MPC BASED AUTONOMOUS LANDING OF MAV ON A MOVING PLATFORM

Instructor **:-** Dr. Indranil Saha
Course **:- CS637** (Fall 2022-23)
Contributor :- Rahul Rustagi (200756), B.Anshuman (200259), Shubham Kumar (200967)

Related Links:

LINK OF REFERRED LITERATURE

LINK OF IMPLEMENTATION

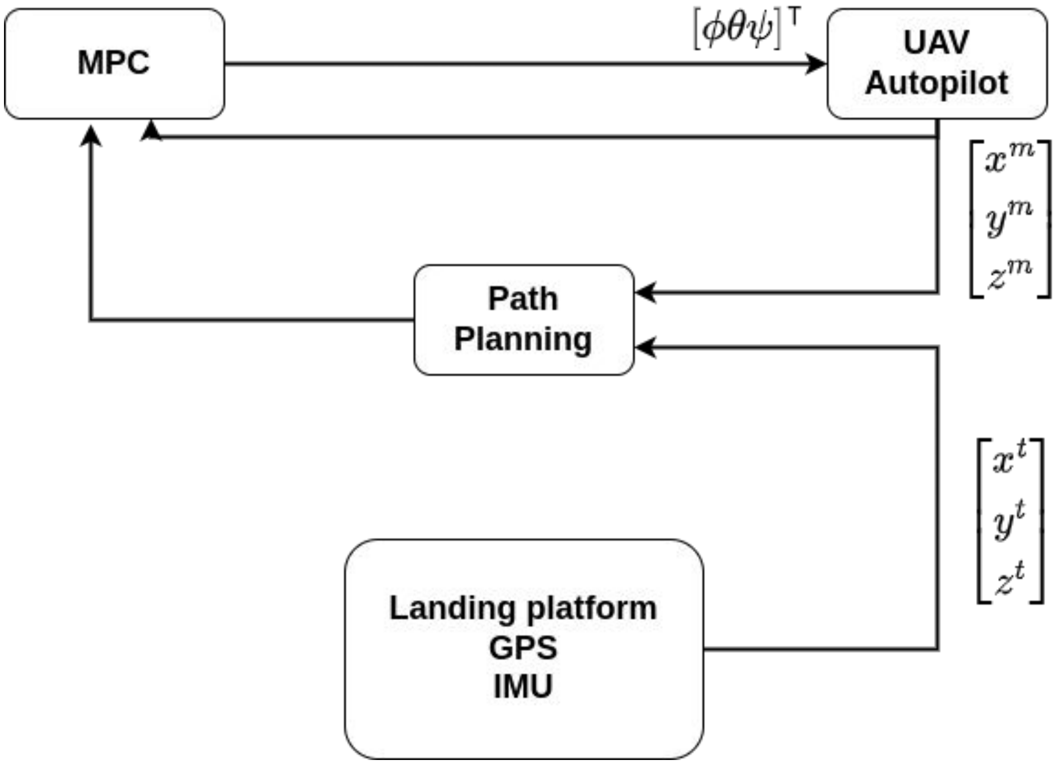
INTRODUCTION

- Autonomous landing of unmanned aerial vehicles (UAVs) on moving targets has the potential to resolve many limitations.
- Here we present an MPC-based guidance and control system for a MAV to land autonomously on a moving landing platform under dynamic uncertainties

RELATED WORK

- An extended back-stepping nonlinear control for landing rotary wing UAVs that are attached to their mobile platforms via tethers has been implemented
- Learning based and intelligent control methods such as fuzzy logic based controllers and adaptive neural networks have also been employed to achieve optimal control policies under uncertainties and disturbances
- However, a control loop for model predictive controller (MPC) would be a better choice for this application

APPROACH


MPC
$[\phi\theta\psi]^\top$
UAV
Autopilot
$\begin{bmatrix} x^m \\ y^m \\ z^m \end{bmatrix}$
Path
Planning
$\begin{bmatrix} x^t \\ y^t \\ z^t \end{bmatrix}$
Landing platform
GPS
IMU

MODEL DYNAMICS

- The UAV dynamics are governed by Newton-Euler equations.

$$ma = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} + R_B^W \begin{bmatrix} 0 \\ 0 \\ -T \end{bmatrix} + F_D$$

- Using the roll pitch and altitude parametrization.

$$m \begin{bmatrix} \ddot{p}^n \\ \ddot{p}^e \\ \ddot{p}^d \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} - \begin{bmatrix} C_\phi S_\theta C_\psi + S_\phi S_\psi \\ C_\phi S_\theta S_\psi + S_\phi C_\psi \\ C_\phi C\theta \end{bmatrix} T - k_d \begin{bmatrix} \dot{p}^n \| \dot{p}^n \| \\ \dot{p}^e \| \dot{p}^e \| \\ \dot{p}^d \| \dot{p}^d \| \end{bmatrix}$$

- For constant flight altitude we have $\ddot{p}^d = 0$ and $\dot{p}^d = 0$. Further assuming $\psi = 0$ and $T = mg/C_\phi C_\theta$.

$$m \begin{bmatrix} \ddot{p}^n \\ \ddot{p}^e \end{bmatrix} = mg \begin{bmatrix} -tan\theta \\ tan\phi/cos\theta \end{bmatrix} - k_d \begin{bmatrix} \dot{p}^n \|\dot{p}^n\| \\ \dot{p}^e \|\dot{p}^e\| \end{bmatrix}$$

