Dipartimento di Ingegneria Informatica, Automatica e Gestionale
(DIAG)

Energy-based swing-up control of the Acrobot

Professor:

Prof. Leonardo Lanari

Tutor:

Dott. Tommaso Belvedere

Autors:

Diego Marussi
Marco Nacca
Luca Ruscitti

Anno accademico 2022/2023

Contents

1 Introduction

In the fields of control theory and robotics, stabilizing an underactuated system like an acrobot, a two-link pendulum with just one actuated joint, poses a considerable problem. Due to their innate nonlinearities and underactuation, such systems frequently resist effective control by traditional control approaches. Energy-based control is a potential strategy that has surfaced in recent years.

In order to create controllers that can stable intricate mechanical systems like the acrobot, energy-based control approaches have become increasingly popular. These controllers concentrate on preserving or controlling the system's energy to achieve stability and the desired behavior rather than depending on explicit models of system dynamics. The acrobot makes a perfect testbed for energy-based control techniques due to its two-link topology and restricted actuation. In this method, the kinetic and potential energy of the system are captured by an energy function. The controller seeks to shape the acrobot's energy landscape to ensure stable and controllable mobility by modifying the control input, often at the actuated joint.

The following work addresses the problem of energy-based swing-up control for the Acrobot. The robot starts from a stable equilibrium position (the robot is positioned downwards) and, after receiving a torque that allows it to arrive close to a desired position, is subject to the action of another controller, called LQR (Linear Quadratic Regulator) whose task is to stabilize the Acrobot to bring it to the desired position, which is unstable, with the aim of balancing it.

Acrobots and gravity-driven passive joint systems are gaining in importance as they are mechanical systems capable of performing tasks with fewer actuators than their degrees of freedom. In recent years, several approaches to the problem have been developed that require the implementation of an energy-based controller. [1] [4]

Spong [3] proposed a controller that combines nonlinear linearization with partial feedback and energy modeling to swing-up the robot.

De Luca, Oriolo [2] have developed a strategy that is based on the use of a simple swing-up that allows you to bring the first link to the position upwards and not in a region where the activation of an LQR controller is required , and the use of an ISS (Iterative State Steering) approach to bring the Acrobot to the desired position.

As already mentioned previously, our goal is to develop a controller that allows the Acrobot to arrive in the vicinity of the desired position and then be balanced, without remaining stuck in other singular balance points.

The following paper is organized as follows: Section 2 contains the preliminary theoretical treatment to address the problem, then the Section 3 explain the problem statement in which we talk about the dynamic model of the Acrobot. Then, in the Section 4 we have the the developed algorithm and finally, in the Section 5, the conclusion about the work.

2 Preliminaries

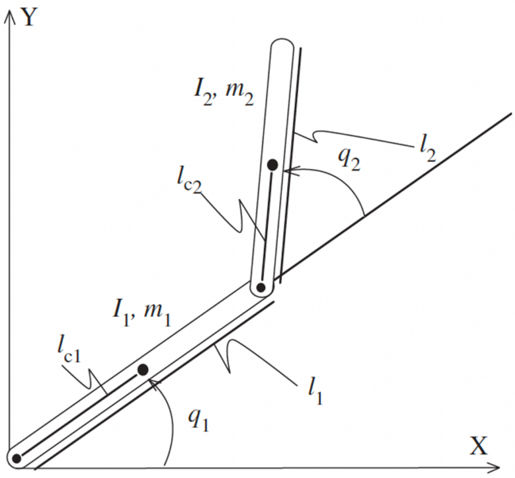

Figure 1: Acrobot model

Referring to the figure (1), the Acrobot dynamics equation can be described as follows

$$M(q)\ddot{q} + C(q, \dot{q}) + G(q) = \tau \tag{1}$$

where $q = [q_1, q_2]^T$ and

$$M(q) = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} = \begin{bmatrix} \alpha_1 + \alpha_2 + 2\alpha_3 \cos q_2 & \alpha 2 + \alpha_3 \cos q2 \\ \alpha 2 + \alpha_3 \cos q2 & \alpha 2 \end{bmatrix}$$

$$C(q, \dot{q}) = \begin{bmatrix} H_1 \\ H_2 \end{bmatrix} = \alpha_3 \begin{bmatrix} -2\dot{q_1}\dot{q_2} - \dot{q_2}^2 \\ \dot{q_1}^2 \end{bmatrix} \sin q_2$$

$$G(q) = \begin{bmatrix} G_1 \\ G_2 \end{bmatrix} = \begin{bmatrix} \beta_1 \cos q_1 + \beta_2 \cos(q_1 + q_2) \\ \beta_2 \cos(q_1 + q_2) \end{bmatrix}$$

