HW4

Theo McArn

December 5, 2025

Problem 1: Route Planning on Occupancy Grid Maps

a) A general A* search implementation can be found in my repository in A_star.py.

b) A visualization of the A* path planning performed directly on 8-connected occupancy grid can be seen in Figure 1. It has a path length of 725 nodes in the graph.

c) The contruction of a Probabilistic Road Map (PRM) is implemented in my repository in prob_road_map.py.

d) A visualization of the constructed PRM can be seen in Figure 2.

e) A visualization of the A* path planning performed on this PRM can be seen in Figure 3. It has a path length of 19 nodes in the graph.

Problem 2: PID Altitude Control

a) **Proportional Control**: A plot of the P Controllers with varying values for K_p can be seen in Figure 4. It seems that as the gain increases, the oscillation frequency increases as well.

b) **PD Control (Underdamped)**: We can tune the K_p and K_d gains in a way to ensure an underdamped system by looking at the generalized second order system for the drone:

$$\ddot{y}(t) + 2\zeta\omega_n\dot{y}(t) + \omega_n^2 y(t) = b_0 u(t) \tag{1}$$

where ζ is the damping ratio, and ω_n is the natural oscillation ratio. Taking the characteristic equation for this system we get:

$$s^2 + 2\zeta\omega_n s + \omega_n^2 = 0 \tag{2}$$

Solving for the roots of this quadratic we get:

$$s_{1,2} = -\zeta\omega_n \pm \omega_n\sqrt{\zeta^2 - 1} \tag{3}$$

When $\zeta^2 - 1 < 0$, therefore when $0 < \zeta < 1$, the system has two complex roots, and is an underdamped system. By setting $\zeta = 0.5$, and setting our ideal settling time, t_s, as 3 seconds, we can use the following equation to determine ω_n:

$$t_s \approx \frac{3}{\zeta\omega_n} \tag{4}$$

By selecting a ζ and ω_n we can calculate a respective K_p and K_d for our drone model with the following equations:

$$K_p = \frac{\omega_n^2 m}{4k_T} \tag{5a}$$

$$K_d = \frac{\zeta \omega_n m}{2k_T} \tag{5b}$$

where k_T is the coefficient of thrust for the drone propellers and m is the drone mass.

A plot of an underdamped PD Controller, using these determined parameters can be seen in Figure 5.

Intuitively, we can verify that this system is underdamped because the K_p gain is much higher than the K_d gain (about $2\times$ larger), and we can see heavy overshoot.

c) **PD Control (Overdamped)**: We can tune the K_p and K_d gains in the same way as part (b) to ensure that the system is instead overdamped system.

When $\zeta^2 - 1 > 0$, therefore when $\zeta > 1$, this results in two real roots for the system (determined from equation (3)), and therefore and overdamped system. To make our system overdamped, we will set $\zeta = 1.5$, and calculate ω_n from equation (4) where the settling time, t_s, is again 3 seconds.

We can then use equations (5a) and (5b) to calculate K_p and K_d respectfully.

A plot of an overdamped PD Controller, using these determined parameters can be seen in Figure 6.

Intuitively, we can verify that this system is overdamped because the K_d gain is much higher than the K_p gain (about $2\times$ larger), and we can see a very long and slow rise time.

Note: The intended settling time using this equation is supposed to be only 3 seconds but the result is a settling time of about 15 seconds. This is likely do to the nauture of the slow rise time of the overdamped system and the approximation in equation (4)

d) Rerunning the controller from part (b) but with an underactuated system, where $u' = 0.95u$, we get the plot seen in Figure 7. We can see that do to this underactuation, there is a lot of steadystate error in the system. By utlizinf the same system but including an integral term to our controller, where $K_i = 10$, we get the plot seen in Figure 8. In this plot, the steady state error is eliminated by the contribution of the integral term in the controller.

Note: the code used to simulate these models and controllers can be found in my repository in pid_control.py.

