Nonlinear control and stability analysis for a reaction wheel pendulum

Juan Salazar
Dept. of Electrical Engineering and Computer Science
MIT
Cambridge, United States
salazarj@mit.edu

***Abstract*—Reaction wheel mechanisms play a significant role in the fuel-limited attitude control of spacecraft and show promise in the development of re-configurable robotic systems. Moreover, exploring nonlinear control techniques for the reaction-driven pendulum can provide a substantial amount of insight for such mechanisms despite it being a model control problem. In addition, transitioning from deriving energy shaping controllers by hand to using computational techniques to derive them teaches much about the power and structure of optimization. To this end, in this report I discuss my implementation of an upright balancing controller for a simulated reaction wheel pendulum and a complementary approach for a stability analysis. Using traditional nonlinear control techniques, I derive by hand and test an energy-shaping controller to drive the pendulum into its homoclinic orbit, then implement a suitable LQR controller to stabilize it around the equilibrium point. I follow this up with an approach for estimating the region of attraction for the LQR-controlled closed-loop system as well an approach for computing a candidate Lyapunov function. While the results in my stability analysis do not actually quantify the region of attraction, the process helped me solidify many of the concepts I learned in the course.**

***Index Terms*—reaction wheel, energy shaping, Lyapunov, nonlinear control**

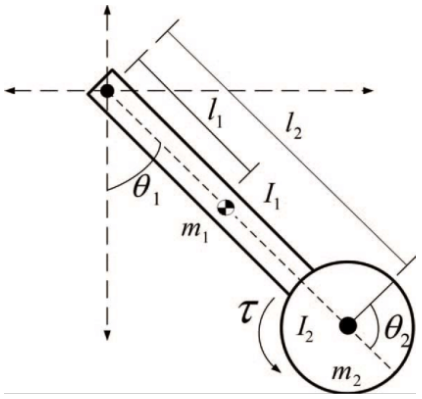


Fig. 1. Diagram of 1-D reaction wheel pendulum. θ_1 is the angle between the link and the vertical axis from the downright, θ_2 is the angle of the wheel from the link axis, m_1 is the wheel mass, m_2 is the link mass, I_1 is the moment of inertia of the link about its COM, I_2 is the moment of inertia of the wheel about its COM, l_1 is the distance from the pivot to the link COM, and l_2 is the distance from the pivot to the wheel COM.

I. Introduction

The reaction wheel pendulum is a model nonlinear control problem that has seen a diverse set of variations, several of which use nonlinear analysis and control techniques. In this report, I will demonstrate my simulated implementation of a balancing controller that consists of 1) an energy shaping controller that drives the pendulum to its homoclinic orbit and 2) a Linear Quadratic Regulator (LQR) controller that stabilizes the pendulum when it enters the region of attraction (ROA). Using the resulting LQR controller, I will describe how I computed two candidate Lyapunov functions for the closed-loop system. Additionally, I will describe my steps leading up to an optimization-based formulation of a Lyapunov function for the closed-loop pendulum along with an estimate of its ROA. Although I was not able to draw reasonable ROA estimates from the optimization, I will discuss my reasoning for its potential faults as well as potential extensions to the formulation when I can make it work.

A. Related Work

In application, reaction wheels are present in the control of spacecraft with the goal of reducing thrust requirements as well as in the control of robots that need to balance. One particular robot that has demonstrated the promise of reaction wheels is the Cubli, a cubical robot fitted with three independent reaction wheels that enable it to swing up and balance on one of its vertices in part by using pre-derived energy shaping controllers via Lyapunov analysis [2]. With this capability, Cubli can potentially serve as a unit within a homogeneous modular swarm that can reconfigure into various structures, an idea that is reminiscent of the re-configurable reaction wheel-driven M-Blocks developed at MIT's Computer Science and Artificial Intelligence Lab [3]. In light of this, I think this model control problem offers a great toolbox for the development of modular robotics.

More closely related to this project is the research conducted by Mark Spong on his own reaction wheel pendula [1]. While his formulation of the energy-shaping control problem used traditional nonlinear control techniques, it serves as a great source of inspiration and intuition for the design and improvement of my own controller.

