Swingbot: underactuated acrobot robot design and collocated partial feedback linearization control

CHARLIE YAN

Human beings on a swing set move their body to 'pump' themselves to achieve high amplitude swing trajectories. These strategies involve either shifting the effective center-of-mass or introducing angular-impulse torque on the system [1] . We aim to design a swinging robot and controller that mimics the various human swing pumping strategies as well as optimizing the timing and strategy used to maximize amplitude gain and minimize actuator output. We explore a subset of the parameter space to see how various robot designs and controllers perform. Finally, present ancillary tools and key findings.

1. INTRODUCTION

The simple pendulum is one of the most well studied and understood dynamical system in physics and controls. A real life example of a damped simple pendulum most people have encountered is the swing set. The person manipulates the pendulum dynamics with their body to overcome to damping forces and achieved a desired amplitude of oscillation.

Some 'pumping' actions a person can take are described in [1] and involve 1. changing the center-of-mass and 2. rotating the body about the center of mass to impose angular impulse. For both of these methods of pumping energy into the system, it is important to point out that the angle the swing makes relative to the vertical (θ_1) is **not** a joint that is under control but also the joint position that we ultimately care about. Like humans and monkeys do in the real world, we aim to influence the trajectory of θ_1 indirectly through the actions distally, and it is the system dynamics that couples the two. Through analysis, can derive *how* the motion in one part of the system (our actuated joints) exactly influences the motion of θ_1.

2. ROBOT DESIGN

Our robot is designed to pump energy into the system through the change in angle strategy outlined in [1]. We will design 2 variations of our robot, design 1 and 2, where design 2 is has an additional motor, link, and mass to pump energy. A figure of the two designs follows a short description of each.

A. Design 1

We model the swing as a 'rigid' link of length l_1 that is at an angle θ_1 from the vertical, at the distal end of which we hang our robot via chain fastenings. Coaxial with where the chain attaches to our robot, we have a motor that we can control the position, acceleration, and velocity of with PID feedback control through PWM signal and encoder feedback. This entire apparatus is modeled as a point mass of m_1. Mechanically coupled to the output rotor of the motor is a link of length l_2, at the distal end of which is fixed a mass m_2.

The biological analog of design 1 can be considered as a trapeze artist, holding his or her body straight, swinging on a bar suspended by rope. Note that unlike [2], we model our design as a **simple** double pendulum where links have negligible mass compared to the masses suspended at their ends. This somewhat simplifies our Lagrangian derived equations of motion in the next section because we do ignore for rotational motion of the links and only consider the translational motion of the end masses. Furthermore, we model the swing set rope as a **single** rigid link and ignores any slack in it that would be perhaps more accurately modeled as a series of unactuated joints.

B. Design 2

Design 2 is based on the model described in [1] where the swinger is a single link and is attached to the swing at its center of mass. The equations of motion derived in [1] are based on the angular momentum change whereas ours will be based on the translational motion of the masses. Design 2 aims to reproduce the geometry of [1] and in turn a human sitting on a swing set in that humans on a swing set have a link on **both** ends of the swing seat, one made of legs, the other the torso. Towards that aim, design 2 is like design 1 but with the addition of another motor coaxial with the design 1 motor with an independent rotor shaft coupled to a separate mass-less link of length l_3 holding mass m_3.

3. MODELING: ASSUMPTIONS AND EQUATIONS OF MOTION AND DYNAMICS

A. Design 1

To simplify the physical system into something we can more easily study without loss of core dynamics, we make some simplifying assumptions. For design 1 and 2, we make the same kinds of assumptions.