- Under the assumption of non aggressive maneuvers, the state of the UAV remain near equilibrium point($\theta = 0$, $\phi = 0$).Linearizing the dynamics about this point.

$$\begin{bmatrix} \ddot{p}^n \\ \ddot{p}^e \end{bmatrix} = -k_d/m \begin{bmatrix} \dot{p}^n \|\dot{p}^n\| \\ \dot{p}^e \|\dot{p}^e\| \end{bmatrix} + g \begin{bmatrix} -\theta \\ \phi \end{bmatrix}$$

- Finally the state representation of the linearized dynamics of the UAV can be written as :

$$\dot{x}^m = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -k_d/m0 & \\ 0 & 0 & 0 & -k_d/m \end{bmatrix} x^m + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ -g & 0 \\ 0 & g \end{bmatrix} u$$

- Where $u = \begin{bmatrix} \theta & \phi \end{bmatrix}^T$ is the input to the system and $x_k = \begin{bmatrix} p_k^{m,n} & p_k^{m,e} & \ddot{p}_k^{m,n} & \ddot{p}_k^{m,e} \end{bmatrix}^T$ represents the state vector of the UAV dynamics.

MODEL PREDICTIVE CONTROL

- We have used a linear MPC control for controlling the motion of the drone
- The predicted state with m calculated future control inputs are given by

$$x_{k+p} = A^p x_k + A^{P-1} B u_k + ... + B u_{k+m-1} = A^p x_k + A^{p-m} \sum_{i=1}^{m} A^{i-1} B u_{k+m-i}$$

Where $$x_k = \begin{bmatrix} p_k^{m,n} & p_k^{m,e} & \dot{p}_k^{m,n} & \dot{p}_k^{m,e} \end{bmatrix}^T$$

Where x_{k+p} is the P -step prediction of the state at time k with p > m.
Hence, the augmented state predictions are

$$X = S^x x_k + S^u u_m$$

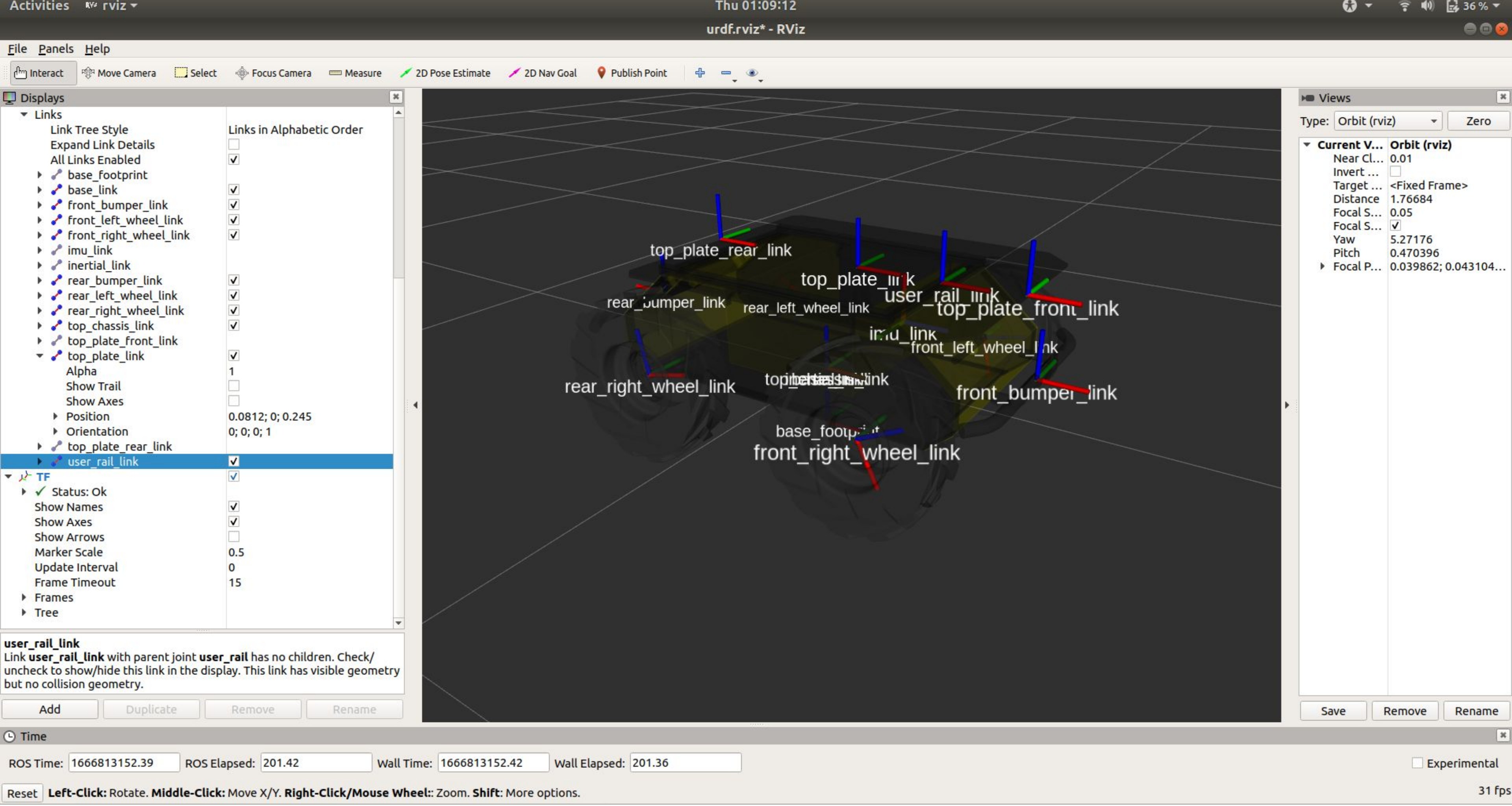
- The quadratic cost function used in this work is formulated as :