$\tau = [\tau_1, \tau_2]^T$ be the torque applied to joint 1 and joint 2 and:

$$\alpha_1 = m_1 l_{c1}^2 + m_2 l_1^2 + I_1$$

$$\alpha_2 = m_2 l_{c2}^2 + I_2$$

$$\alpha_3 = m_2 l_1 l_{c2}$$

$$\beta_1 = (m_1 l_{c1} + m_2 l_1) g$$

$$\beta_2 = m_2 l_{c2} g$$

and g the gravity acceleration. The robot is called the Acrobot if $t_1 = 0$. The energy of the Acrobot is expressed as

$$E(q, \dot{q}) = \frac{1}{2}\dot{q}^T M(q)\dot{q} + P(q) \tag{2}$$

where $P(q)$ is the potential energy and is expressed as:

$$P(q) = \beta_1 \sin q_1 + \beta_2 \sin(q_1 + q_2) \tag{3}$$

3 Problem Statement

The essential part of our project concerns the derivation and use of an energy-base swing-up controller for an Acrobot. According to the work of [1], it is assumed that there are no singularity points in the control law. If the balance point that the Acrobot has when it is in a vertical position has been considered:

$$q_1 = \pi/2 \qquad\qquad q_2 = 0 \qquad\qquad \dot{q_1}, \dot{q_2} = 0$$

the Lyapunov function can be defined like this:

$$V = \frac{1}{2}(E - E_r)^2 + \frac{1}{2}k_D\dot{q}^2 + \frac{1}{2}k_P q_2^2$$

where $E_r = E(q, \dot{q}) = \beta_1 + \beta_2$ is the energy evaluated at the upright equilibrium point and k_p, k_D are positive gains.

Taking into account that $\dot{E} = \dot{q}^T\tau = \dot{q_2}\tau_2$, then the derivative of the Lyapunov function along the dynamic model of the Acrobot can be calculated

$$\dot{V} = \dot{q_2}((E - E_r)\tau_2 + k_D\ddot{q_2} + k_P q_2) \tag{4}$$

At this point, τ_2 can be chosen such that

$$(E - E_r)\tau_2 + k_D\ddot{q_2} + k_P q_2 = -k_V\dot{q_2} \tag{5}$$

and $\dot{V} = -k_V\dot{q_2} \leq 0$.

Obtaining $\ddot{q_2}$ from (1) and substituting it into (5), as in [1]

$$\left(k_D + \frac{(E - E_r)\Delta}{M_{11}}\right)\tau_2 = -\frac{(k_V\dot{q_2} + k_P q_2)\Delta + k_D(M_{21}(H_1 + G_1) - M_{11}(H_2 + G_2))}{M_{11}} \tag{6}$$

where $\Delta = M_{11}M_{22} - M_{12}M_{21} = \alpha_1\alpha_2 - \alpha_3^2\cos^2 q_2 > 0$. So, when $k_D + \frac{(E-E_r)\Delta}{M_{11}} \neq 0$, we can calculate the torque needed to perform the task

$$\tau_2 = -\frac{(k_V\dot{q_2} + k_P q_2)\Delta + k_D(M_{21}(H_1 + G_1) - M_{11}(H_2 + G_2))}{k_D M_{11} + (E - E_r)\Delta} \tag{7}$$

In the torque equation just mentioned, there are the weighted gains k_D, k_P which are respectively the derivative gain and the proportional gain. They must necessarily respect limits, which are given by:

$$k_D > max_q f(q) \tag{8}$$

with $f(q) = \frac{(E_r - P(q))\Delta}{M_{11}}$.

Instead, regarding proportional gain, the [1] uses a proposition about it, called *Proposition 4*. It states that if the gain k_D satisfies (8), then k_P must satisfy the following proposition:

$$k_P > \frac{2}{\pi} min(\beta_1^2, \beta_2^2) \tag{9}$$

with $k_V > 0$.

To simplify the (8), we can make an assumption, i.e. that the maximum of a function $f(q_1, q_2)$ equals a function that depends exclusively on the parameter q_2. For every q such that $f(q)$ takes on its maximum value

$$\frac{\partial f(q)}{\partial q_1} = \frac{\partial}{\partial q_1} \left(\frac{(E_r - P(q))\Delta}{M_{11}} \right) = 0 \tag{10}$$

must hold. Since Δ / M_{11} is a function of only q_2, then

$$\frac{\partial P(q)}{\partial q_1} = G_1 = \beta_1 \cos q_1 + \beta_2 \cos(q_1 + q_2) = 0 \tag{11}$$

From $P^2 = P^2 + G_1^2 = \beta_1^2 + \beta_2^2 + 2\beta_1\beta_2 \cos q_2$, then

$$P(q) = \pm\Phi(q_2) \tag{12}$$

when $G_1 = 0$ and where $\Phi = \sqrt{\beta_1^2 + \beta_2^2 + 2\beta_1\beta_2 \cos q_2}$.