Problem 3: Inverted Pendulum Control

a) Second-order equations of motion expressed in the form of a first-order differential system for x, where $x = (\theta, \dot{\theta}) \in \mathbb{R}^2$:

$$\dot{x} = f(x, \tau) = \begin{bmatrix} x_2 \\ \frac{-g}{l}\sin(x_1) - \frac{\mu x_2 + \tau}{ml^2} \end{bmatrix} \tag{6}$$

b) **Linearized Stability Analysis at $x^* = (\pi, 0)$:**

$$J_x = \frac{\partial f}{\partial x}(x_1, x_2) = \begin{bmatrix} 0 & 1 \\ \frac{-g}{l}\cos(x_1) & -\frac{\mu}{ml^2} \end{bmatrix} \tag{7}$$

$$J_{x^*} = \frac{\partial f}{\partial x}(\pi, 0) = \begin{bmatrix} 0 & 1 \\ \frac{g}{l} & \frac{-\mu}{ml^2} \end{bmatrix} \tag{8}$$

Calculating the Eigenvalues of this Jacobian, we get:

$$\lambda = \frac{-\frac{\mu}{ml^2} \pm \sqrt{\left(-\frac{\mu}{ml^2}\right)^2 + 4\frac{g}{l}}}{2} \tag{9}$$

Since $g > 0$ and $l > 0$, we can determine that one of the eigenvalues will always be positive. Therefore, this system, without any external τ applied, is unstable at $x^* = (\pi, 0)$

c) **Linear Control Synthesis**:

Plugging in our PD Controller for τ we get:

$$\dot{x} = f(x, k_p, k_d) = \begin{bmatrix} x_2 \\ \frac{-g}{l}\sin(x_1) - \frac{\mu x_2 + k_p \sin(x_1) + k_d x_2}{ml^2} \end{bmatrix} \tag{10}$$

d) Linearized Stability Analysis with PD Controller:

Calculating the Jacobian we get:

$$J_x = \frac{\partial f}{\partial x}(x_1, x_2) = \begin{bmatrix} 0 & 1 \\ \frac{-g}{l}\cos(x_1) + \frac{k_p}{ml^2}\cos(x_1) & -\frac{\mu + k_d}{ml^2} \end{bmatrix} \tag{11}$$

Plugging in for $x^* = (\pi, 0)$:

$$J_{x^*} = \frac{\partial f}{\partial x}(x_1, x_2) = \begin{bmatrix} 0 & 1 \\ \frac{g}{l} - \frac{k_p}{ml^2} & -\frac{\mu + k_d}{ml^2} \end{bmatrix} \tag{12}$$

Taking the Eigenvalues of this Jacobian, we get:

$$\lambda = \frac{-\left(\frac{\mu + k_d}{ml^2}\right) \pm \sqrt{\left(\frac{\mu + k_d}{ml^2}\right)^2 + 4\left(\frac{mgl - k_p}{ml^2}\right)}}{2} \tag{13}$$

In order for both eigenvalues to be negative:

$$\frac{mgl - k_p}{ml^2} < 0 \tag{14a}$$

$$\frac{\mu + k_d}{ml^2} > 0 \tag{14b}$$

Equation (14a) guaranties that both roots have the same sign, and equation (14b) then guaranties that both of those roots are negative.

Simplifying these inequalities, we get the following sufficient conditions for k_p and k_d to guaranty that x^* is an asymptotically stable point:

$$k_p > mgl \qquad\qquad k_d < \mu \tag{15}$$

e) **Lyapunov Analysis**:

Prove $V(x)$ is Non-Negative, and $\nabla^2 V(x^*)$ Positive-Definite:

$$\nabla V(x) = \begin{bmatrix} mgl(2\alpha\cos(x_1) + 1)\sin(x_1) \\ ml^2 x_2 \end{bmatrix} \tag{16}$$

$$\nabla^2 V(x) = \begin{bmatrix} -mgl(2\alpha\sin^2(x_1) - 2\alpha\cos^2(x_1) - \cos(x_1)) \\ ml^2 \end{bmatrix} \tag{17}$$

