DP0: Control of a platform with a reaction wheel in gravity

Timothy Bretl
University of Illinois at Urbana-Champaign

In this report, we design a controller to stabilize an inverted rotating platform with a reaction wheel in gravity, and test this controller with experiments in simulation. We linearize equations of motion to derive a state-space model and apply eigenvalue placement to choose linear state feedback that would, in theory, produce a stable closed-loop system that achieves target values of peak time and peak overshoot. Our experiments show that we achieved a 30% smaller peak time and a 120% larger peak overshoot than intended, and that stable closed-loop responses were obtained only when starting from initial conditions close to equilibrium.

I. Nomenclature

A, B	=	matrices that describe the state-space model
J_1, J_2	=	moments of inertia of the platform and wheel ($\text{kg} \cdot \text{m}^2$)
K (k_1, k_2, k_3)	=	gain matrix (and the three elements of this matrix) for linear state feedback
l	=	distance from center of platform to center of wheel (m)
m, g	=	mass of wheel (kg) and acceleration of gravity (m/s^2)
$q_{1e}, v_{1e}, v_{2e}, \tau_e$	=	equilibrium values of nonlinear states and inputs
q_1, v_1, v_2	=	platform angle (rad) platform angular velocity (rad/s), and wheel angular velocity (rad/s)
τ	=	torque applied by wheel to platform ($\text{N} \cdot \text{m}$)
T_p, M_p	=	peak time (s) and peak overshoot
x, u	=	state and input of linear system
ζ, μ	=	state and input of nonlinear system

II. Introduction

Consider an inverted platform in a gravitational field that is constrained to rotate about one axis and to which is mounted a reaction wheel (Fig. 2). Suppose it is possible to measure the platform angle, the platform angular velocity, and the wheel angular velocity, and to choose a torque that is applied by the platform to the wheel. Our goal is to design a controller that stabilizes the inverted platform (and that maintains the wheel at some non-zero velocity)—in particular, at a configuration that would otherwise be unstable. This challenge is representative of what we might see when designing attitude determination and control systems for spacecraft in low Earth orbit that use gravity gradient torques to dump momentum [1]. Section III derives a state-space model of our system by linearizing about an equilibrium point and then applies eigenvalue placement to design linear state feedback that would achieve target peak time and peak overshoot, which are common step response specifications [2]. Section IV describes how we implemented and tested our controller in simulation, and in particular describes the experiments we conducted to quantify the extent to which target values of peak time and peak overshoot were achieved as well as the experiments we conducted to identify the *basin of attraction*—the set of initial conditions from which the closed-loop system is stable in simulation. Section V presents the results of these experiments. Section VI concludes with opportunities for future work.

III. Theory

A. Derive state-space model

The motion of the platform and wheel together under gravity is described by

$$\begin{aligned} J_1 \ddot{q}_1 &= \tau - mgl \sin(q_1) \\ J_2 \dot{v}_2 &= -\left(\frac{J_1 + J_2}{J_1}\right)\tau + \left(\frac{J_2}{J_1}\right) mgl \sin(q_1), \end{aligned} \tag{1}$$

where q_1 is the angle of the platform, v_2 is the angular velocity of the wheel, τ is the torque applied by the platform to the wheel through a motor, and all other parameters are given constants. Before proceeding with control design, we will linearize these equations of motion about an equilibrium point that corresponds to some desired platform angle. First, we rewrite (1) as a set of three first-order ordinary differential equations

$$\begin{aligned} \dot{q}_1 &= v_1 \\ \dot{v}_1 &= \left(\tau - mgl\sin(q_1)\right)/J_1 \\ \dot{v}_2 &= -\left(\left((J_1+J_2)\tau + \left(J_2 mgl\sin(q_1)\right)/(J_1 J_2)\right.\right., \end{aligned} \tag{2}$$

where we have introduced a variable v_1 to denote the angular velocity of the platform. If we define

$$\zeta = \begin{bmatrix} q_1 \\ v_1 \\ v_2 \end{bmatrix} \qquad \mu = \begin{bmatrix} \tau \end{bmatrix} \tag{3}$$

then we can rewrite (2) in standard form as

$$\dot{\zeta} = f(\zeta, \mu) \tag{4}$$

where

$$f(\zeta,\mu) = \begin{bmatrix} v_1 \\ \left(\tau - mgl\sin(q_1)\right)/J_1 \\ -\left(\left((J_1+J_2)\tau + \left(J_2 mgl\sin(q_1)\right)/(J_1 J_2)\right.\right. \end{bmatrix}. \tag{5}$$

