quadratic program (QP). At each time step t the controller

solves for the optimal inputs to minimize tracking error without violating the quadcopter dynamics.

The MPC solves the following finite-horizon optimization problem at each time step t :

$$J_{0 \rightarrow N}^*(x(0)) = \min_{X, U, S_\theta} \underbrace{p(x_N)}_{\text{terminal cost}} + \sum_{k=0}^{N-1} \underbrace{q(x_k, u_k)}_{\text{stage cost}} \quad (4)$$

In this implementation, the specific cost terms are quadratic forms derived from the project requirements:

$$\begin{aligned} q(x_k, u_k) &= \|x_k - x_{ref,k}\|_Q^2 + \|u_k - u_{ref,k}\|_R^2 + Q_s s_{\theta,k}^2 \\ p(x_N) &= \|x_N - x_{ref,N}\|_P^2 + Q_s s_{\theta,N}^2 \end{aligned}$$

$$\text{s.t. } x_{k+1} = A_d x_k + B_d u_k \quad k = 0, \dots, N-1 \quad (5)$$

$$u_{min} \leq u_k \leq u_{max} \quad k = 0, \dots, N-1 \quad (6)$$

$$|\theta_k| \leq \theta_{lim} + s_{\theta,k} \quad k = 0, \dots, N \quad (7)$$

$$x_0 = x(0) \quad (8)$$

The weighting matrices Q and R were tuned to prioritize position tracking accuracy over control effort (see Appendix A for exact values).

The solution yields the optimal control sequence $U_{0 \rightarrow N}^* = [u_0^*, u_1^*, \dots, u_{N-1}^*]$. In the standard MPC receding horizon strategy, we apply only the first element to the system:

$$u(t) = u_0^* \quad (9)$$

The process is then repeated at $t + 1$ with the updated state measurement.

E. Terminal Cost Design (LQR)

To guarantee closed-loop stability despite the finite horizon ($N = 50$), we add a terminal cost $p(x_N) = x_N^T P x_N$ to the objective. The matrix P estimates the infinite-horizon "cost-to-go" beyond step N . In this implementation, P was computed offline by solving the Discrete Algebraic Riccati Equation (DARE) using the linearized dynamics (A_d, B_d) and stage weights (Q, R). This ensures the drone ends the horizon in a state that is recoverable by a standard linear controller.

F. Nonlinear MPC

To evaluate the performance limitations imposed by linearization we also implemented a Nonlinear Model Predictive Controller (NMPC). Unlike the standard MPC, which relies on a convex QP formulation, the NMPC optimizes the control inputs directly subject to the full nonlinear equations of motion.

1) *Implementation Framework*: The problem is formulated using **Pyomo** and solved using **IPOPT**.

2) *Nonlinear Constraints*: The dynamics are enforced with equality constraints at each time step k using Explicit Euler integration. The formulation retains the nonlinearities that define the coupling between attitude and acceleration:

$$v_{x,k+1} = v_{x,k} + T_s \left(-\frac{u_{1,k} + u_{2,k}}{m} \sin(\theta_k) \right) \quad (10)$$

$$v_{z,k+1} = v_{z,k} + T_s \left(\frac{u_{1,k} + u_{2,k}}{m} \cos(\theta_k) - g \right) \quad (11)$$

This allows the optimizer to accurately predict system behavior even when the small-angle approximation ($\sin \theta \approx \theta$) breaks down.

3) *Constraint Relaxation and Wind Rejection*: Utilizing the NMPC we are able to relax the safety constraints on the pitch angle because we are modeling the true nonlinear system dynamics at each time step not a linear approximation about hover :

$$|\theta|_{max,NMPC} = 45^\circ \quad \text{vs.} \quad |\theta|_{max,Linear} = 25^\circ \quad (12)$$

The NMPC allows for more aggressive maneuvers, providing improved disturbance rejection against strong wind gusts.

III. SIMULATION/RESULTS

A. Simulation Setup

All simulations were conducted in closed loop using the full nonlinear quadcopter dynamics described in Section II, while the MPC used a linearized prediction model about the hover equilibrium. The controller operated with a sampling time of $T_s = 0.05$ s and a prediction horizon of $N = 20\text{--}50$ steps depending on the scenario. Unless otherwise stated, results shown use $N = 50$.