$V(x^*) = 0$ is a local minimizer if $\nabla^2 V(x^*) \succ 0$, therefore to determine the sufficient condition for α:

$$\nabla^2 V(x^*) = \begin{bmatrix} -mgl(2\alpha\sin^2(\pi) - 2\alpha\cos^2(\pi) - \cos(\pi)) & 0 \\ 0 & ml^2 \end{bmatrix} \tag{18}$$

which simplifies to:

$$\nabla^2 V(x^*) = \begin{bmatrix} -mgl(-2\alpha+1) & 0 \\ 0 & ml^2 \end{bmatrix} \tag{19}$$

In order for $\nabla^2 V(x^*) \succ 0$, the eigenvalues, $-mgl(-2\alpha+1) > 0$ and $ml^2 > 0$

Solving for these inequalities we arrive at the sufficient condition for α such that $V(x^*) = 0$ is a local minimizer:

$$\alpha > \frac{1}{2} \tag{20}$$

Since $V(x^*) = 0$ is a local minimizer, this means that there exists some neighborhood containing x^* such that $V(x) \geq 0$, and x^* is the unique point satisfying $V(x^*) = 0$

f) **Prove V(x) is non-increasing:**

$$\dot{V}(x) = \frac{d}{dt}(V(x)) = \nabla V \cdot \dot{x} = \begin{bmatrix} mgl(2\alpha\cos(x_1)+1)\sin(x_1) \\ ml^2 x_2 \end{bmatrix}^T \begin{bmatrix} x_2 \\ \frac{-g}{l}\sin(x_1) - \frac{\mu x_2 + \tau}{ml^2} \end{bmatrix} \tag{21}$$

Expanding this expression and simplifying we get:

$$\dot{V}(x) = 2\alpha mlg\cos(x_1)\sin(x_1)x_2 - g\mu x_2^2 + \tau x_2 \tag{22}$$

If we assume that

$$\tau = -2\alpha mlg\cos(x_1)\sin(x_1) \tag{23}$$

then $\dot{V}$ simplifies to:

$$\dot{V}(x) = -g\mu x_2^2 \tag{24}$$

Since $g > 0$, $\mu > 0$ and $x_2^2 \geq 0$, $\dot{V}(x) \leq 0$. Therefore $V(x)$ is non-increasing.

g) Plugging this new control law, τ, into dynamics equation, we get:

$$\dot{x} = \begin{bmatrix} x_2 \\ \frac{-g}{l}\sin(x_1) - \frac{\mu x_2 - 2\alpha mlg\cos(x_1)\sin(x_1)}{ml^2} \end{bmatrix} \tag{25}$$

h) The set S of stationary points for the control system is defined as $S = \{x : \dot{x} = 0\}$

Therefore the system is stationary when:

$$x_2 = 0 \tag{26a}$$

$$\frac{-g}{l}\sin(x_1) - \frac{\mu x_2 - 2\alpha mlg\cos(x_1)\sin(x_1)}{ml^2} = 0 \tag{26b}$$

Solving for 26b, we get:

$$-\sin(x_1)(2\alpha\cos(x_1)+1) = 0 \tag{27}$$

Therefore when $\sin(x_1) = 0$ and $x_2 = 0$ the system is stationary. This leaves us with a set S, with two stationary points:

$$S = \{(0,0), (\pi, 0)\}$$

i) Proved $\dot{V}(x) \leq 0$ in part (f)

j) Since $V(x)$ is non-negative and $\nabla^2 V(x^*)$ is positive-definite, and $\dot{V}(x) \leq 0$, then x^* is stable in the sense of Lyapunov, but it is not necessarily asymptotically stable since $\dot{V}(x)$ is not **strictly** less than 0.