- The person's body between the head and waist is a mass-less, rigid link

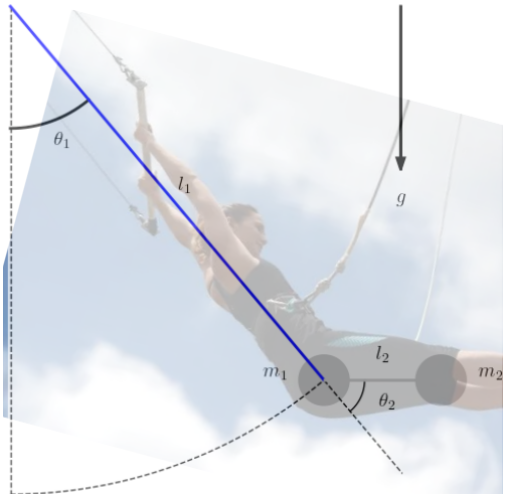

Fig. S1. Design 1

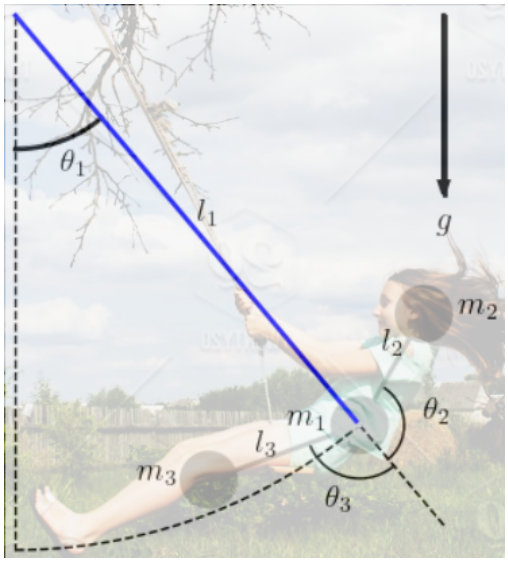

Fig. S2. Design 2

- The person's body between waist and feet is a mass-less, rigid link
- The person's arms are also mass-less, rigid. In design 1, the arms are a part of the link between waist and head, and in design 2, the arms are a part of the link from waist to swing axis.
- The swing link is mass-less, rigid, and there is 1 link for the entire physical object.
- The angle between waist and feet or head has no angle limits, nor trajectory limits
- There is no damping on the system except for with respect to the waist mass

From our free-body diagrams, we can introspect out the system translational velocity terms. We try our best to adhere to naming conventions found elsewhere in literature for clarity.

$$x_1 = l_1 \sin(\theta_1) \tag{S1}$$
$$y_1 = -l_1 \cos(\theta_1) \tag{S2}$$
$$x_2 = x_1 + l_2 * \sin(\theta_2) \tag{S3}$$
$$y_2 = y_1 - l_2 * \cos(\theta_2) \tag{S4}$$
$$v_1^2 = \dot{x_1}^2 + \dot{y_1}^2 \tag{S5}$$
$$v_2^2 = \dot{x_2}^2 + \dot{y_2}^2 \tag{S6}$$

We will use Lagrangian mechanics to derive the equations of motion for our design 1 system. From the L term in S9, we can derive the **Euler-Lagrange** equations of motion. Note that the equation of motion for our actuated joint θ_2 (S11) equates to a nonzero input torque τ, and the unactuated joint θ_1 (S10) has no input force so it's equation of motion equals 0.

$$T = \frac{m_1}{2} v_1^2 + \frac{m_2}{2} v_2^2 \tag{S7}$$
$$V = m_1 g y_1 + m_2 g y_2 \tag{S8}$$
$$L = T - V \tag{S9}$$
$$\frac{\partial L}{\partial \dot{\theta_1}} - \frac{\partial L}{\partial \theta_1} = 0 \tag{S10}$$
$$\frac{\partial L}{\partial \dot{\theta_2}} - \frac{\partial L}{\partial \theta_2} = \tau \tag{S11}$$

We use the python open-source package **sympy** to help us perform symbolic computation to derive the full expression for S10, S12 and S11, S13. Note how both equations of motion describe dynamics that are functions of the motion of the other joints in the system.

$$g m_1 \sin(\theta_1(t)) + g m_2 \sin(\theta_1(t)) + l_1 m_1 \frac{d^2}{dt^2}\theta_1(t) + l_1 m_2 \frac{d^2}{dt^2}\theta_1(t) + l_2 m_2 \sin(\theta_1(t) - \theta_2(t)) \left(\frac{d}{dt}\theta_2(t)\right)^2 + l_2 m_2 \cos(\theta_1(t) - \theta_2(t)) \frac{d^2}{dt^2}\theta_2(t) = 0 \tag{S12}$$

$$g \sin(\theta_2(t)) - l_1 \sin(\theta_1(t) - \theta_2(t)) \left(\frac{d}{dt}\theta_1(t)\right)^2 + l_1 \cos(\theta_1(t) - \theta_2(t)) \frac{d^2}{dt^2}\theta_1(t) + l_2 \frac{d^2}{dt^2}\theta_2(t) = \tau \tag{S13}$$

We also recognize the practice of decomposing these equations of motion into 3 categories: terms related to the acceleration of each joint, terms related to velocity-products, and terms related to potential energy / gravity. As outlined in [3] and [4], these are often known as the **manipulator matrices** M S14, C S15, τ_g S16 and through matrix manipulation are used to derive, among other expressions, the expressions for the system **unforced dynamics**. Note that the manipulator matrices pertain to the **unforced** system, so do not include terms such as *damping* or *input forces* (τ) which belong in the right hand side (RHS) general term Q. The full, generalized system equations of motion in this form are shown in S17. Again, we use the open-source python package **sympy** to derive these equations.