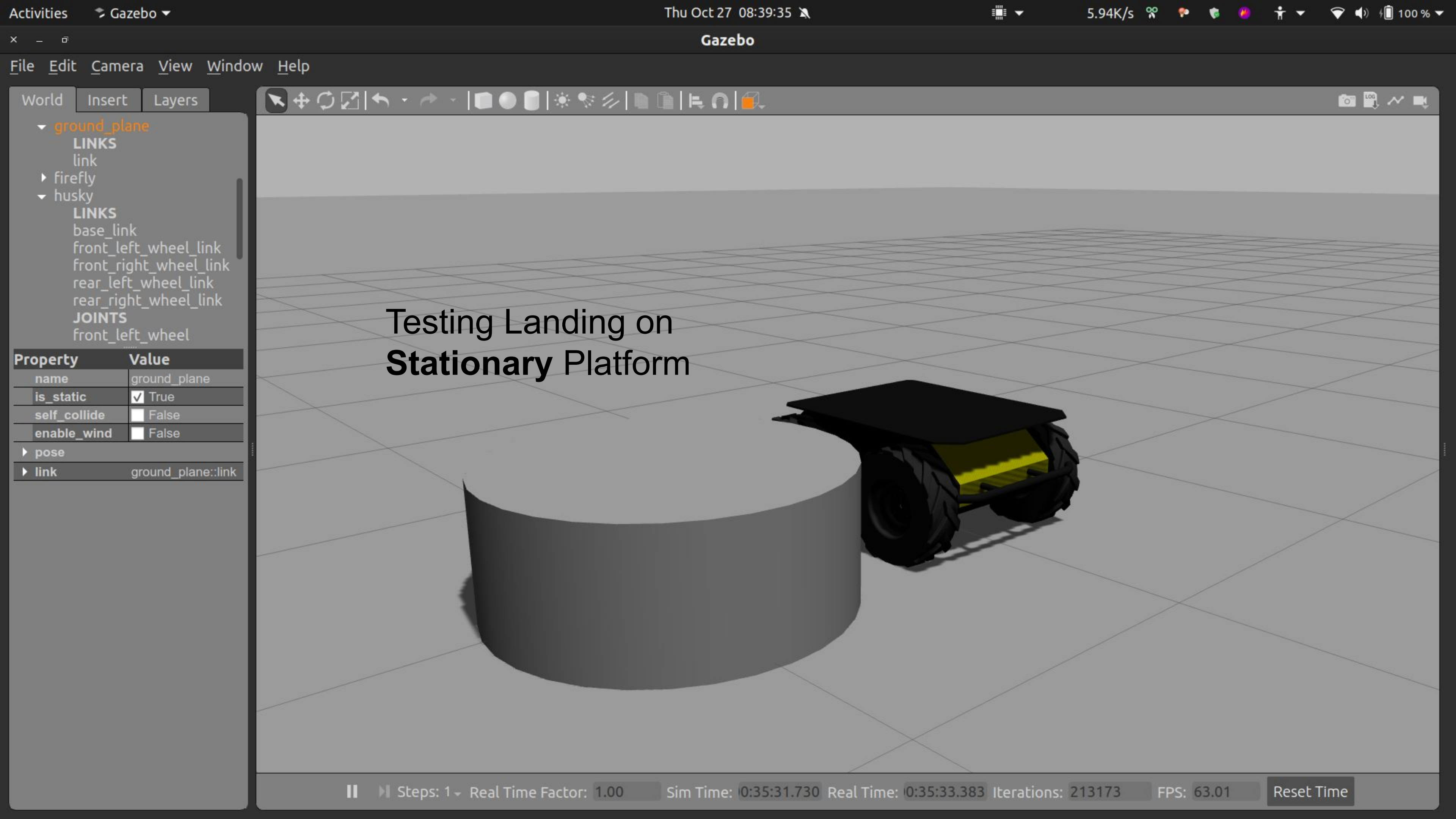
$$j_k = q\tilde{x}^T Q \tilde{x} + (1-q) u_m^T R u_m$$