By solving

$$0 = f(\zeta_e, \mu_e), \tag{6}$$

we see that any equilibrium point of (4) has the form

$$\zeta_e = \begin{bmatrix} n\pi \\ 0 \\ v_{2e} \end{bmatrix} \qquad \mu_e = \begin{bmatrix} 0 \end{bmatrix} \tag{7}$$

for some integer $n \in \mathbb{Z}$ and some choice of wheel velocity $v_{2e} \in \mathbb{R}$. Recognizing that the angles $n\pi$ and $(n+2)\pi$ correspond to the same orientation of the platform for any n, we restrict our choice of equilibrium platform angle to either 0 radians or π radians. Of these two, since we would like to consider what seems to be the more challenging case of stabilizing the platform at an inverted configuration, we choose π radians as the equilibrium platform angle. Since reaction wheels are typically designed to operate most efficiently at non-zero angular velocity, we choose

$$v_{2e} = 50 \text{ rpm} = 5\pi/3 \text{ radians/second} \tag{8}$$

and so arrive at

$$\zeta_e = \begin{bmatrix} \pi \\ 0 \\ 5\pi/3 \end{bmatrix} \qquad \mu_e = \begin{bmatrix} 0 \end{bmatrix}. \tag{9}$$

If we define

$$x = \zeta - \zeta_e \qquad u = \mu - \mu_e, \tag{10}$$

then (4) can be approximated close to the equilibrium point by the linear system

$$\dot{x} = Ax + Bu \tag{11}$$

where

$$A = \left.\frac{\partial f}{\partial \zeta}\right|_{\zeta_e,\mu_e} = \begin{bmatrix} 0 & 1 & 0 \\ mgl/J_1 & 0 & 0 \\ -mgl/J_1 & 0 & 0 \end{bmatrix} \qquad B = \left.\frac{\partial f}{\partial \mu}\right|_{\zeta_e,\mu_e} = \begin{bmatrix} 0 \\ 1/J_1 \\ -(J_1+J_2)/(J_1 J_2) \end{bmatrix}. \tag{12}$$

B. Design linear state feedback

Suppose we choose to apply an input of the form

$$u = -Kx \tag{13}$$

where

$$K = \begin{bmatrix} k_1 & k_2 & k_3 \end{bmatrix} \tag{14}$$

for some choice of gains k_1, k_2, and k_3. The resulting closed-loop system is

$$\dot{x} = (A - BK)x. \tag{15}$$

We will choose gains by eigenvalue placement, applying a common heuristic to choose desired closed-loop eigenvalue locations that achieve given step response specifications [2]. In particular, suppose we want the closed-loop response to a unit disturbance in the platform angle to have a peak time T_p and peak overshoot M_p of about

$$T_p = \pi/4 \approx 0.8 \text{ seconds} \qquad M_p = e^{-\pi/2} \approx 0.2. \tag{16}$$

If the system were second-order, we could achieve these specifications by placing eigenvalues at $-\sigma \pm j\omega$ where

$$\sigma = -\ln(M_p)/T_p = 2 \qquad \omega = \pi/T_p = 4. \tag{17}$$

We place the third eigenvalue at -5σ so that it has a small effect on the closed-loop response relative to these two dominant eigenvalues (i.e., so that the part of the response associated with this third eigenvalue decays much more quickly). In summary, the desired eigenvalue locations are as follows:

$$s_1 = -2 + 4j \qquad s_2 = -2 - 4j \qquad s_3 = -10. \tag{18}$$

The characteristic polynomial that would produce these eigenvalues is

$$(s - s_1)(s - s_2)(s - s_3) = s^3 + 14s^2 + 60s + 200. \tag{19}$$

The characteristic polynomial of the closed-loop system, as a function of the gains k_1, k_2, and k_3, is

$$s^3 + ((-J_1k_3 + J_2k_2 - J_2k_3)/(J_1J_2))\, s^2 + ((k_1 - mgl)/J1)\, s + mglk_3/(J_1J_2). \tag{20}$$

Equating coefficients of (19) and (20), we find that

$$k_1 = mgl + 60J_1 \qquad k_2 = 200J_1\left(\frac{J_1 + J_2}{mgl}\right) + 14J_1 \qquad k_3 = \frac{100J_1J_2}{mgl}. \tag{21}$$

We emphasize that the resulting closed-loop system is asymptotically stable by construction, since all three eigenvalues (18) placed by these gains have negative real part.