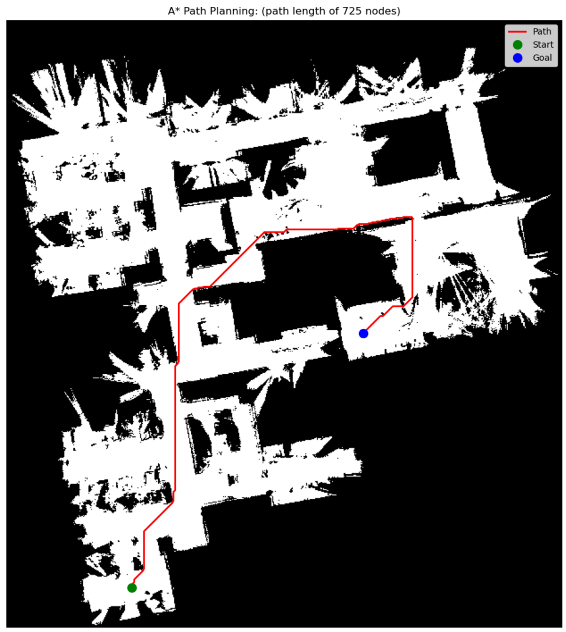

Figure 1: A* Path Planning Performed on Occupancy Grid

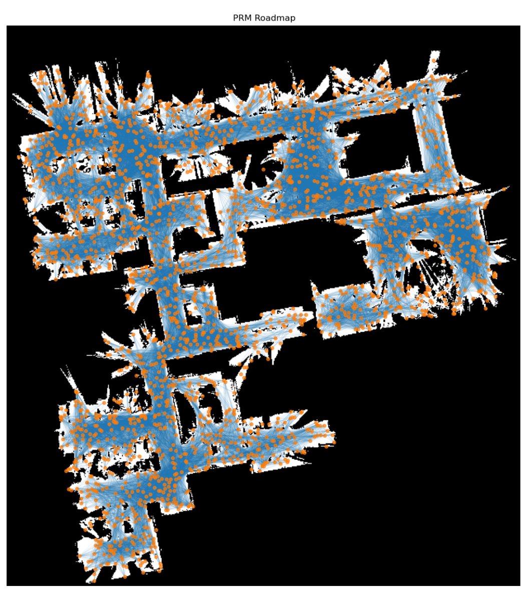

Figure 2: Constructed Probabilistic Road Map