$$\begin{bmatrix} l_1 m_1 + l_1 m_2 & l_2 m_2 \cos(\theta_1(t) - \theta_2(t)) \\ l_1 \cos(\theta_1(t) - \theta_2(t)) & l_2 \end{bmatrix} \tag{S14}$$

$$\begin{bmatrix} 0 & l_2 m_2 \sin(\theta_1(t) - \theta_2(t)) \frac{d}{dt}\theta_2(t) \\ -l_1 \sin(\theta_1(t) - \theta_2(t)) \frac{d}{dt}\theta_1(t) & 0 \end{bmatrix} \tag{S15}$$

$$\begin{bmatrix} g m_1 \sin(\theta_1(t)) + g m_2 \sin(\theta_1(t)) \\ g \sin(\theta_2(t)) \end{bmatrix} \tag{S16}$$

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + \tau_g = Q \tag{S17}$$

B. The full Q term

The Q term on the RHS represents all external forces acting on our system. This includes input torque on θ_2 and *damping* on θ_1. We ignore any damping on θ_2 because it is an actuated motor under active PID control. Thus, we derive the following Q S18, again in matrix form.

$$\begin{bmatrix} -\beta_{damping} \frac{d}{dt}\theta_1(t) \\ \tau \end{bmatrix} \tag{S18}$$

C. Unforced dynamics

Now that we have all the expressions that make up S17, we can use the full expression and matrix manipulations to derive the **unforced dynamics** of the system, that is how the system behaves when only damping is applied. To do that, solve to isolate $\ddot{\theta}$ on the left hand side (LHS). The full operation is shown in S19. In the case of unforced dyanmics, we can zero the parameter for input force in Q, $\tau = 0$. The full RHS derivations equating to $\ddot{\tau}$ are given in S20 and S21.

$$\ddot{\theta} = M(\theta)^{-1}(Q - C(\theta, \dot{\theta})\dot{\theta} - \tau_g) \tag{S19}$$

$$-\frac{g m_1 \sin(\theta_1(t)) + \frac{g m_2 \sin(\theta_1(t) - 2\theta_2(t))}{2} + \frac{g m_2 \sin(\theta_1(t))}{2} + \frac{l_1 m_2 \sin(2\theta_1(t) - 2\theta_2(t)) \left(\frac{d}{dt}\theta_1(t)\right)^2}{2} + l_2 m_2 \sin(\theta_1(t) - \theta_2(t)) \left(\frac{d}{dt}\theta_2(t)\right)^2 + p_{1damping} \frac{d}{dt}\theta_1(t)}{l_1 (m_1 - m_2 \cos^2(\theta_1(t) - \theta_2(t)) + m_2)} \tag{S20}$$

$$
\begin{aligned}
&- (m_1 + m_2) \left(g \sin(\theta_2(t)) - l_1 \sin(\theta_1(t) - \theta_2(t)) \left(\frac{d}{dt}\theta_1(t) \right)^2 \right) \\
&+ \left(g m_1 \sin(\theta_1(t)) + g m_2 \sin(\theta_1(t)) + l_2 m_2 \sin(\theta_1(t) - \theta_2(t)) \left(\frac{d}{dt}\theta_2(t) \right)^2 + \beta_{damping} \frac{d}{dt}\theta_1(t) \right) \\
&\frac{\cos(\theta_1(t) - \theta_2(t))}{l_2 (m_1 - m_2 \cos^2(\theta_1(t) - \theta_2(t)) + m_2)}
\end{aligned}
\tag{S21}
$$

4. CONTROLLER DESIGN

Our system state x is a 4-dimensional vector S22. With the manipulator equations in the previous section, we can also derive expressions for what is outlined in [2] and [4] as **collocated partial feedback linearization** to try to control our system in some way.

$$
x = \begin{bmatrix} \theta_1 \\ \dot{\theta_1} \\ \theta_2 \\ \dot{\theta_2} \end{bmatrix} \tag{S22}
$$

Prior literature describes **feedback linearization** as the practice of taking a given **nonlinear system** (such as ours) of the form $\dot{x} = f(x) + b(u)$ and through a change of variable, transforming it to a **linear** system of the form $\dot{z} = Ax + Bu$. If the linearized state $z = x_{desired} - x$ and A is shown to be Hurwitz, then we can guarantee that $z(t)$ converges to 0, and thus $x_{desired}(t) = x(t)$ (we achieve tracking to some reference).

Moreover, **partial feedback linearization (PFL)** refers to the practice where we apply **feedback linearization** to a subset of the entire system state as opposed to all dimensions of the state. Which state(s) chosen to be linearized relates to the goals of the controller and how the controller operates, as well as requirements and limitations of the controller.