IV. Experimental methods

A. Control implementation and simulation architecture

We verified our control design with experiments in simulation. The simulator itself already existed and was implemented using PyBullet [3]. We added a python-based implementation of our controller, starting from template code that was provided with the simulator [4]. In particular, this controller did the following things at a rate of 100 Hz:

- It received measurements of the platform angle, the platform angular velocity, and the wheel angular velocity from the simulator (along with the current time, which we did not use).
- It computed the state x from these measurements, given our choice of equilibrium point (Section III.A).
- It computed the input u from x by application of linear state feedback (Section III.B).
- It computed the torque τ from u, again given our choice of equilibrium point (Section III.A).
- It sent the wheel torque command $-\tau$ to the simulator, where the change in sign was due to our having chosen to define τ as the torque applied by the wheel to the platform, while the simulator expected the controller to specify the equal and opposite torque applied by the platform to the wheel.

Each experiment began at initial conditions—initial values of the platform angle, the platform angular velocity, and the wheel angular velocity—that we chose, and lasted five seconds of simulation time. Both our controller and the simulator were deterministic, so the results of an experiment were always the same if repeated from the same initial conditions. Data produced by each experiment were a time-history (at 100 Hz) of the platform angle, the platform angular velocity, the wheel velocity, the commanded wheel torque, and the actual wheel torque after saturation at torque limits.

B. Experiments to find peak time and peak overshoot

We had designed our controller to achieve a peak time of approximately 0.80 seconds and a peak overshoot of approximately 0.20, starting from a unit disturbance in the platform angle. To quantify the extent to which this goal was met, we conducted a single experiment with the following initial conditions:

$$q_1(0) = q_{1e} + 0.1 = \pi + 0.1 \qquad v_1(0) = v_{1e} = 0 \qquad v_2(0) = v_{2e} = 5\pi/3. \tag{22}$$

These initial conditions correspond to starting with an initial error in platform angle of 0.1 radians and with zero initial error in platform angular velocity and wheel angular velocity. Starting from these initial conditions, we computed the *peak time* as the first non-zero time T_p at which the platform angle reaches a local minimum (i.e., stops decreasing and starts increasing again). We computed the *peak overshoot* as the fraction

$$M_p = \left| \frac{q_1(T_p) - q_{1e}}{q_1(0) - q_{1e}} \right|, \tag{23}$$

in other words as the error in platform angle at the peak time, normalized by the initial error in platform angle.

C. Experiments to find the basin of attraction

We had designed our controller to make the linear system (11) asymptotically stable in closed loop. In theory, this would imply that $x(t) \to 0$ as $t \to \infty$ for any set of initial conditions $x(0)$. In practice, however, the model (11) is only valid near the equilibrium point about which we linearized. For this reason, we conducted a set of experiments to find the *basin of attraction*—that is, to find the set of initial conditions from which the closed-loop system converges to equilibrium (within some reasonable amount of time) in simulation. We assumed that $v_2(0) = v_{2e}$ for all of these experiments and restricted our attention only to changes in the initial platform angle $q_1(0)$ and angular velocity $v_1(0)$. These initial conditions were said to be inside the basin of attraction when the resulting trajectory satisfied

$$\left.\begin{array}{l} |q_1(t) - q_{1e}| < 0.01 \\ |v_1(t) - v_{1e}| < 0.01 \\ |v_2(t) - v_{2e}| < 0.01 \end{array}\right\} \text{ for all } t \in [4, 5]. \tag{24}$$

We found points on the boundary of the basin of attraction by searching along rays in the space of initial errors in platform angle and angular velocity. In particular, for each $n_\theta \in \{0, 1, 2, \ldots, 100\}$ we found the smallest $n_d \in \{0, 1, 2, \ldots\}$ for which the initial conditions

$$q_1(0) = q_{1e} + (\delta n_d) \cos(2\pi n_\theta/100) \qquad v_1(0) = v_{1e} + (\delta n_d) \sin(2\pi n_\theta/100) \qquad v_2(0) = v_{2e} \tag{25}$$

did *not* produce a trajectory satisfying (24), and added that point to our boundary.