Hard constraints were enforced on the front and rear motor thrusts, and pitch angle constraints were imposed to ensure safe operation. To preserve feasibility during aggressive maneuvers, the pitch constraint was softened through the use of slack variables with a large quadratic penalty. For simulations involving a moving platform, the reference trajectory was generated online by predicting the platform's future position and velocity over the MPC horizon under a constant-velocity assumption. All controllers were evaluated under identical initial conditions and plant dynamics.

To evaluate robustness under realistic operating conditions, all simulations include additive wind disturbances acting on the quadcopter translational dynamics. The disturbance profile consists of bounded, time-varying horizontal and vertical wind components, representing gusts and steady bias commonly encountered during outdoor flight.

B. Linear MPC Tracking of a Moving Platform

Our first simulation evaluates Linear MPC performance for tracking a moving platform traveling at constant horizontal velocity and fixed height. The MPC reference trajectory encodes both the predicted platform position and velocity, requiring the quadcopter to synchronize its motion prior to landing.

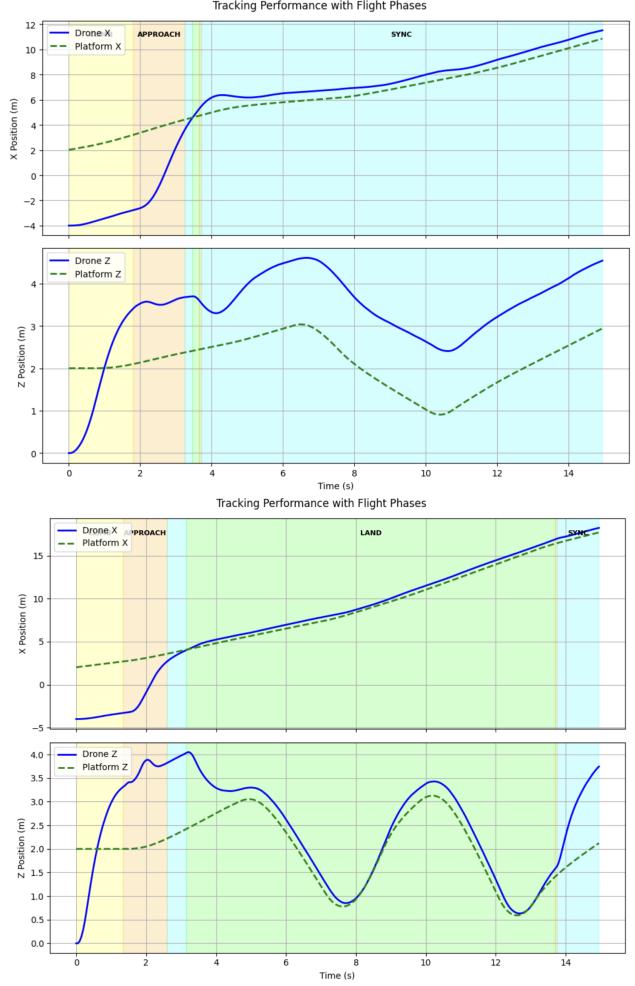
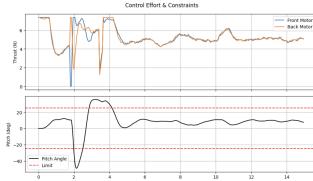


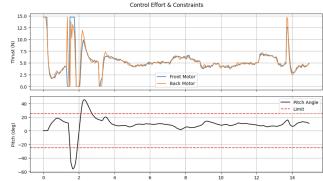
Fig. 1: Comparison of tracking performance. (a) Linear MPC shows overshoot during the landing phase. (b) NMPC maintains tighter tracking of the moving platform.

As shown in Fig. 1(a), the quadcopter accelerates to match the platform's horizontal velocity while descending toward the platform height. Although accurate tracking is achieved during the approach phase, mild overshoot is observed during the landing attempt, particularly in the vertical direction. This behavior is consistent with the limited pitch authority imposed by the linearized prediction model and conservative pitch constraints. Throughout the maneuver, thrust inputs remain within their allowable bounds, and pitch deviations remain bounded.

In the presence of wind disturbances, the Linear MPC controller does not consistently achieve a stable landing on the moving platform. Residual vertical oscillations and steady-state error persist near the touchdown phase, indicating sensitivity to external disturbances and unmodeled nonlinear effects inherent to the linearized prediction model.



(a) Linear MPC



(b) NMPC

Fig. 2: Control inputs. Top: Linear MPC hits saturation limits abruptly. Bottom: NMPC smooths the control effort while adhering to constraints.