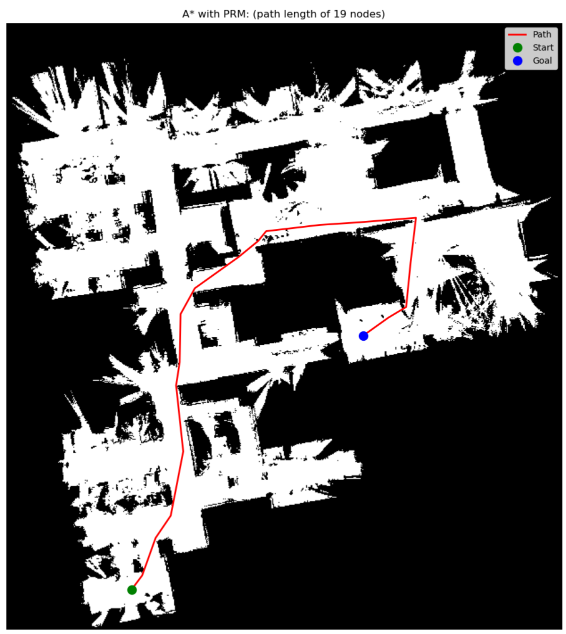

Figure 3: A* Path Planning Performed on PRM

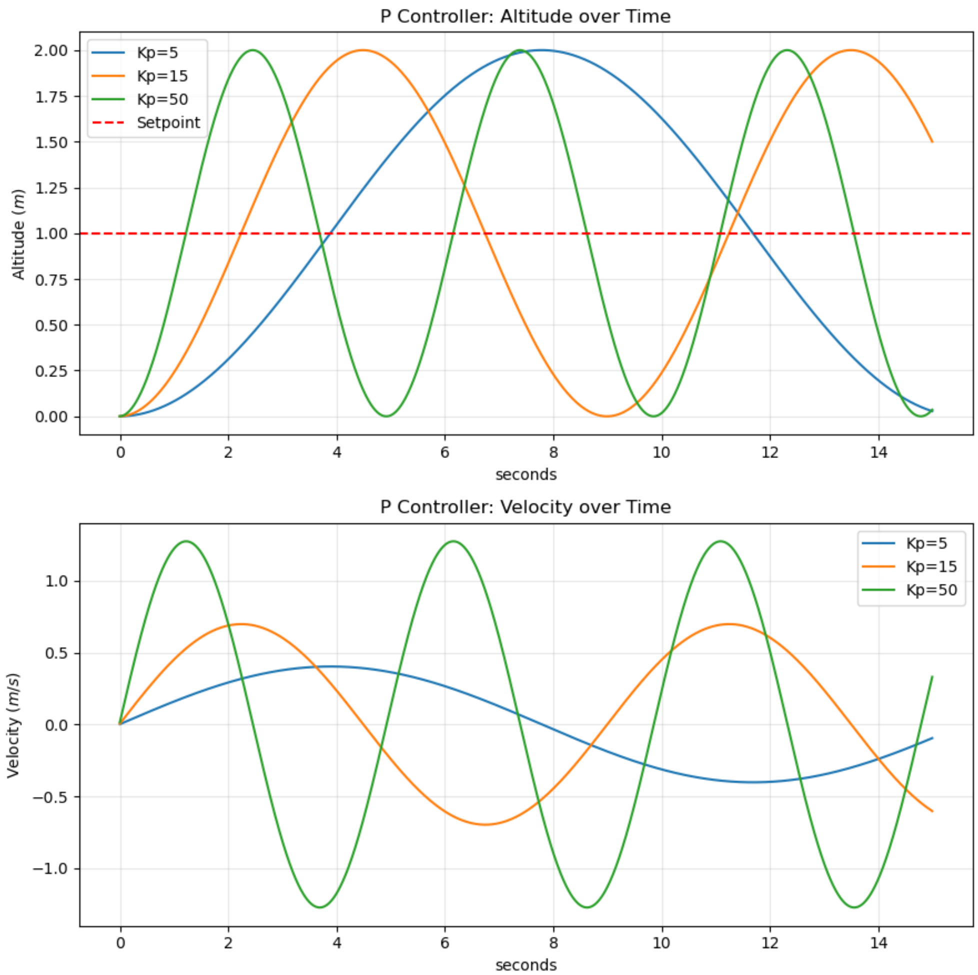

Figure 4: Proportional Controllers with Varying Values of K_p