V. Results and discussion

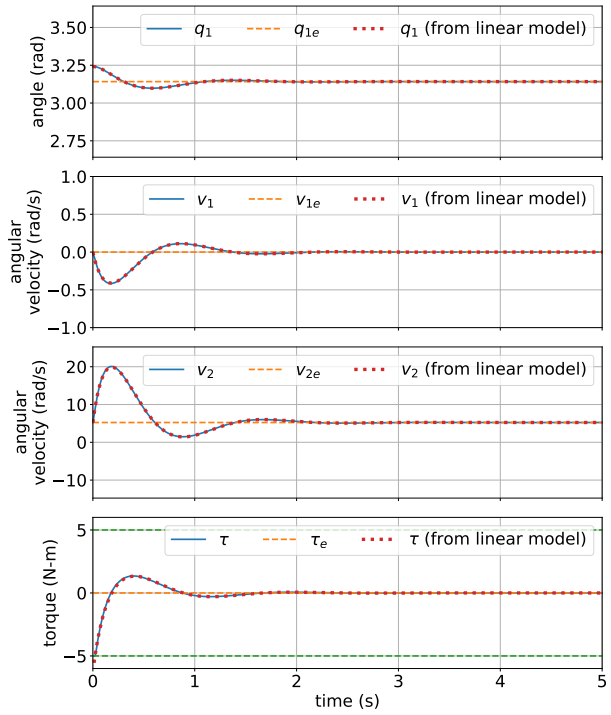
Figure 1a shows the results of our experiment to find peak time and peak overshoot (Section IV.B), starting from the initial conditions (22). We note that the closed-loop response is stable in this case—the platform angle, platform angular velocity, and wheel velocity all converge to equilibrium values. The platform angle $q_1(t)$ exhibits a clear peak from which we computed the following peak time and peak overshoot:

$$T_p = 0.58 \text{ seconds} \qquad M_p = 0.44. \tag{26}$$

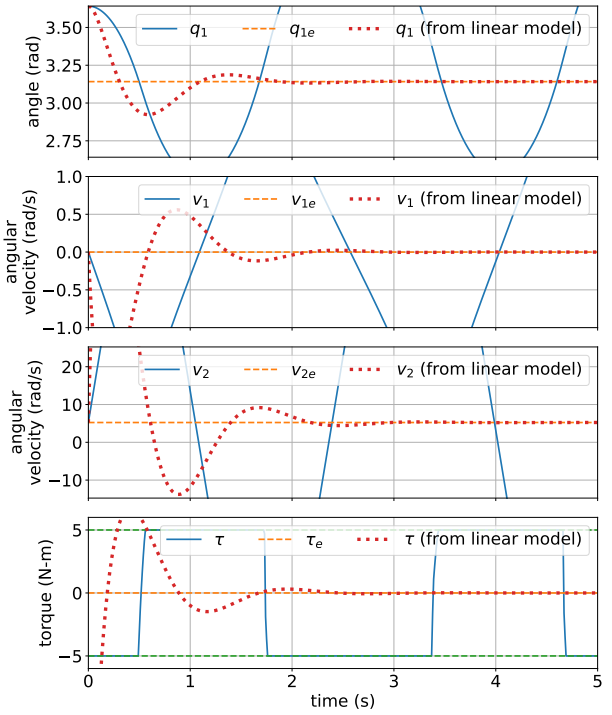
Comparing these quantities to their target values (16), we see that our design produced a peak time that was nearly 30% smaller than intended and a peak overshoot that was about 120% larger than intended. To rule out the possibility that errors in the linear model (11) caused this discrepancy, we also show in Figure 1a the trajectory

$$x(t) = e^{(A-BK)t} x(0) \tag{27}$$

that was predicted by this linear model. There appears to be negligible difference between what was derived from simulation and from this linear model. So, we conclude that the likely reason for not having achieved target values was the difference between our third-order linear system (11) and the canonical second-order system from which the



(a) If the initial error in platform angle is 0.1 rad and the initial error in both platform angular velocity and wheel angular velocity is zero, then the closed-loop response in simulation is *stable* (i.e., states *do* converge to equilibrium within four seconds).



(b) If the initial error in platform angle is 0.5 rad and the initial error in both platform angular velocity and wheel angular velocity is zero, then the closed-loop response in simulation is *unstable* (i.e., states do *not* converge to equilibrium within four seconds).

Fig. 1 The closed-loop response in simulation can be stable or unstable depending on initial conditions.

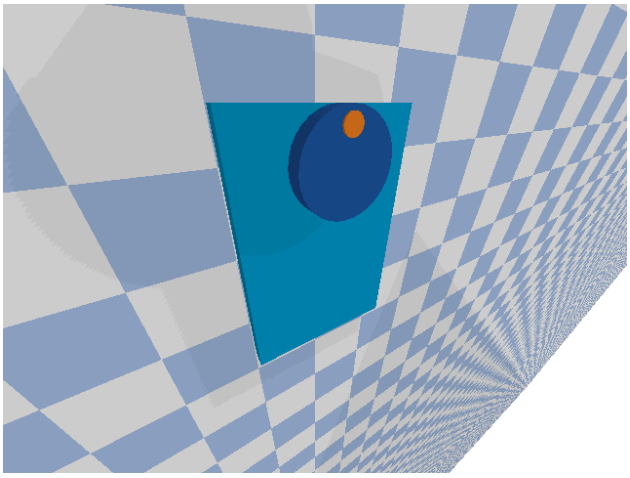


Fig. 2 The goal is to stabilize a platform (light blue) and a reaction wheel (dark blue) at an inverted configuration in gravity.