II. Control approach

A. Dynamics

The equations of motion for the undamped reaction wheel pendulum were drawn from the Lagrange Equations, where the Lagrangian L is defined as the kinetic energy minus the potential energy of the system. The expressions for these energies are determined by referring to the configuration in Fig. 1.

$$L = T - U \tag{1}$$

The resulting equations of motion are derived using the Lagrange Equations, where q_i and $\dot{q}_i$ are respectively the generalized coordinates and the generalized time derivatives, and F_i are the generalized forces.

$$\frac{d}{dt}(\frac{\partial L}{\partial \dot{q}_i}) - \frac{\partial L}{\partial q_i} = \sum_{i=0}^{n} F_i \tag{2}$$

To avoid the chance of error, I simply decided to reference the equations of motion laid out by Spong, which can be packaged into the corresponding manipulator equations [1], where u is the torque input from the reaction wheel.

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} = \tau_g(q) + Bu \tag{3}$$

$$M = \begin{bmatrix} m_1 l_1^2 + m_2 l_2^2 + I_1 + I_2 & I_2 \\ I_2 & I_2 \end{bmatrix} \tag{4}$$

$$C = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{5}$$

$$\tau_g(q) = \begin{bmatrix} -(m_1 l_1 + m_2 l_2) g sin(\theta_1) \\ 0 \end{bmatrix} \tag{6}$$

$$B = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{7}$$

B. Energy shaping

The principle of energy shaping bridges the control of a dynamic system with its total energy. In the case of the reaction wheel, the input torque u adds energy into the system, increasing the kinetic energy the moment it is introduced with the goal of reaching the total energy required by the target state. Although the solution for the input torque for the upright balancing problem is limited by torque-limits and damping within the system, using energy feedback we can formulate a solution that guarantees the pendulum will reach the target energy-level for a certain set of initial conditions. In my implementation, I decided to not consider damping factors, but did set a torque-limit albeit a generous one.

To begin, we need to express the total energy in the system or the sum of the kinetic and potential energies.

$$E = T - U \tag{8}$$

For the reaction wheel pendulum defined in our problem, the expressions for the kinetic and potential energies are dependent on the configuration velocities and positions, where I_w is the moment of inertia of the reaction wheel about the pendulum pivot, and I_L is the moment of inertia of the pendulum link about the pivot, and $\bar{m}$ is the mass-distance sum for the wheel and the link.

$$T = \frac{1}{2} I_w (\dot{\theta}_1 + \dot{\theta}_2)^2 + \frac{1}{2} I_L \dot{\theta}_1^{\,2} \tag{9}$$

$$U = \bar{m} g \cos\theta_1 \tag{10}$$

$$\bar{m} = m_1 l_1 + m_2 l_2 \tag{11}$$

Next, we define the target energy E_{target} at the upright position and the energy error E_{error}.

$$E_{target} = -\bar{m} g \cos\pi = \bar{m} g \tag{12}$$

$$E_{error} = E - E_{target} \tag{13}$$

By substituting in the kinetic and potential energies then computing the time derivative we get an expression for the energy error derivative, which serves as our preliminary tool for guaranteeing convergence to zero.

$$\begin{gathered} \dot{E}_{error} = \frac{d}{dt}[E - E_{target}] = I_2 \dot{\theta}_1 \ddot{\theta}_1 + \\ I_w(\ddot{\theta}_1 \dot{\theta}_2 + \dot{\theta}_1 \ddot{\theta}_2) + I_2 \dot{\theta}_2 \ddot{\theta}_2 + \bar{m} g \sin\theta_1 \dot{\theta}_1 \end{gathered} \tag{14}$$

Before proceeding with energy, we can use non-collocated partial feedback linearization (PFL) to rewrite the dynamics such that the unactuated joint θ_1 can be controlled with arbitrary torques and that the actuated joint θ_2 is indirectly controlled by these torques. To do this, we can solve (3) for a torque input τ that linearizes the system in θ_1. Note, in the bottom expression I_{total} is equal to the sum of I_w and I_L.

$$\tau = I_2 u + I_2 \ddot{\theta}_2 \tag{15}$$

$$\ddot{\theta}_1 = u \tag{16}$$

$$\ddot{\theta}_2 = \frac{-\bar{m} g \sin\theta_1 - I_{total} u}{I_2} \tag{17}$$

After substituting (17) into (14) and performing more algebra, I arrived at the final expression for the energy error derivative in terms of u.

$$\dot{E}_{error} = \ddot{\theta}_2 [(I_2 - I_{total})u - \bar{m} g \sin\theta_1] \tag{18}$$

To guarantee the derivative is negative (for points where $\dot{\theta}_2$ is not zero), we can set u such that the expression becomes a strictly negative sum of squares, where k is positive constant gain.

$$u = k\dot{\theta_2} + \frac{\bar{m} g \sin\theta_1}{I_2 - I_{total}} \tag{19}$$

$$\dot{E}_{error} = -k(I_{total} - I_2)\dot{\theta_2}^2 \tag{20}$$

As I will demonstrate in a later section, this derivation comes with problems. In fact, it does not actually render the homoclinic orbit attractive, but nevertheless does continually pump energy into the system such that an LQR controller can be switched on near the upright position to stabilize the pendulum.