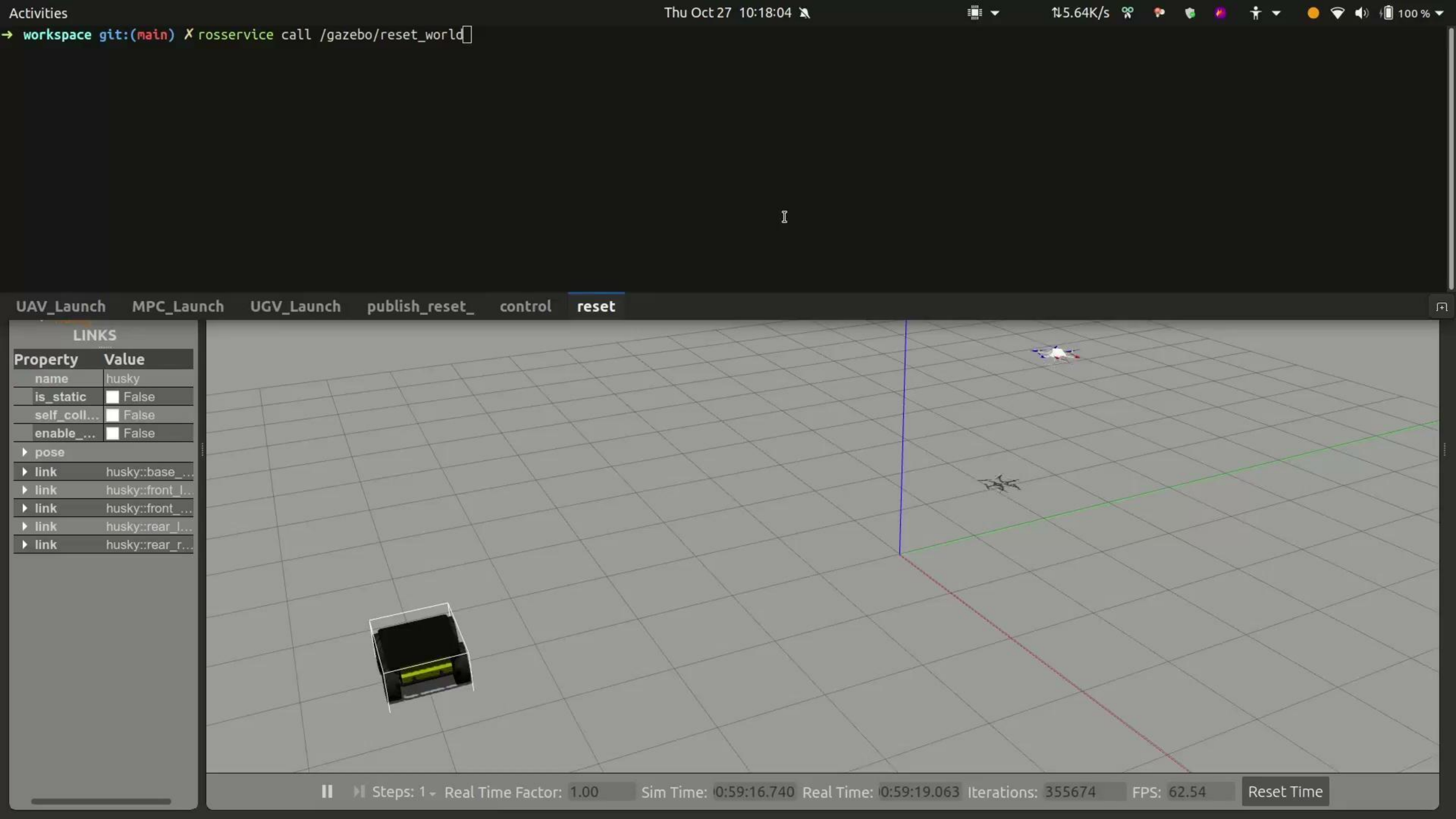
$$\tilde{x} = r_p - X$$

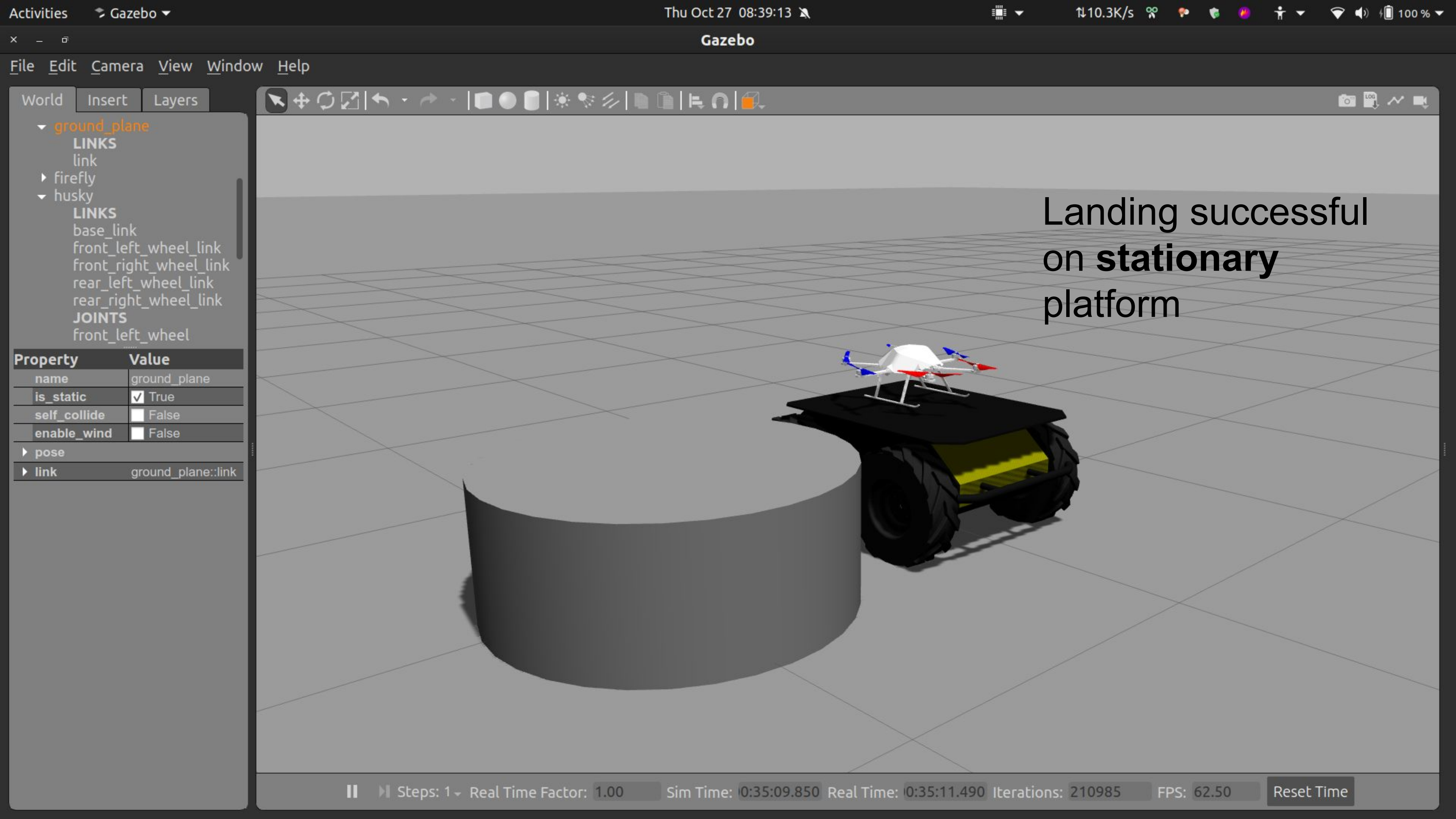
Here Q and R are state and input weights matrices.