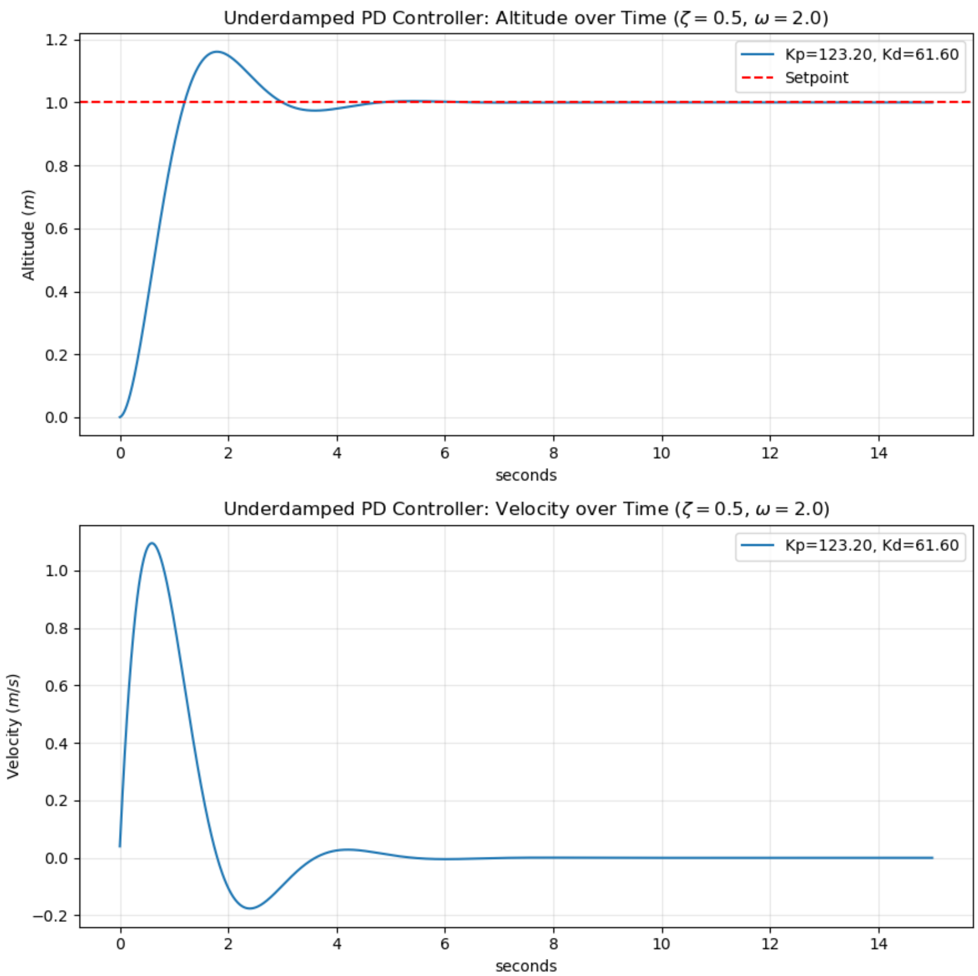

Figure 5: PD-Controller (Underdamped) that converges in about 3 seconds

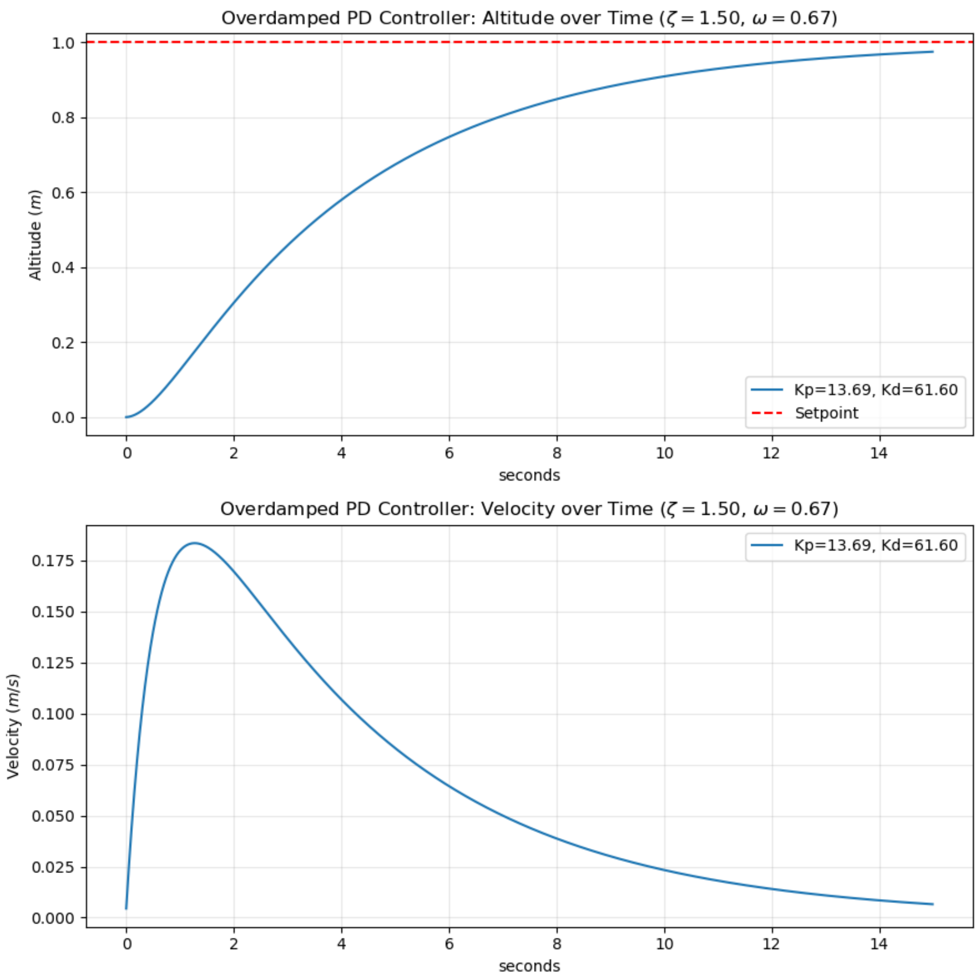

Figure 6: PD-Controller (Overdamped)