C. LQR

To complement the energy shaping controller, an LQR controller is activated when the pendulum is near the upright position. Specifically, when the pendulum's state enters the ROA around the upright equilibrium point, the LQR controller takes over control and stabilizes the system. This means the policy is computed before control is even initiated, according to the continuous-time LQR objective.

$$J = \int_0^\infty \left(x(\tau)^T Q x(\tau) + u(\tau)^T R u(\tau) \right) \, d\tau \tag{21}$$

My formulation of the LQR problem is relatively straightforward and is composed of two parts: 1) linearizing the pendulum dynamics about the upright equilibrium point to get the linear system matrices A_{lin} and B_{lin} and 2) solving the quadratic-cost optimization in (21) using the Algebraic Ricatti Equation (ARE) given the linearized dynamics as well as Q and R matrices. Solving the ARE generates a gain matrix K, which expresses the full-state feedback gain for the controller, and S, which is the cost-to-go coefficient matrix (further detail provided later).

$$x = [\theta_1, \theta_2, \dot{\theta_1}, \dot{\theta_2}]^T \tag{22}$$

$$\dot{x} = A_{lin} x + B_{lin} u \tag{23}$$

$$u = -Kx \tag{24}$$

The upright equilibrium point is defined as $x = [\pi, 0, 0, 0]^T$ according to the diagram in Fig. 1. For the Q and R matrices, I selected two simple yet commonly used configurations as a starting point.

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{25}$$

$$R = [1] \tag{26}$$

III. Stability Analysis

In this section I describe my approach to characterizing the stability of the closed-loop pendulum via Lyapunov functions. As I mentioned earlier, the energy shaping controller drives the pendulum to the homoclinic orbit, wherein once it reaches a ROA near the upright fixed point the LQR controller takes over and stabilizes the system. The aim of my stability analysis is first to quantify this ROA by formulating an optimization routine. This optimization routine, which we will address later as the S-procedure, tries to find the maximum bounded region G (around the upright equilibrium point) in which a Lyapunov function $V \succ 0$ for the system satisfies the following conditions everywhere in G, where ρ is the level-set that defines the region.

$$G : \{x | V(x) \le \rho\} \tag{27}$$

$$\dot{V}(x) \prec 0, \forall x \in G \tag{28}$$

Next in my analysis, I plan an approach to compute a candidate Lyapunov function using a model polynomial function generated from a predefined basis of monomials. Similar to ROA estimation, the idea is to rely on the S-procedure to search for the coefficients of this polynomial by solving the following optimization problem.

$$\begin{aligned} & \underset{\alpha}{\text{find}} \\ & \text{subject to} \quad V \text{is SOS}, -\dot{V} \text{is SOS} \end{aligned} \tag{29}$$

Although I was not able to quantify a ROA or to compute a candidate Lyapunov via optimization for the closed-loop pendulum, I intend to demonstrate what I learned in the process of building the optimization infrastructure.

A. ROA Estimation using the S-Procedure

1) Computing a Lyapunov Function from the Linearized Model: To estimate the ROA of a regionally stable system, one must present a valid Lyapunov function. In particular, this function must satisfy the constraints (27) and (28). For a nonlinear system, as it turns out, the cost-to-go matrix generated from solving the LQR problem in (21) is enough to come up with a good candidate Lyapunov function. Comparing the LQR problem, a spin-off of the Hamilton-Jacobi-Bellman equations from dynamic programming, to finding a candidate Lyapunov one can realize that the latter is a relaxation of the primer [4]. This implies that the cost-to-go function $J^*_{lqr}(x)$ is a good Lyapunov candidate.