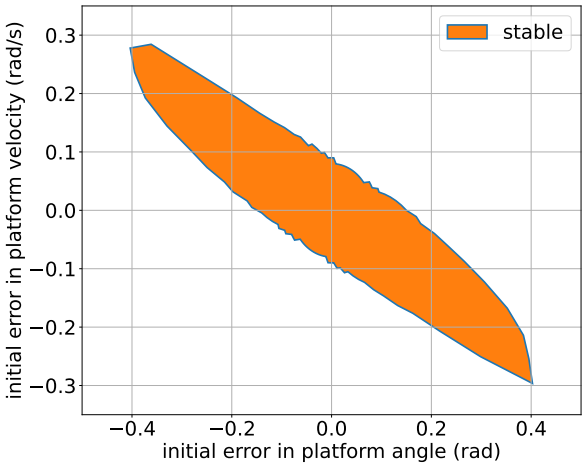


Fig. 3 The *basin of attraction* is the set of initial conditions (with zero initial error in wheel angular velocity) that produced stable closed-loop responses in simulation (i.e., for which all states converged to equilibrium within four seconds).

heuristic (17) was derived. It is also possible that we failed to place the third eigenvalue s_3 in (18) far enough into the left-half plane, producing a non-negligible effect on the closed-loop response.

Figure 3 shows the results of our experiments to find the basin of attraction (Section IV.C). Again, all points inside the basin of attraction correspond to initial conditions that produced trajectories—like the one shown in Figure 1a and *not* like the one shown in Figure 1b—satisfying the conditions (24) for convergence to equilibrium. As expected, the basin of attraction contains the equilibrium point (i.e., the set of initial conditions that correspond to zero initial error in q_1, v_1, and v_2) but is *not* of infinite extent—initial conditions too far from equilibrium produce trajectories that do not converge. Although Fig. 3 seems to imply that the basin of attraction has a slightly uneven or noisy boundary, we suspect that this result is an artifact of the way we searched for points on the boundary (Section IV.C) and that the boundary would have appeared smoothed if we had increased the resolution of our search. Figure 3 also seems to imply that the basin of attraction is *star-shaped*—that is, if

$$q_1(0) = q_{1e} + \alpha \qquad v_1(0) = v_{1e} + \beta \qquad v_2(0) = v_{2e} \tag{28}$$

is in the basin of attraction, then so is

$$q_1(0) = q_{1e} + c\alpha \qquad v_1(0) = v_{1e} + c\beta \qquad v_2(0) = v_{2e} \tag{29}$$

for all $c \in [0, 1]$. This conclusion, however, is likely false—our method of finding points on the boundary is *only* capable of producing a star-shaped inner approximation to the basin of attraction.

VI. Conclusion

This report described our approach to design and test a controller that stabilizes a rotating platform with a reaction wheel in gravity at an inverted configuration. Our first key result was to show that we achieved a 30% smaller peak time and 120% larger peak overshoot in simulation than were predicted in theory. Noting the close match between trajectories predicted by our linear model and trajectories produced by simulation, future work might focus on tuning the control design prior to implementation so that the target peak time and peak overshoot are achieved more closely. Our second key result was to show that the closed-loop system was stable in simulation only when initial errors in platform angle and platform angular velocity were small, and to produce a star-shaped inner approximation to the basin of attraction. Future work might focus on obtaining a less conservative approximation to this basin of attraction.

Appendix

A review of the control system from the perspective of stakeholders (for example) might appear here.

Acknowledgments

We thank the many students in the Department of Aerospace Engineering at the University of Illinois at Urbana-Champaign who have completed design projects like the one described in this report over the past decade.

References

[1] Markley, F. L., and Crassidis, J. L., *Fundamentals of Spacecraft Attitude Determination and Control*, Springer, New York, NY, 2014.

[2] Astrom, K., and Murray, R., *Feedback systems: an introduction for scientists and engineers*, 2nd ed., Princeton University Press, 2021.

[3] Coumans, E., and Bai, Y., "PyBullet, a Python module for physics simulation for games, robotics and machine learning," `http://pybullet.org`, 2016–2021.

[4] Bretl, T., "AE353 (Spring, 2022) Code," `https://github.com/tbretl/ae353-sp22`, 2022.