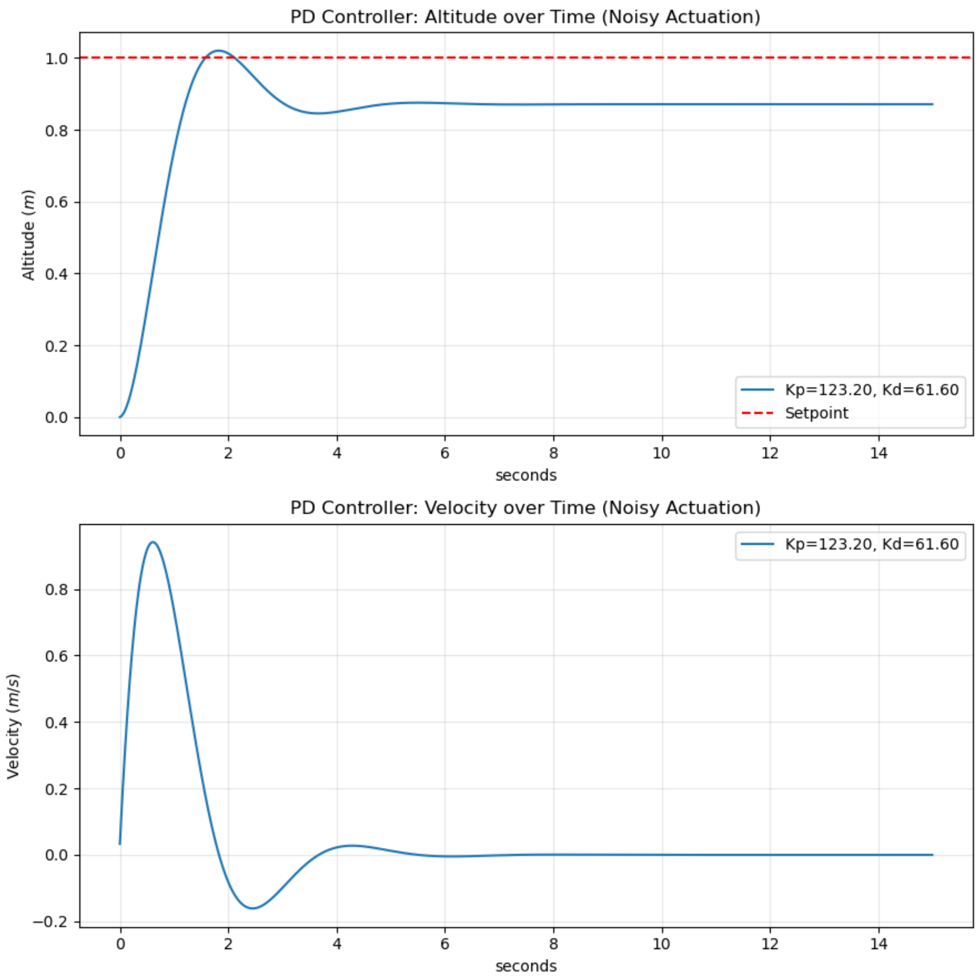

Figure 7: PD-Controller (Underactuated)