$$V_{lqr}(x) = J^*_{lqr}(x) = x^T S x \tag{30}$$

$$\dot{V}_{lqr}(x) = 2x^T S x \tag{31}$$

In addition to computing V_{lqr}, I also used the linearized form of the pendulum dynamics to compute $V_{lyapunov}$, the candidate function from solving the Lyapunov Equation for a given matrix $Q_{lyapunov}$. By substituting (24) into (23), we get the linearized closed-loop dynamics, where the matrix $A_{cl} = A_{lin} - B_{lin}K$. Afterwards, we substitute this matrix into the Lyapunov Equation to solve for P, the coefficient matrix of a new candidate Lyapunov function $V_{lyapunov}$.

$$\dot{x} = A_{cl}x \tag{32}$$

$$A_{cl}^T P + P A_{cl} = -Q_{lyapunov} \tag{33}$$

$$V_{lyapunov} = x^T P x \tag{34}$$

Although this is indeed a valid candidate Lyapunov function (from the requirements of the Lyapunov Equation), I proceeded with the LQR-derived candidate for the estimation of the ROA.

2) S-Procedure Formulation: Moving forward, the S-procedure is an optimization formulation technique with which we can write the Lyapunov conditions (27) and (28). In general, we formulate the problem this way in order to demonstrate that a scalar polynomial $p(x)$ is greater than or equal to zero everywhere in $g(x)$, where λ is a polynomial and $g(x) \prec 0$ is a semi-algebraic set [4]. In other terms, if p(x) satisfies the following constraint for a certain λ, it is sufficient to state that $p(x)$ is positive semidefinite in $g(x)$.

$$p(x) + \lambda(x)g(x) \text{ is SOS}, \lambda(x) \text{ is SOS}. \tag{35}$$

Next, we will specify this formulation to the Lyapunov conditions by defining $g(x) = V(x) - \rho$ and $p(x) = -\dot{V}$. Recall that ρ a level-set of $V(x)$, which means that if we find both a ρ and λ such that the constraint is satisfied, we have certified the Lyapunov function $V(x)$ in $g(x) = V(x) - \rho \prec 0$ and determined that the level-set ρ is inside the ROA.

$$-\dot{V} + \lambda(x)(V(x) - \rho) \text{ is SOS}, \lambda(x) \text{ is SOS}. \tag{36}$$

Finally, we need to refactor this in order to maximize the level-set ρ, which defines the outer boundary of the ROA. The new formulation, which I ended up using, enforces that $V(x) - \rho \geq 0$ when $\dot{V} \geq 0$, even when $x = 0$. Conveniently, the formulation is also convex in ρ and λ [4].

$$\begin{aligned} &\max_{\rho, \lambda(x)} && \rho \\ &\text{subject to} && (x^T x)(V(x) - \rho) + \lambda(x)\dot{V} \text{ is SOS}, \lambda(x) \text{ is SOS}. \end{aligned}$$

At last, now that we have the template formulation for the S-procedure we can bring back V_{lqr} defined in (30) and estimate the ROA in a script. By substituting (24) into (3), then reducing the order of the model by removing the θ_2 state (we do not care about the angular position of the wheel), removing the corresponding rows and columns from S and K, we have a closed-loop dynamic model for the S-procedure to work with. Note that because the optimizer I used does not support functions like $\sin\theta_1$ in the dynamics I used a Taylor polynomial approximation for the the gravitational torque term.

Simulation Parameters

Parameter	Value
m_1	1
m_2	1
l_1	0.5
l_2	1
I_1	0.1
I_2	0.2
$\lvert u_{max} \rvert$	10

Fig. 2. Table of parameters used in simulation.

B. Computing a Lyapunov Function using the S-Procedure

My approach for estimating a Lyapunov function of the damped version of an open-loop pendulum from the optimization routine begins with changing the state coordinates in the pendulum dynamics from $x = [\theta_1, \theta_2, \dot{\theta_1}, \dot{\theta_2}]^T$ to $x = [\sin\theta_1, \cos\theta_2, \dot{\theta_1}, \dot{\theta_2}]^T$, a change that renders the trigonometric terms $\sin\theta_1$ in the dynamics into polynomials with the additional constraint that $(\sin\theta_1)^2 + (\cos\theta_1)^2 = 1$. I then parameterize a polynomial candidate $V(\sin\theta_1, \cos\theta_1, \dot{\theta}_1)$, which has unknown coefficients α_i that act as decision variables in the optimization. Using the S-procedure, we take the template in (35), assign $p(x) = -\dot{V}$ and $g(x) = \sin\theta_1)^2 + (\cos\theta_1)^2 - 1$, and add an SOS constraint on V. We choose $g(x)$ this way in order to enforce the constraint that $p(x) = -\dot{V}$ is SOS in the domain of states we care about; on the unit circle [4].