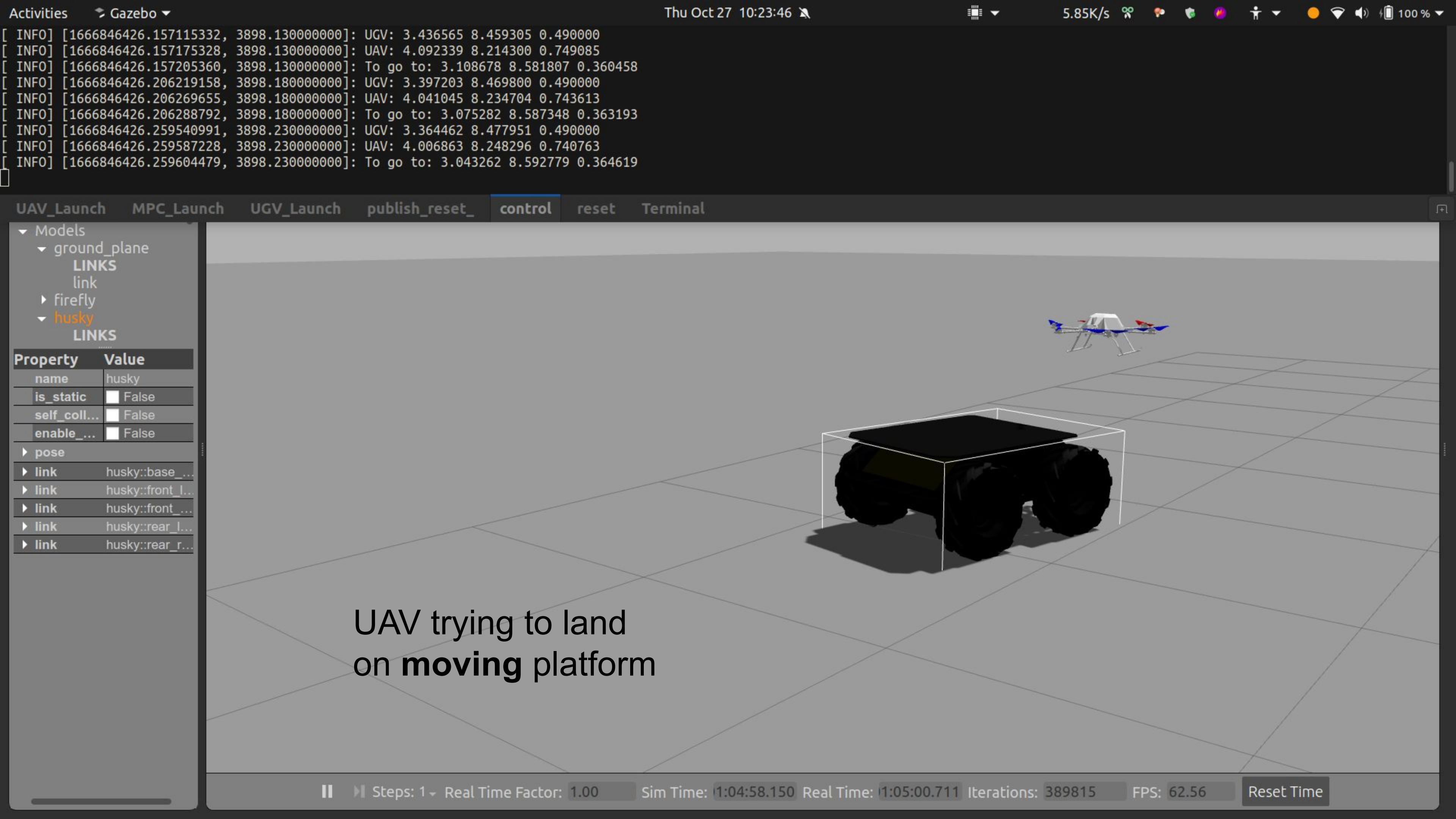
IMPLEMENTATION

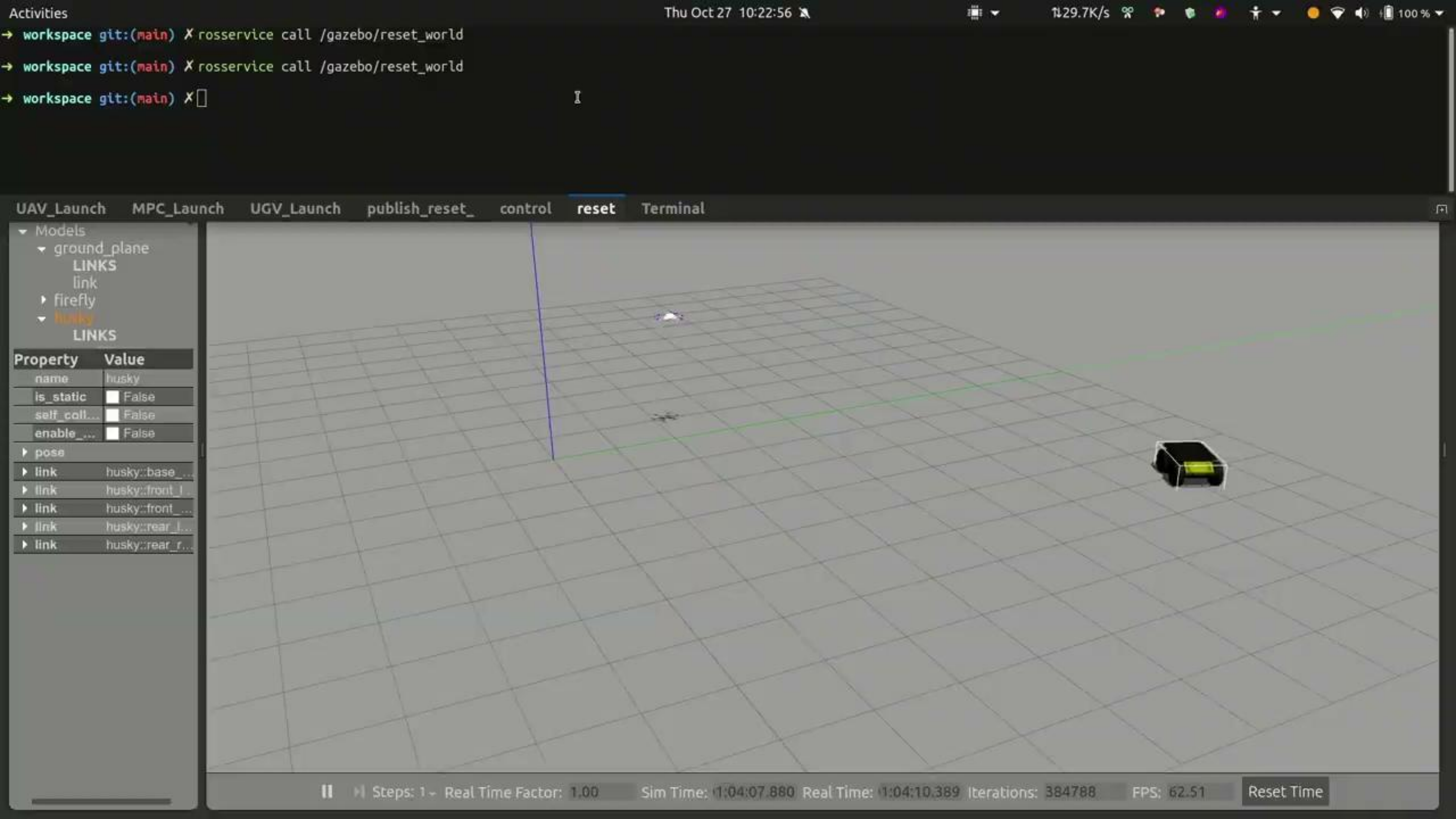

Activities
rviz
Thu 01:09:12
36 %
urdf.rviz* - RViz
File
Panels
Help
Interact
Move Camera
Select
Focus Camera
Measure
2D Pose Estimate
2D Nav Goal
Publish Point
Displays
Links
Link Tree Style
Links in Alphabetic Order
Expand Link Details
All Links Enabled
base_footprint
base_link
front_bumper_link
front_left_wheel_link
front_right_wheel_link
imu_link
inertial_link
rear_bumper_link
rear_left_wheel_link
rear_right_wheel_link
top_chassis_link
top_plate_front_link
top_plate_link
Alpha
1
Show Trail
Show Axes
Position
0.0812; 0; 0.245
Orientation
0; 0; 0; 1
top_plate_rear_link
user_rail_link
TF
Status: Ok
Show Names
Show Axes
Show Arrows
Marker Scale
0.5