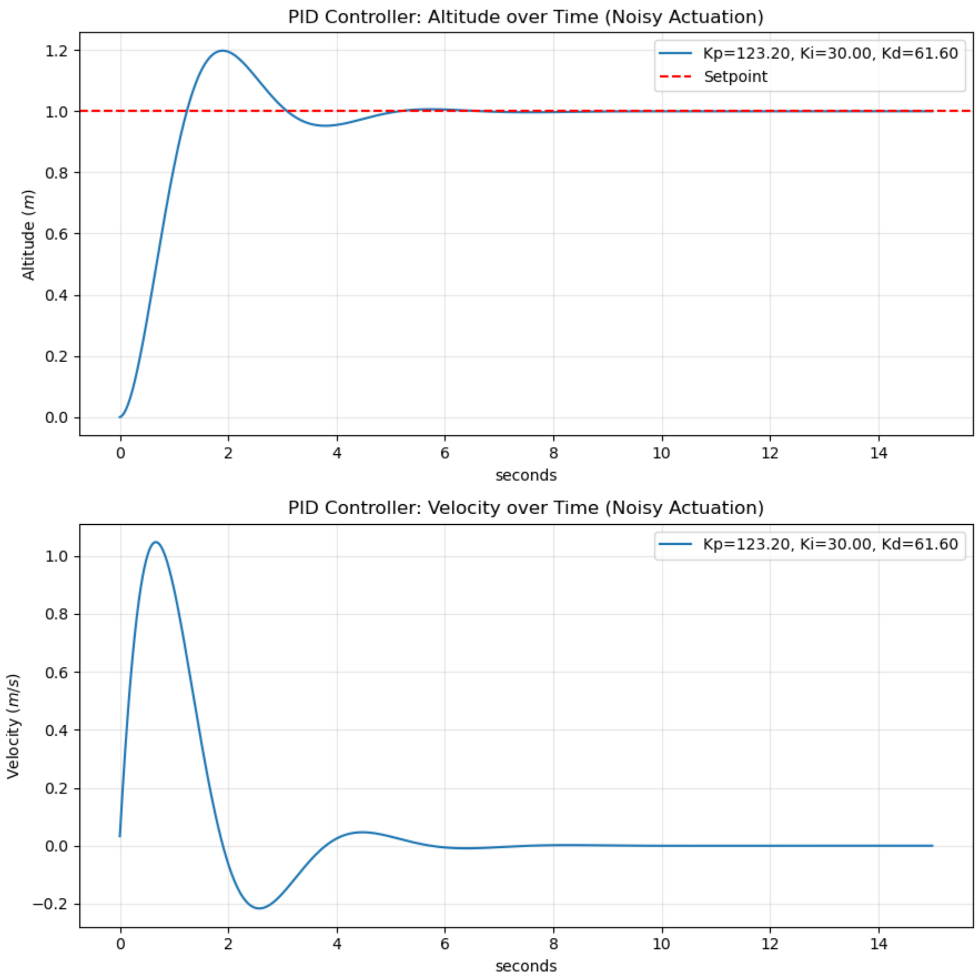

Figure 8: PID-Controller (Underactuated)