$$\begin{aligned} &\underset{\rho, \lambda(x)}{\text{find}} \\ &\text{s.t.} - \dot{V} - \lambda(x)(\sin\theta_1)^2 + (\cos\theta_1)^2 - 1) \text{ is SOS}, V \text{ is SOS}. \end{aligned}$$

IV. Results

All simulation and optimization routines were written using Drake's Python toolbox. Below I will demonstrate the results from running the two working controllers in simulation and from my stability analysis. The pendulum I built in the simulation used the parameters tabulated in Fig. 2. Hyperlinks for videos of the example simulations discussed are included at the end of this section.

A. Results from Energy-shaping

With respect to energy-shaping, two controllers were implemented and tested in simulation. The first was my hand-derived solution (19), whereas the second was borrowed from a similar research project that used passivity control concepts to derive it [1]. I selected $k = 0.1$ in (19) and $k_e = 500, k_v = 1$ in Spong's solution.

$$u_{Spong} = I_2 k_e E_{error}\dot{\theta}_1 - k_v\dot{\theta}_2 \tag{37}$$

The results from both controllers are presented in Fig.3 and 4 in the form of phase plots. By inspection, the controller stated in (19) corresponding to Fig. 3 does not converge on the desired homoclinic orbit and instead continually increases the total energy and diverges (it can reach the upright state with excessive angular velocity $\dot{\theta}_1$). Spong's controller results

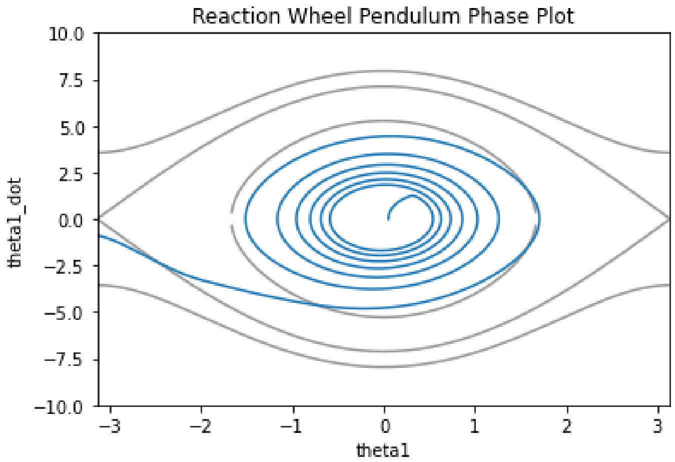


Fig. 3. Resulting swing-up phase plot of $[\theta_1, \dot{\theta}_1]$ from a simulation of the pendulum using my hand-derived controller. The initial condition was set to $x = [0.5, 0, 0, 0]^T$ to avoid remaining fixed at $x = [0.0, 0, 0, 0]^T$ due to the dependence of (19) on $\dot{\theta}_2$

.

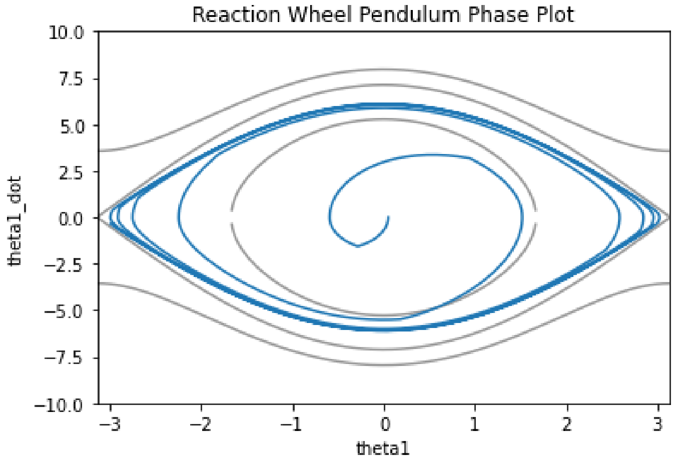
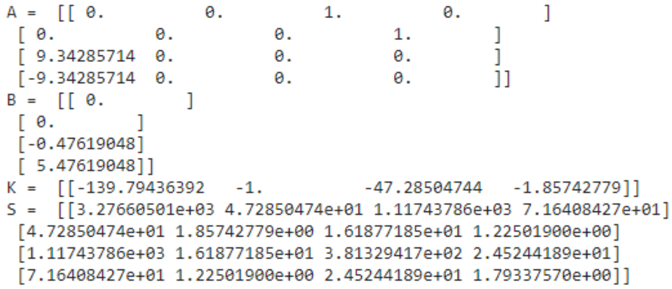