Update Interval
0
Frame Timeout
15
Frames
Tree
user_rail_link
Link user_rail_link with parent joint user_rail has no children. Check/uncheck to show/hide this link in the display. This link has visible geometry but no collision geometry.
Add
Duplicate
Remove
Rename
top_plate_rear_link
top_plate_link
user_rail_link
top_plate_front_link
rear_bumper_link
rear_left_wheel_link
imu_link
front_left_wheel_link
rear_right_wheel_link
front_bumper_link
base_footprint
front_right_wheel_link
Views
Type:
Orbit (rviz)
Zero
Current V...
Orbit (rviz)
Near Cl...
0.01
Invert ...
Target ...
<Fixed Frame>
Distance
1.76684
Focal S...
0.05
Focal S...
Yaw
5.27176
Pitch
0.470396
Focal P...
0.039862; 0.043104...
Save
Remove
Rename
Time
ROS Time:
1666813152.39
ROS Elapsed:
201.42
Wall Time:
1666813152.42
Wall Elapsed:
201.36
Experimental
Reset
Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click/Mouse Wheel:: Zoom. Shift: More options.
31 fps

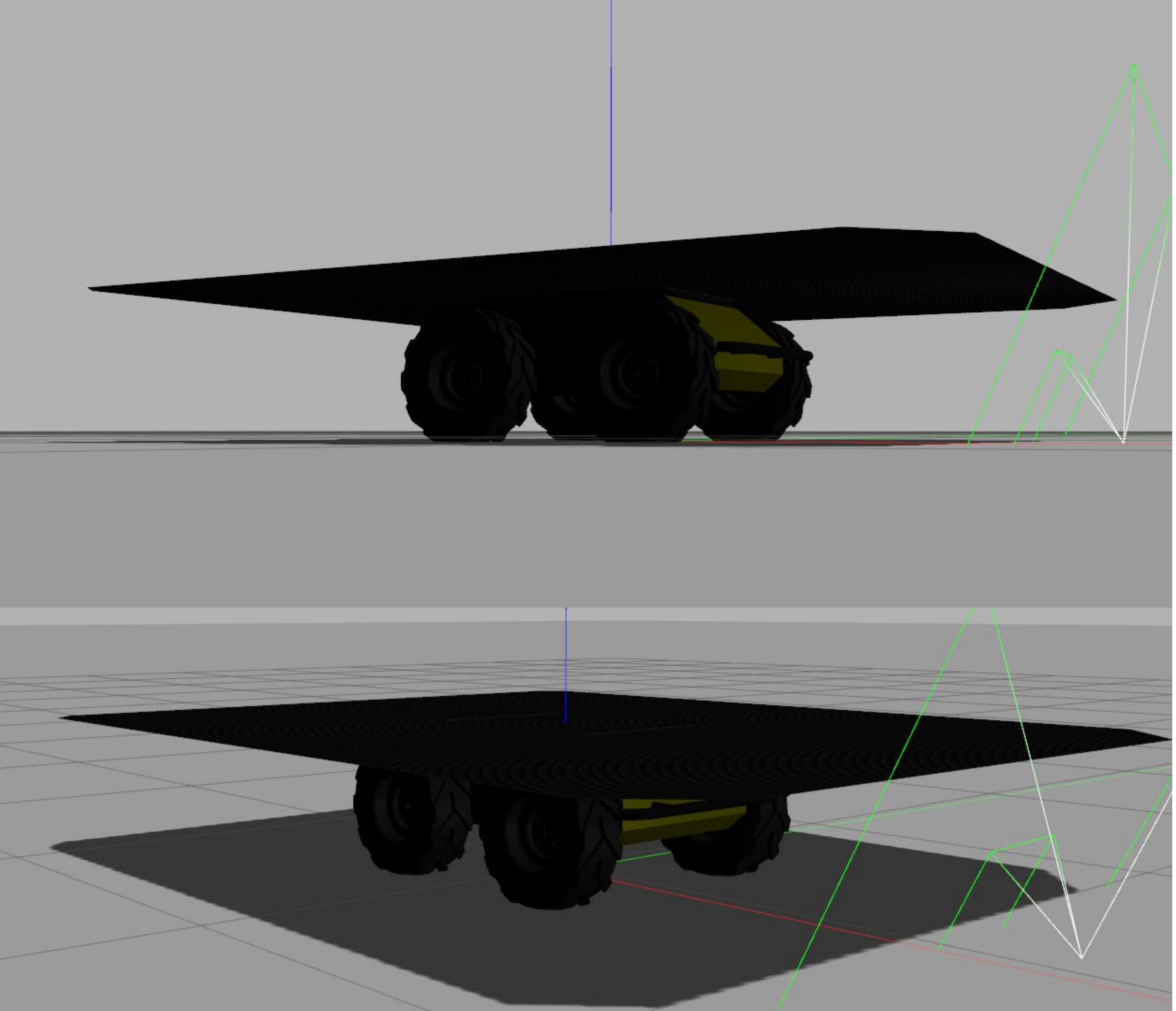

Activities
Gazebo
Thu Oct 27 08:39:35
5.94K/s
100 %
Gazebo
File Edit Camera View Window Help
World Insert Layers
ground_plane
LINKS
link
firefly
husky
LINKS
base_link
front_left_wheel_link
front_right_wheel_link
rear_left_wheel_link
rear_right_wheel_link
JOINTS
front_left_wheel
Property Value
name ground_plane
is_static True
self_collide False
enable_wind False
pose
link ground_plane::link
Testing Landing on
Stationary Platform
Steps: 1
Real Time Factor: 1.00
Sim Time: 0:35:31.730
Real Time: 0:35:33.383
Iterations: 213173
FPS: 63.01
Reset Time