C. Constraint Handling and Robustness

To assess robustness, the Linear MPC controller was tested under tight thrust and pitch constraints and large initial position errors. In these scenarios, the controller must balance rapid error correction against strict actuator and attitude limits.

As illustrated in Fig. 2 (a), the Linear MPC solution frequently approaches thrust saturation during aggressive phases of the maneuver. Feasibility is preserved through selective activation of the pitch slack variables, allowing temporary violations of the nominal pitch bound while avoiding actuator saturation. Once the transient subsides, the pitch angle smoothly returns within allowable limits, demonstrating effective constraint management. However, under wind disturbances, this conservative constraint handling further limits the controller’s ability to reject disturbances during the final landing phase.

D. Comparison with Nonlinear MPC

To evaluate the performance limitations imposed by linearization, a Nonlinear Model Predictive Controller (NMPC) was implemented using the full nonlinear quadcopter dynamics. Unlike Linear MPC, NMPC does not rely on small-angle approximations and permits larger pitch excursions during aggressive maneuvers.

A key qualitative difference between the two controllers is observed during the landing phase in the presence of wind disturbances. While Linear MPC can track the moving platform and reduce relative position error, it fails to consistently achieve a stable touchdown under external disturbances and modeling uncertainty. In contrast, NMPC successfully achieves landing and maintains contact with the platform.

Under identical simulation conditions, NMPC achieves improved tracking performance. As shown in Fig. 1(b), NMPC maintains tighter synchronization of horizontal and vertical motion with reduced overshoot. This improvement is further reflected in the control inputs (Fig. 2b), where NMPC avoids abrupt saturation and produces smoother thrust profiles. The improved disturbance rejection and landing performance highlight the benefit of explicitly modeling nonlinear dynamics when operating near actuator and attitude limits. Although NMPC incurs a higher computational cost, all simulations converged within the sampling period, indicating feasibility for real-time implementation.

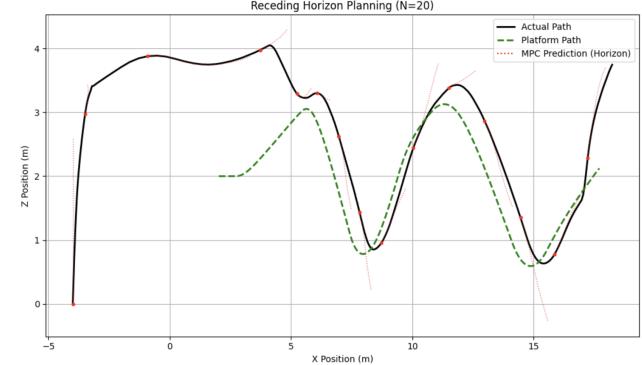
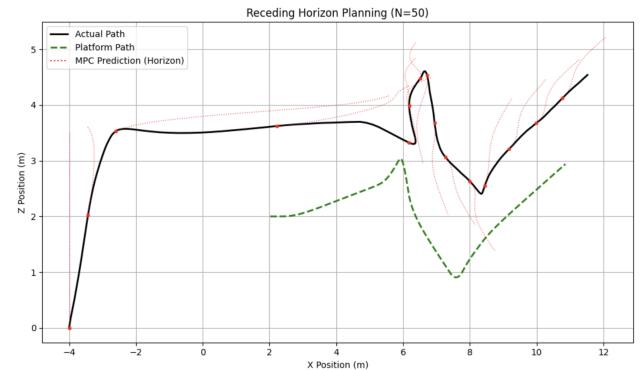


Fig. 3: Receding Horizon Control (RHC) trajectory planning. The blue line represents the predicted horizon, updating at every time step as the quadcopter approaches the moving platform.

E. Receding Horizon Prediction

The receding-horizon nature of the controller is illustrated in Fig. 3, which shows the predicted state trajectory updating at each time step as the quadcopter approaches the moving platform. The continuously updated prediction enables anticipation of future platform motion and contributes to the successful tracking behavior observed in both controllers.

Nevertheless, only the NMPC formulation consistently achieves a robust and successful landing under wind disturbances.

IV. CONCLUSION

All in all, we trialed two different methods of MPC primarily focused on different representations of the dynamics to try and ensure convergence of our algorithm to a solution to the quadcopter problem. In this case, the nonlinear MPC came out as the distinct winner over the linear MPC model. The nonlinear controller delivered a simulated quadcopter that lands on the moving truck platform. On the other hand, the linear controller fails to land on the platform because its z position does not converge the z-position of the platform even though it manages to match the x-position of the platform.