Fig. 4. Resulting swing-up phase plot of $[\theta_1, \dot{\theta}_1]$ from a simulation of the pendulum the controller from (37). The initial condition was set to $x = [0.5, 0, 0, 0]^T$ to avoid remaining fixed at $x = [0.0, 0, 0, 0]^T$ due to the dependence of this controller on $\dot{\theta}_2$.

in a phase plot that visually converges to a particular constant-energy orbit that is reflected in Fig. 4. In the simulated visualization, both controllers have similar swing-up behaviors but ultimately only the controller in (37) renders the homoclinic orbit attractive.

B. Results from Energy-shaping + LQR

First, we can briefly evaluate the performance of the LQR controller on its own when the pendulum is initialized close to the equilibrium point. With the formulation described within equations (21) through (26), the controller successfully stabilizes the pendulum when θ_1 lies within an approximate deviation of $\pi/6$ around the equilibrium point; this was determined by simply running the controller with different various initial conditions. Linearizing the system dynamics generated matrices A_{lin} and B_{lin} while the LQR controller produced the controller gain K and cost-to-go matrix S, all of which are presented in Fig. 5. Video clips for the LQR-only simulation are included with the rest of the videos.



```
A =  [[ 0.          0.          1.          0.        ]
 [ 0.          0.          0.          1.        ]
 [ 9.34285714  0.          0.          0.        ]
 [-9.34285714  0.          0.          0.        ]]
B =  [[ 0.        ]
 [ 0.        ]
 [-0.47619048]
 [ 5.47619048]]
K =  [[-139.79436392   -1.          -47.28504744   -1.85742779]]
S =  [[3.27660501e+03 4.72850474e+01 1.11743786e+03 7.16408427e+01]
 [4.72850474e+01 1.85742779e+00 1.61877185e+01 1.22501900e+00]
 [1.11743786e+03 1.61877185e+01 3.81329417e+02 2.45244189e+01]
 [7.16408427e+01 1.22501900e+00 2.45244189e+01 1.79337570e+00]]
```

Fig. 5. Output matrices from the system linearization and LQR.

.

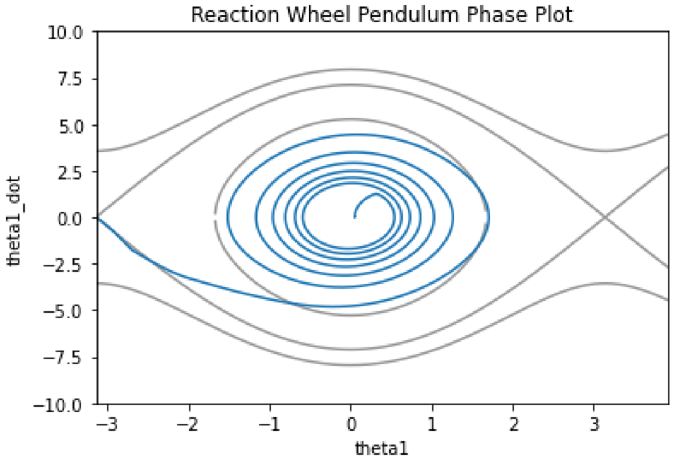


Fig. 6. Resulting phase plot of the my swing-up + LQR balancing controller.

.

The results from both energy-shaping controller options are presented in the form of phase plots in Fig. 6 and 7. The phase plot for the swing-up/LQR combination using the energy controller in (19) converges on the equilibrium point. The phase plot for the swing-up/LQR combination using the energy controller in (37) also converges on the equilibrium point. One can clearly see the difference in phase trajectories between the two, as the plot in Fig. 6 illustrates a trajectory that appears to slowly converge on the homoclinic orbit but would not do so if an LQR controller was not activated near the ROA. Fig. 7 presents a more desirable phase trajectory for a swing-up controller, since the homoclinic orbit is clearly traced and converges faster than in the scenario with controller (19) in Fig. 6. Video clips for these simulations are also provided below.