Activities
Thu Oct 27 10:18:04
5.64K/s
100 %
→ workspace git:(main) ✗ rosservice call /gazebo/reset_world
UAV_Launch
MPC_Launch
UGV_Launch
publish_reset_
control
reset
LINKS
Property
Value
name
husky
is_static
False
self_coll...
False
enable_...
False
pose
link
husky::base_...
link
husky::front_l...
link
husky::front_...
link
husky::rear_l...
link
husky::rear_r...
Steps: 1
Real Time Factor:
1.00
Sim Time:
0:59:16.740
Real Time:
0:59:19.063
Iterations:
355674
FPS:
62.54
Reset Time


Landing successful
on **stationary**
platform
Activities
Gazebo
Thu Oct 27 08:39:13
Gazebo
File Edit Camera View Window Help
World Insert Layers
ground_plane
LINKS
link
firefly
husky
LINKS
base_link
front_left_wheel_link
front_right_wheel_link
rear_left_wheel_link
rear_right_wheel_link
JOINTS
front_left_wheel
Property Value
name ground_plane
is_static True
self_collide False
enable_wind False
pose
link ground_plane::link
Steps: 1 Real Time Factor: 1.00 Sim Time: 0:35:09.850 Real Time: 0:35:11.490 Iterations: 210985 FPS: 62.50 Reset Time


Activities
Gazebo
Thu Oct 27 10:23:46
5.85K/s
100 %
[INFO] [1666846426.157115332, 3898.130000000]: UGV: 3.436565 8.459305 0.490000
[INFO] [1666846426.157175328, 3898.130000000]: UAV: 4.092339 8.214300 0.749085
[INFO] [1666846426.157205360, 3898.130000000]: To go to: 3.108678 8.581807 0.360458
[INFO] [1666846426.206219158, 3898.180000000]: UGV: 3.397203 8.469800 0.490000
[INFO] [1666846426.206269655, 3898.180000000]: UAV: 4.041045 8.234704 0.743613
[INFO] [1666846426.206288792, 3898.180000000]: To go to: 3.075282 8.587348 0.363193
[INFO] [1666846426.259540991, 3898.230000000]: UGV: 3.364462 8.477951 0.490000
[INFO] [1666846426.259587228, 3898.230000000]: UAV: 4.006863 8.248296 0.740763
[INFO] [1666846426.259604479, 3898.230000000]: To go to: 3.043262 8.592779 0.364619
UAV_Launch
MPC_Launch
UGV_Launch
publish_reset_
control
reset
Terminal
Models
ground_plane
LINKS
link
firefly
husky
LINKS
Property
Value
name
husky
is_static
False
self_coll...
False
enable_...
False
pose
link
husky::base_...
link
husky::front_l...
link
husky::front_...
link
husky::rear_l...
link
husky::rear_r...
UAV trying to land
on **moving** platform
Steps: 1
Real Time Factor: 1.00
Sim Time: 1:04:58.150
Real Time: 1:05:00.711
Iterations: 389815
FPS: 62.56
Reset Time


Activities
Thu Oct 27 10:22:56
29.7K/s
100 %
→ workspace git:(main) ✗ rosservice call /gazebo/reset_world
→ workspace git:(main) ✗ rosservice call /gazebo/reset_world
→ workspace git:(main) ✗
UAV_Launch
MPC_Launch
UGV_Launch
publish_reset_
control
reset
Terminal
Models
ground_plane
LINKS
link
firefly
LINKS
Property
Value
name
husky
is_static
False
self_coll...
False
enable_...
False
pose
link
husky::base_...
link
husky::front_l...
link
husky::front_...
link
husky::rear_l...
link
husky::rear_r...
Steps: 1
Real Time Factor: 1.00
Sim Time: 1:04:07.880
Real Time: 1:04:10.389
Iterations: 384788
FPS: 62.51
Reset Time



Integrated a large plate on top of moving platform

ToDo:
Implementing a robust path planning problem to achieve landing on the **moving** platform

Summary

- Literature review of Model Predictive Control
- Used MPC for position control of UAV
- Integrated path planning with MPC for autonomous landing
- Implemented stationary and moving platform with coded trajectory
- Integrated and Simulated the modules in gazebo

Further Work

- Used an extended platform for giving more margin for the UAV to land on.
- Included platform path predictive aspect for avoiding steady state chasing of platform.
- Tried using Velocity control through MPC for the above problem.
- Implemented 3 trajectory planning algorithms:
  - XY-approach then changing altitude once XY-radius is within a threshold.
  - Half waypoint approach that directs the next position be the midpoint of UAV and platform pose.
  - 1/k waypoint approach that extends the above idea.

Contribution

- Shubham Kumar - Read Literature on MPC, prepared presentation and integrated modules
- Rahul Rustagi - Implemented stationary/moving platform and coded platform trajectory
- B.Anshuman - Created pipeline for using MPC, and implemented UAV trajectory