Prior literature [2] mentions 3 types of PFL: collocated, noncollocated, and task-space PFL. Collocated and noncollocated refer to choosing to linearize either the actuated or unactuated joints, respectively. Each of these PFL choices implies different matrix operations on our system forced equations of motion S17.

Collocated PFL, or choosing to linearize the actuated joint, we aim to solve for the other state variables in terms of the actuated joint state variables. In general and for our system, the expression for doing this is outlined in [2] and expressed below in S23 and S24. This expression lets us calculate how the unactuated joint $\dot{\theta_1}$ responds to a commanded $\ddot{\theta_2}$ trajectory.

$$
\ddot{\theta_1} = M[0,0]^{-1}(-\tau_g[0] - C(\theta, \dot{\theta})\dot{\theta} - M[0,1]\ddot{\theta_2}) \tag{S23}
$$

$$
\begin{aligned}
&- g m_1 \sin(\theta_1(t)) - g m_2 \sin(\theta_1(t)) - l_2 m_2 \sin(\theta_1(t) - \theta_2(t)) \left(\frac{d}{dt}\theta_2(t) \right)^2 \\
\ddot{\theta_1} = &\frac{- l_2 m_2 \cos(\theta_1(t) - \theta_2(t)) \frac{d^2}{dt^2}\theta_2(t) - \beta_{damping} \frac{d}{dt}\theta_1(t)}{l_1 m_1 + l_1 m_2}
\end{aligned}
\tag{S24}
$$

We can then define a reference trajectory $\theta_2^{desired}$ to do PD control on the θ_2 motor. This control law is described in [4] and the output of the PD controller will be set as $\ddot{\theta_2}$. The control law is described below in S25. Note that the control law requires state feedback on θ_2 and $\dot{\theta_2}$, but not $\ddot{\theta_2}$ which we can estimate these with an encoder on the motor rotor.

$$
\ddot{\theta_2} = k_d \dot{\theta_2} + k_p \left(\theta_2^{desired} - \theta_2 \right) \tag{S25}
$$

We need to make a decision of what *kind* of reference trajectory we want to set. For this work, we choose 3 trajectory candidates. Our intention is to see how the different references perform through the system dynamics on the unactuated joint.

Reference 1 S26 trajectory is taken from prior literature on acrobot swing-up control outlined in [3] and exposes a α parameter that sets the max amplitude of the arctan trajectory of θ_2. The arctan term is scaled by a $\frac{2}{\pi}$ so that as $\dot{\theta_1}$ grows in magnitude, the scaled term plateaus to $\alpha \frac{2}{\pi} \frac{\pi}{2} = \pm\alpha$. The integration of $\dot{\theta_1}$ into arctan means that $\|\dot{\theta_1}\| \to 0 \Rightarrow \theta_2^{desired} \to 0$, or in other words as the swing link slows down around its max amplitude, our $\theta_2^{desired}$ straights out to a straight vertical. This is a mathematical expression that corresponds to the intuition for human swinging outlined in [1].

$$
\theta_2^{desired} = \frac{2\alpha}{\pi} \arctan(\dot{\theta_1}) \tag{S26}
$$

Reference 2 S27 is a simple sinusoidal one. Unlike reference 1 S26 however, it is a function of **time**, and therefore makes the system **non-autonomous**.

$$\theta_2^{desired} = \alpha \sin(t) \tag{S27}$$

Reference 3 is based on system potential energy cascade control S29 and the control law is different than the prior two.

$$u = gm_1 l_1 \left(1 - \cos(\theta_1)\right) + gm_2 \left[l_1(1 - \cos(\theta_1)) + l_2(1 - \cos(\theta_2))\right] \tag{S28}$$

$$\ddot{\theta_2} = k_p \dot{\theta_1}(e^{desired} - u) + k_d \dot{\theta_2} \tag{S29}$$

5. SIMULATION

Our goals with the simulator was to sanity-check our system dynamics under different situations and control techniques. We wanted to both visualize and quantify data and be able to reproduce runs under different parameters.

We built our simulation with relatively simple open-source python packages including **numpy**, **scipy**, and **matplotlib**. For inter-process communication to stream data between two python programs we used the open-source multi-language library **zmq** and python bindings.

The general simulation procedure proceeds as follows, and the example of the simulation output is seen in S3.

- Use **scipy.integrate.odeint** to integrate the state given an initial condition, sampling times, and state transition $\dot{x} = f(x)$ python function.
- Create an **matplotlib.animation.Animation** that steps through the integrated states and plots the state as blue links and orange track referring to the position of m_2
- To plot the state as a vector of scalars at the same time as the system 2D plot, publish the state to **zmq** messages, subscribe and process it in another program, and use **matplotlib** with a **python deque** datastructure to plot a circular buffer of state values as 2D lines.