The reason for this is clear, our constraints were too strict for a linear MPC to truly converge too. In approximating quadcopter dynamics using the linear model, the error between linear dynamics and real world dynamics proved just too large

to solve our problem. It required more realistic, in other words, nonlinear dynamics to allow the quadcopter to meet the tight tolerances we set for our solution.

This reveals two key ingredients for developing MPC algorithms for quadcopter delivery systems. For one, accuracy is paramount in achieving our goal; in this case, simply being close to our goal simply does not matter; any deviation from our goal results in a completely failed mission. Secondly, the dynamics of the environment are complex and involve too many disturbances (like wind) to model linearly. Before model development, research needs to be done to develop effective non-linear models to simulate the environment.

That being said, owing to the added specificity required by our controller, we automatically guarantee an increase in computational expense and time. However, in real world applications, this is completely justified, as quadcopter delivery systems simply cannot function without these ingredients

Hence, our project holds strong success. Not only did we find a solution to the isolated problem of an MPC that controls the quadcopter to land on the moving platform, but we did so while trying to truly simulate real-world environments. Our success hence lies in the fact that we have developed a controller that does not simply work in theory, but can be possibly be applied, with refinement, to real-life quadcopter delivery.

REFERENCES

- [1] R. Firooz and F. Borrelli, “Nonlinear Constrained Finite Time Optimal Control (NCFTOC) with Pyomo,” ME C231A / EE C220B Laboratory Notes, University of California, Berkeley, 2023.
- [2] F. Borrelli and M. Morari, “MPC” UC Berkeley ME C231A Course Materials, Fall 2025.
- [3] F. Borrelli, A. Bemporad, and M. Morari, *Predictive Control for Linear and Hybrid Systems*, 1st ed. Cambridge University Press, 2017.
- [4] A. Borowczyk, D.-T. Nguyen, A. P.-V. Nguyen, D. Q. Nguyen, D. Saussie, and J. Le Ny, “Dynamic landing of an autonomous quadrotor on a moving platform in turbulent wind conditions,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Vancouver, BC, Canada, 2017, pp. 3428–3435.
- [5] Kim, E., Long, K. (n.d.). Amazon will pay a whopping 63 per package for drone delivery in 2025 and it shows just how the company is still grappling with cost issues for last-mile delivery. Business Insider. <https://www.businessinsider.com/amazon-prime-air-drone-delivery-cost-63-per-package-2025-2022-4>

APPENDIX A DERIVATION OF DYNAMICS

$$A_c = \frac{\partial f}{\partial x} \Big|_{x_{eq}, u_{eq}} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & -g & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (13)$$

$$B_c = \frac{\partial f}{\partial u} \Big|_{x_{eq}, u_{eq}} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{1}{m_l} & \frac{1}{m_l} \\ \frac{m_l}{2I_{yy}} & -\frac{m_l}{2I_{yy}} \end{bmatrix} \quad (14)$$

$$A_d = e^{A_c T_s} \quad (15)$$

$$B_d = \left(\int_0^{T_s} e^{A_c \tau} d\tau \right) B_c \quad (16)$$

APPENDIX A: SIMULATION PARAMETERS

The specific values used to tune the Linear MPC and NMPC controllers, as well as the environmental parameters for the simulation, are listed in Table I.

TABLE I: Simulation and Controller Parameters

Parameter	Symbol	Value
<i>Controller Tuning</i>		
Prediction Horizon	N	50(Lin), 20(NL) steps
Sampling Time	T_s	0.05 s
State Weights (Diag)	Q	[30, 30, 20, 5, 15]
Control Weights (Diag)	R_{linear}	[1.0, 1.0]
Control Weights (Diag)	$R_{Nonlinear}$	[0.1, 0.1]
Slack Penalty	Q_s	1000
<i>Constraints</i>		
Thrust Limits	u_{min}, u_{max}	0, $1.5 \times u_{hover}$
Pitch Limit (Linear)	θ_{lim}	$\pm 25^\circ$
Pitch Limit (NMPC)	θ_{lim}	$\pm 45^\circ$
<i>Environment & Model</i>		
Mass	m	1.0 kg
Arm Length	l	0.3 m
Wind Mean Force	F_{wind}	$[1.5, -0.2]^T$ N
Wind Turbulence	σ	0.5
Platform Max Speed	v_{plat}	2.5 m/s
Platform Max Grade	ϕ_{plat}	$\pm 15^\circ$