C. Results from Stability Analysis

With respect to the S-procedure formulation for ROA estimation, the candidate Lyapunov function was defined using the LQR cost-to-go matrix, presented alongside the P matrix

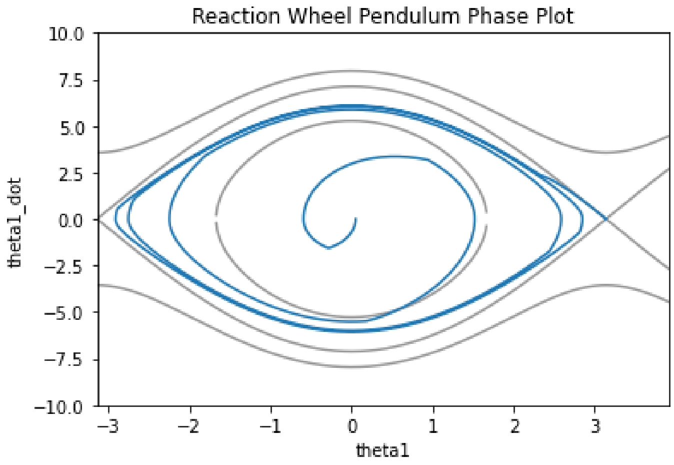
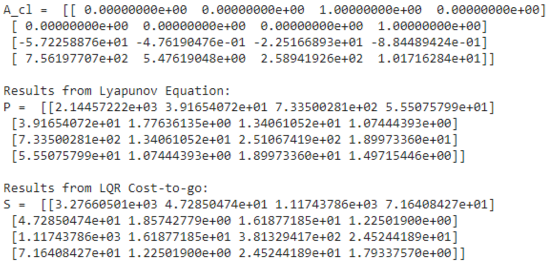


Fig. 7. Resulting phase plot of Spong's swing-up + LQR balancing controller.

.



```
A_cl =  [[ 0.00000000e+00  0.00000000e+00  1.00000000e+00  0.00000000e+00]
 [ 0.00000000e+00  0.00000000e+00  0.00000000e+00  1.00000000e+00]
 [-5.72258876e+01 -4.76190476e-01 -2.25166893e+01 -8.84489424e-01]
 [ 7.56197707e+02  5.47619048e+00  2.58941926e+02  1.01716284e+01]]

Results from Lyapunov Equation:
P =  [[2.14457222e+03 3.91654072e+01 7.33500281e+02 5.55075799e+01]
 [3.91654072e+01 1.77636135e+00 1.34061052e+01 1.07444393e+00]
 [7.33500281e+02 1.34061052e+01 2.51067419e+02 1.89973360e+01]
 [5.55075799e+01 1.07444393e+00 1.89973360e+01 1.49715446e+00]]

Results from LQR Cost-to-go:
S =  [[3.27660501e+03 4.72850474e+01 1.11743786e+03 7.16408427e+01]
 [4.72850474e+01 1.85742779e+00 1.61877185e+01 1.22501900e+00]
 [1.11743786e+03 1.61877185e+01 3.81329417e+02 2.45244189e+01]
 [7.16408427e+01 1.22501900e+00 2.45244189e+01 1.79337570e+00]]
```

Fig. 8. Resulting cost-to-go matrix from LQR and P matrix from the Lyapunov Equation.

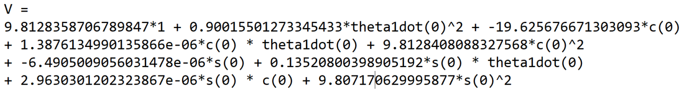
.

from solving the closed-loop Lyapunov Equation in Fig. 8. The optimization routine has returned no reasonable estimate of the ROA for the closed-loop system using the LQR controller. Fig. 9 presents the result from using the S-procedure to compute a Lyapunov function of the pendulum open-loop dynamics. The expression calculated closely resembles the total energy of the free reaction wheel pendulum, which is similar to the traditional simple pendulum.

D. Videos

Below are clickable hyperlinks to the videos mentioned above.

1) Simulation of hand-derived swing-up controller in (19): https://youtu.be/xslzLpM8my4
2) Simulation of Spong's swing-up controller in (37): https://youtu.be/kVTJ-E65aoM



```
V =
9.8128358706789847*1 + 0.90015501273345433*theta1dot(0)^2 + -19.625676671303093*c(0)
+ 1.3876134990135866e-06*c(0) * theta1dot(0) + 9.8128408088327568*c(0)^2
+ -6.4905009056031478e-06*s(0) + 0.13520800398905192*s(0) * theta1dot(0)
+ 2.9630301202323867e-06*s(0) * c(0) + 9.8071706299958777*s(0)^2
```

Fig. 9. Result from S-procedure ROA estimation and Lyapunov function estimation.