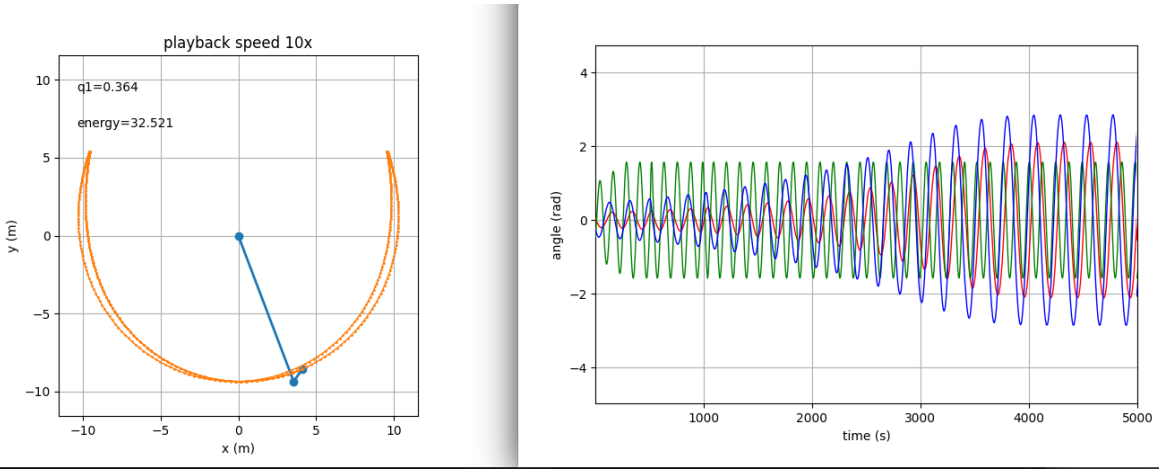

Fig. S3. Simulation, red, blue, green = $\theta_1, \dot{\theta_1}, \theta_2$ respectively

6. DISCUSSION AND CONCLUSION

For standardization, all experiments are done integrating the system dynamics every **0.02s** from **t=0-500s**. This was chosen to give the system enough time to **pump energy** and either **converge or go unstable**.

Design 1 describes a system where the outside of the 4D state variables, there are system parameters $(l_1, m_1, l_2, m_2, \beta_{damping})$ and control parameters $(k_p, k_d, \alpha, e^{desired})$. We set as fixed constant the gravity constant $g = 9.8$. Aside from these, we constrain the system to start at rest $(\dot{\theta}_1^0 = 0, \dot{\theta}_2^0 = 0)$ and can vary the initial state position (θ_1^0, θ_2^0).

Finally, we performed phase space analysis looking at $\theta_1, \dot{\theta_1}$. We choose this subset of the entire 4D state space because we care ultimately about θ_1, and it is not possible to plot a 4D change space on a 2D plot. We select the columns for $\theta_1, \dot{\theta_1}$ from our integration output, and add arrows to show directionality.

Across this array of **11** (5 system + 2 initial state position + 4 control) parameters to tune, we can evaluate them based on some performance metrics outlined below. You can see an example of the metrics in the title for S4. Note that to the **Hilbert transform** and **butter high-pass noise filtering** techniques were core to the data analysis to obtain an **envelope** on θ_1. All analysis was done using the open-source python packages **scipy, numpy**.

1. whether or not the system is able to pump energy into θ_1, (specifically, if the ratio between $\frac{\theta_1^{max}}{\theta_1^{initial}} > 2$)
2. if the system is stable (specifically, if the velocity in the last minute is less than threshold)
3. the rise time (seconds) to reach within some threshold to the maximum θ_1 (defined as time from start to the first occurrence of $\theta_1^{max} \pm \epsilon$)
4. actuator effort (sum of norm of accelerations) to achieve (3) (defined as $\Sigma \|\ddot{\theta_2}\|$)

We performed phase space analysis looking at $\theta_1, \dot{\theta_1}$. We choose this subset of the entire 4D state space because we care ultimately about θ_1, and it is not possible to plot a dynamic 4D state space on a 2D plot. Each point on the phase trajectory represents a snapshot of the system dynamics, so it is impossible to **statically** present the vector field outside of the trajectory point. We select the columns for $\theta_1, \dot{\theta_1}$ from our integration output, and add arrows to show directionality.