Taking $P(q) = -\Phi(q_2)$, the (8) is equivalent to

$$k_D > max_{q_2 \in [0, 2\pi]} \frac{(\Phi(q_2) + E_r)\Delta(q_2)}{M_{11}(q2)} \tag{13}$$

Using LaSalle's theorem as in [1] for analysis of stability for dynamical systems, it provides information about the behavior of trajectories of a dynamical system in terms of invariant sets, under the controller (7), and knowing that $\dot{V} = -k_v \dot{q_2}^2 \leq 0$, and that V is bounded. Defining, as in [1],

$$\Psi = \{(q, \dot{q}) | V(q, \dot{q}) \leq c\} \tag{14}$$

where c is a costant, then any solution $q(t), \dot{q}(t)$ starting in Ψ remains in this set for all time.

The biggest invariant set, as used in LaSalle's theorem, is the largest subset of the state space where the system's paths remain within it under specific circumstances. In this case, it is defined as W in

$$S = \{(q, \dot{q}) | \dot{V} = 0\} \tag{15}$$

through LaSalle's theorem we know that every pair $(q(t), \dot{q}(t))$ starting in Ψ approaches W as $t \to \infty$, as seen in [1]. Since $\dot{V} = 0$ holds in W, then V, q_2 are constant in W.

Using (4), it's know also that E is constant in W. This can be summarized as

$$\lim_{t\to\infty} E = E^* \qquad\qquad \lim_{t\to\infty} q_2 = q_2^* \tag{16}$$

Finally, substituting $q_2 \equiv q_2^*$ and $E \equiv E^*$ into (2),

$$\dot{q_1}^2 = \frac{2E^* - 2\beta_1 \sin q_1 - 2\beta_2 \sin(q_1 + q_2^*)}{\alpha_1 + \alpha_2 + 2\alpha_3 \cos q_2^*} \tag{17}$$

The largest invariant set W can be expressed as

$$W = \{(q, \dot{q}) | (q_1, \dot{q}_1) \text{ satisfies (17) and } q_2 \equiv q_2^*\} \tag{18}$$

This can be summarized by saying that the controller (7) has no singular point for the Acrobot starting from any initial state for all future time if and only if (13) hold. But also (16) hold, and every pair $(q(t), \dot{q}(t)$ approaches the invariant set W.

Once the energy-based torque is obtained for the purpose of performing a swing-up maneuver with the aim of reaching close to the equilibrium position, there is a need for the second controller, i.e. the LQR, to be activated with the aim of stabilize the system.

The Linear Quadratic Regulator (LQR) controller is a used control technique in the field control theory. It is important for regulating the behavior of linear dynamical systems.

The LQR controller aims to optimize the performance of a system by designing a control law that minimizes a quadratic cost function. This cost function typically captures the trade-off between control effort and system state deviations from desired values. By carefully selecting gain factors in the cost function, control objectives can be obtained. By solving the algebraic Riccati equation in continuous or discrete time, the optimal control law that minimizes the cost function and satisfies the system dynamics can be derived.

An important role for the LQR controller is the choice of identifying the appropriate weighting matrices, Q and R, which play a key role in determining controller performance and behavior.

The Q matrix, often called the state weighting matrix, is responsible for the cost associated with deviations of the system state variables from desired values. It can be set to give more importance to some values than to others. For example, if stability is critical, the Q matrix can be configured to give greater weight to angular positions or velocities to ensure that they are tightly controlled.

On the other hand, the R matrix, known as the control input weighting matrix, determines the cost associated with manipulating system control inputs with the aim of prolonging components' life or reducing power consumption.

To calculate the controller, the following formula is used

$$\tau_2 = -Fx \tag{19}$$

where x is the state error, in particular the difference between the actual and desired state.

Obviously, there must be a way for us to switch from one type of control to another. This way is provided by a switching condition that involves state entirely, that is, position and velocity:

$$|x_1| + |x_2| + 0.1|x_3| + 0.1|x_4| < \zeta \tag{20}$$

where, in the [1], $\zeta = 0.04$.

Once we have reached the threshold, we make sure that from this moment on the LQR controller is activated and that the latter remains active for the duration of the experiment. In fact, as we will see later from the graphs, we will get a stable situation around 10 seconds.

4 Algorithm

Once we have seen the theoretical part, the following chapter will look at the development of the algorithm that is based on the notions explained above.