.

3) Simulation of LQR controller starting far away from equilibrium: https://youtu.be/hQlQcK6aVQQ
4) Simulation of LQR controller starting near equilibrium: https://youtu.be/RNMYdTGTv6c
5) Simulation of hand-derived swing-up and LQR: https://youtu.be/6dF-pg5ov0I
6) Simulation of Spong's swing-up and LQR: https://youtu.be/oz5cp_WY7t0

E. Code

This is a clickable hyperlink to the GitHub repo where the code is stored (in Colab notebook): https://github.com/juansala/underactuated_project.git

V. Discussion

Overall, the aim of this project was to design an upright balancing controller for a reaction wheel pendulum and to complement the control with a stability analysis. Despite the issues with the energy-shaping controller I derived by hand, the simulation still demonstrated an upright stable system; the energy error did not converge to zero, but the swing-up behavior was still captured. To verify that there was not another factor at play, I tested the same pendulum simulation using a swing-up controller derived in previous work via passivity control and zero dynamics [1] and was able to plot the homoclinic orbit. In both cases, the LQR controller was able to stabilize about the equilibrium. In hindsight, to improve my controller I need to control the velocity of the wheel $\dot{\theta}_2$ so as not allow it a large contribute in the total kinetic energy. This is because in my formulation a valid energy policy is to spin-up the wheel to satisfy the target energy, neglecting the actual required potential energy at the upright position. I could potentially achieve this by adding a restraint on the wheel velocity that is subtracted from the control input at times when it is excessive. Alternatively, I could consider formulating the total kinetic energy in such a way that it ignores the wheel velocity to avoid excess spin-up in the first place.

While my stability analysis was cut short due to time constraints, I was able to lay out an approach for estimating the ROA of the closed-loop undamped reaction wheel pendulum using the S-procedure. A number of factors could be affecting the result of the optimization, including but not limited to numerical issues, inadequate approximation of the gravitational torque, or issues stemming from the LQR output matrices K and S. Regardless, until this component of my stability analysis can output reasonable estimates, an alternative method is to take a large sample of points from the candidate Lyapunov function $V_{lqr}(x)$ and iteratively searching the sample set for points that fail to meet the (27) and (28). This could serve well as a sanity check for the output of the S-procedure.

Although the formulation for computing Lyapunov functions is useful for verifying global stability, which is inherent in the damped pendulum, I would need to adjust it in order to search for a closed-loop Lyapunov function that is limited to regional stability (such as in the LQR case). This requires enforcing the conditions for a level-set bounded region in (27) and (28) while also accounting for the change in coordinates. Such a Lyapunov function would be a useful tool for verifying the stability of the LQR control near the equilibrium, and if the conditions in the optimization were tweaked further to guarantee a stable phase trajectory it could even inform new energy-shaping controllers, as I had originally planned to investigate.

Additionally, I would like to extend this work to a 1-D version of the Cubli. While the pendulum model is canonical in this field, the form factor of the Cubli lends better to modular robotics applications. I believe that upgrading the model to represent even more non-trivial inertial couplings is a reasonable step to take towards developing novel reconfigurable robots.

References

[1] M. Muehlebach, G. Mohanarajah and R. D'Andrea, "Nonlinear analysis and control of a reaction wheel-based 3D inverted pendulum," 52nd IEEE Conference on Decision and Control, 2013, pp. 1283-1288, doi: 10.1109/CDC.2013.6760059.

[2] Mark W. Spong, Peter Corke, Rogelio Lozan, "Nonlinear control of the Reaction Wheel Pendulum," Automatica, Volume 37, Issue 11, 2001, pp. 1845-1851, doi: 10.1016/S0005-1098(01)00145-5.

[3] J. W. Romanishin, K. Gilpin, S. Claici and D. Rus, "3D M-Blocks: Self-reconfiguring robots capable of locomotion via pivoting in three dimensions," 2015 IEEE International Conference on Robotics and Automation (ICRA), 2015, pp. 1925-1932, doi: 10.1109/ICRA.2015.7139450.

[4] Russ Tedrake. Underactuated Robotics: Algorithms for Walking, Running, Swimming, Flying, and Manipulation (Course Notes for MIT 6.832). Downloaded on [date] from http://underactuated.mit.edu/