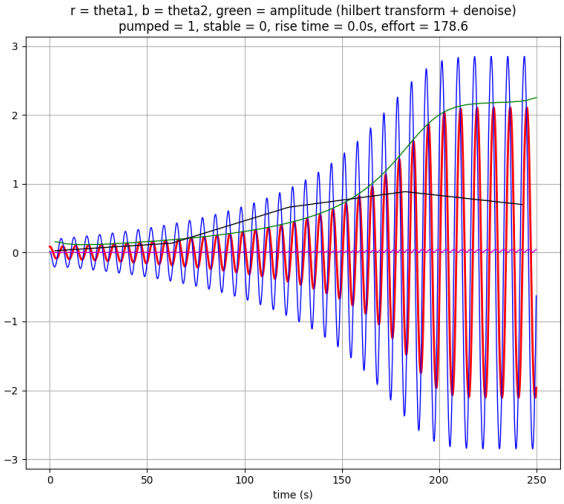

Fig. S4. Simulation metrics, black is $\|\ddot{\theta_1}\|$ and pink is $\|\ddot{\theta_2}\|$, other colors are in the plot title above

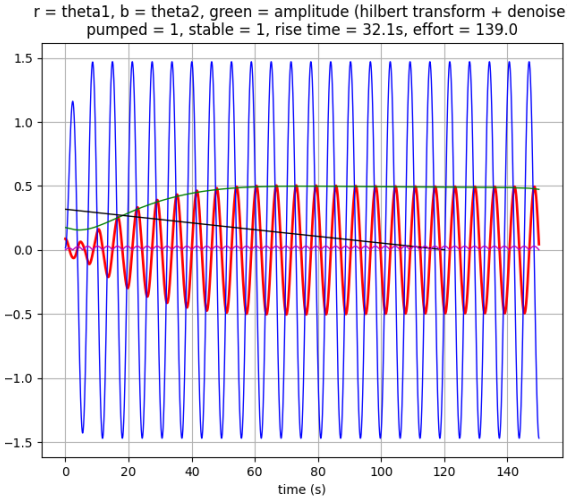

Fig. S5. Damping = 0.1 for non-autonomous reference type 2

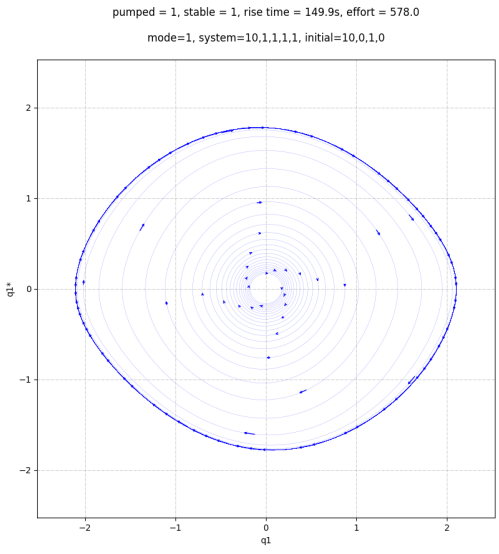

Fig. S6. Phase plot of θ_1, $\dot{\theta_1}$ for S4, note the homoclinic orbit

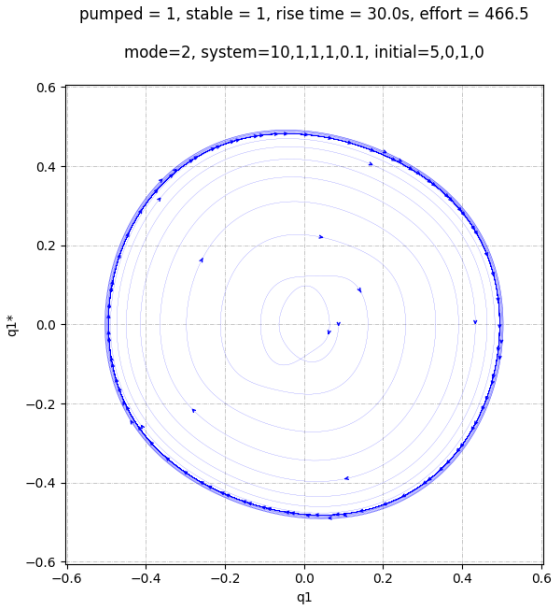

Fig. S7. Phase plot of θ_1, $\dot{\theta_1}$ for S5, note the homoclinic orbit

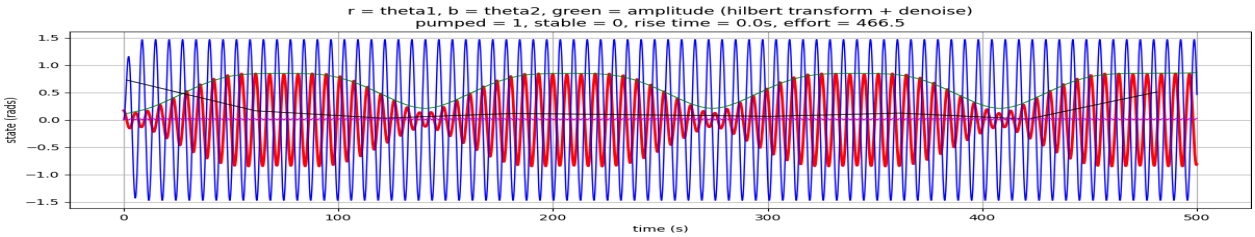

Fig. S8. No damping for non-autonomous reference type 2

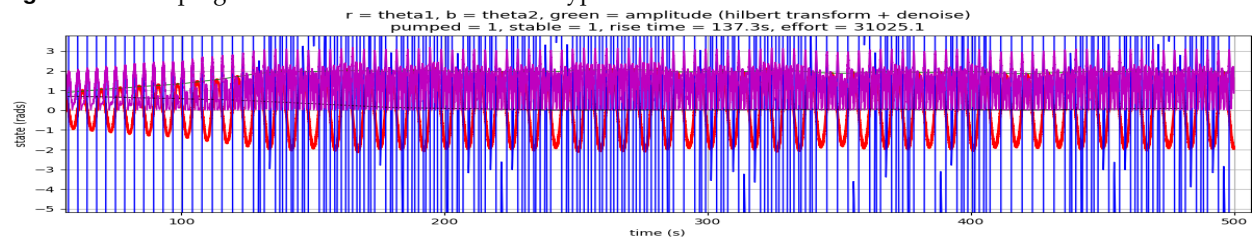

Fig. S9. Reference type 3, no damping

The figures and table 8 in the appendix show a selection of performances for various system, reference type, and control parameters. The metrics provided also quantify what is qualitatively observable. The state space plots show the system performance as a function of time, while the phase plots show how the **unactuated** θ_1 joint angle and velocity change along the trajectory and quantify some intuitions. One thing explored here is how systems perform with and without damping, and another is an introduction to how different references impact the unactuated joint θ_1. Some general statements can be made with this selection of experiments alone.

We can see that reference 3 exerts significantly more effort than reference 1 or 2. Because this reference is based on the potential energy which is a non-linear metric and constantly changing the pendulum swings, this perhaps explains why the commanded acceleration fluctuates very rapidly. It was observed that very often the gains are needed to 'damp' down the actuator acceleration, otherwise it 'interferes' with its own pumping antagonistically. The phase plot S11 also shows that the magnitude of $\theta_1, \dot{\theta}_1$ is **very** noisy and there is no high level pattern of change in phase space.

We can also see that reference 2 performs only with there is no or little damping, otherwise it does not achieve pumping into the system. We see that this is the only reference that is **non-autonomous**, or in other words it is a function of time, rather than system state. As a consequence of this, its performance is a kind of accident with the system timing. It is possible but not thoroughly explored if the timing of the system dynamics can constructively match the timing of the non-autonomous reference to achieve better pumping action. The phase plot S7 provided shows the system with damping = 0.1, and shows the magnitude of $\theta_1, \dot{\theta}_1$ increasing as energy is pumped into the system. The system then reaches a homoclinic orbit and is able to sustain that orbit indefinitely. It is also interesting that in a system with **no damping** show in S8 and S10, the system **cannot** sustain the homoclinic orbit with the tested reference timing, and therefore cycles between low and high energy. It is possible that a different reference trajectory could sustain this homoclinic orbit.

Finally, reference 1 is the reference found most commonly in literature, and the one that most closely resembles human behavior [1]. While the phase plot S6 does not show energy any information about actuator energy, it does show like for the reference 2 phase plot S7 that the system can reach a homoclinic orbit and sustain it indefinitely. We can see that the actuator effort is on the order of reference 2, and while it does not pump as quickly as reference 2 when it does, it is the only one thus far that can pump **in the presence of damping**. From this perspective, this reference is the most optimal one of the 3.

This paper details details the modeling, derivations of mathematical expressions, control techniques, and simulation results of select experiments of the acrobot mechanical system. We have shown quantitative results that validate the human swing-set pumping behavior as the **most optimal** control method among a set of possible alternatives from the perspective of pumping, stability, rise time, and energy spent. We have also outlined data analysis methods for evaluating swing set sinusoidal performance.

7. ACKNOWLEDGEMENTS

We would like to thank Professor Teodorescu at UC Santa Cruz and his ECE 216 course for introducing us to the study of biologically inspired robotics. His materials and guidance shaped this study in both content and structure, and fostered a systematic understanding of locomotion grounded in physics and mathematics.

8. APPENDIX

Parameters	Pumped	Stable	Rise time	Actuator Effort
$reference = \mathbf{1}$ *S4* $l_1 = 10.0, m_1 = 1.0, l_2 = 1.0, m_2 = 1.0, \beta_{damping} = 1.0$ $\alpha = 5.1, k_p = 20.0, k_d = 20.0$ $\theta_1 = 0.2, \theta_2 = 0.0$	True	True	150s	577.8
$reference = \mathbf{2}$ $l_1 = 10.0, m_1 = 1.0, l_2 = 1.0, m_2 = 1.0, \beta_{damping} = 1.0$ $\alpha = 1.0, k_p = 1.0, k_d = 1.0$ $\theta_1 = 0.2, \theta_2 = 0.0$	False	True	-	466.5
$reference = \mathbf{2}$ *S8* $l_1 = 10.0, m_1 = 1.0, l_2 = 1.0, m_2 = 1.0, \beta_{damping} = 0.0$ $\alpha = 1.0, k_p = 1.0, k_d = 1.0$ $\theta_1 = 0.2, \theta_2 = 0.0$	True	False	-	466.5
$reference = \mathbf{2}$ *S5* $l_1 = 10.0, m_1 = 1.0, l_2 = 1.0, m_2 = 1.0, \beta_{damping} = 0.1$ $\alpha = 1.0, k_p = 1.0, k_d = 1.0$ $\theta_1 = 0.2, \theta_2 = 0.0$	True	True	29.5s	466.5
$reference = \mathbf{3}$ *S9* $l_1 = 10.0, m_1 = 1.0, l_2 = 1.0, m_2 = 1.0, \beta_{damping} = 0.0$ $\text{energy}_goal = 100.0, k_p = 5.0, k_d = 5.0$ $\theta_1 = 0.3, \theta_2 = 0.0$	True	True	137.3	31025.1

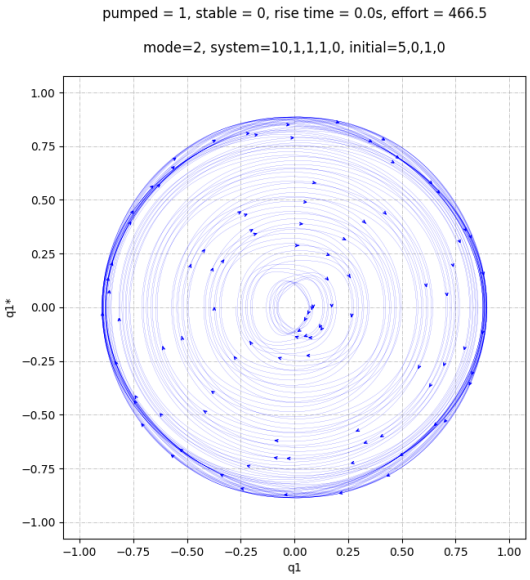

Fig. S10. Phase plot of $\theta_1, \dot{\theta}_1$ for S8

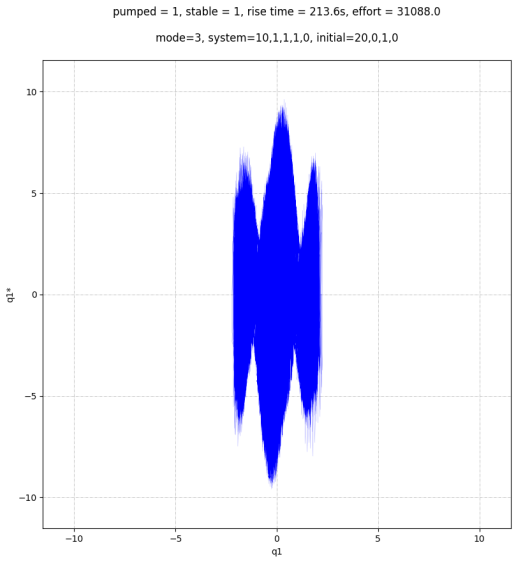

Fig. S11. Phase plot of $\theta_1, \dot{\theta}_1$ for S9

REFERENCES

1. S. Wirkus, R. H. Rand, and A. Ruina, "How to pump a swing," Coll. Math. J. **29**, 266–275 (1998).

2. R. Tedrake, *Underactuated Robotics* (2022).
3. T. Chuangyanyong, P. Chinsakuljaroen, W. Ketrungsri, and T. Choopojcharoen, "Flying trapeze act motion planning algorithm for two-link free-flying acrobatic robot," CoRR **abs/2111.03823** (2021).
4. A. Shkolnik and R. Tedrake, "High-dimensional underactuated motion planning via task space control," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, (2008), pp. 3762–3768.