As a starting point, the statement of the parameters that are needed to calculate our dynamic system was made. In particular, the dynamic parameters of the Acrobot are:

$$\begin{cases} m_1 = 1kg \\ m_2 = 1kg \\ l_1 = 1m \\ l_2 = 2m \\ l_{c1} = 0.5m \\ l_{c2} = 1m \\ I_1 = 0.083kgm^2 \\ I_2 = 0.33kgm^2 \end{cases} \tag{21}$$

In addition to this, starting and goal state has been defined as in [1]

$$q_{start} = [-1.4046, 0, 0, 0] \tag{22}$$

$$q_{goal} = [\pi/2, 0, 0, 0] \tag{23}$$

The first step is to calculate the swing up-motion of the Acrobot. The swing-up, as said before, is composed by two parts:

- Energy computation;
- Control torque computation.

In the first part, as said before, the energy shaping has been calculated

$$E(q, \dot{q}) = \frac{1}{2}\dot{q}^T M(q)\dot{q} + \beta_1 \sin q_1 + \beta_2 \sin(q_1 + q_2)$$

By simply going to replace the values declared at the beginning within the clauses

$$\alpha_1 = m_1 l_{c1}^2 + m_2 l_1^2 + I_1$$

$$\alpha_2 = m_2 l_{c2}^2 + I_2$$

$$\alpha_3 = m_2 l_1 l_{c2}$$

$$\beta_1 = (m_1 l_{c1} + m_2 l_1) g$$

$$\beta_2 = m_2 l_{c2} g$$

Once we have done this, with the energy we have just calculated, we calculate the value of the torque as

$$\tau_2 = -\frac{(k_V \dot{q_2} + k_P q_2)\Delta + k_D(M_{21}(H_1 + G_1) - M_{11}(H_2 + G_2))}{k_D M_{11} + (E - E_r)\Delta} \tag{24}$$

where $E_r = \beta_1 + \beta_2$ and $\Delta = \alpha_1 \alpha_2 - \alpha_3^2 \cos^2 q_2$. Obviously, as said before, the maximum value that the gain can be has been calculated as

$$k_D > max_{q_2 \in [0, 2\pi]} \frac{(\Phi(q_2) + E_r)\Delta(q_2)}{M_{11}(q2)} \tag{25}$$

$$k_P > \frac{2}{\pi} min(\beta_1^2, \beta_2^2) \tag{26}$$

In particular, to achieve a rapid swing-up control of the Acrobot, $k_V = 66.3, k_D = 35.8, k_P = 61.2$ has been chosen to satisfy the constraints (8)(26) as in [1]. At this point, what we get is the torque and error in energy that the controller needs. This information is critically important because it allows us to monitor the trend in torque as the energy approaches its desired value, so as we come to convergence with $E = E - E_r = 0$. In fact, as we will see later from the graphs, the more error in energy we have, the more torque we need to bring the Acrobot to the desired position.

As of now, the next step is to get the next state of the robot. It can be obtained through integration using a library called *scipy.odeint*; this library performs integration more precisely than integration done using Euler's method, since it allows us to perform a number n of integrations of a single simulation timestep.

In this regard, given the dynamics of our Acrobot, the torque and initial velocity of the joints, we can obtain the next state through integration of the dynamics, thus calculating the acceleration of the current state as

$$\ddot{q_1} = -\frac{M_{21}\tau_2 - G_2 M_{21} + G_1 M_{22} - H_2 M_{21} + H_1 M_{22}}{M_{11} M_{22} - M_{21} M_{21}} \tag{27}$$

$$\ddot{q_2} = \frac{M_{11}\tau_2 - G_2 M_{11} + G_1 M_{21} - H_2 M_{11} + H_1 M_{21}}{M_{11} M_{22} - M_{21} M_{21}} \tag{28}$$

The response of the Acrobot to control inputs has been fully characterized by solving for acceleration and numerically integrating it over time. By precisely simulating the system's motion, we have gained a better understanding of its stability, controllability, and sensitivity to various initial circumstances and control methods.

Lastly, we moved from the swing-up controller to a stabilizing controller when the Acrobot was swing up to a specified small neighborhood of the upright equilibrium point. The same LQR controller as the stabilizing controller is employed

$$\tau_2 = -Fx \tag{29}$$

where $x = q - q_{des}$ is the state error and $F = [246.481 - 98.690 - 106.464 - 50.138])$ was computed using the LQR function with $Q = 4 \times 4$ matrix and $R = 1$.

5 Simulation Results

The simulation, that implement the algorithm, was developed in Python, with the help of a library called PyBullet for the graphics part, which makes use of a template *urdf* representing Acrobot. This library allows setting the parameters necessary for the robot to perform movements, such as the value of the position of the joints and their velocities. In this way, graphical feedback can be obtained that is useful in understanding whether the goal of the task is being achieved.

Referring to the starting state (22) and goal state (23), we can see, from the following images, the same results were obtained as the [1]. Regarding the position of the Acrobot, the link 2 approaches 0 while the link 1 swings up quickly to the vertical from the time response of $q_1 - \pi/2$.

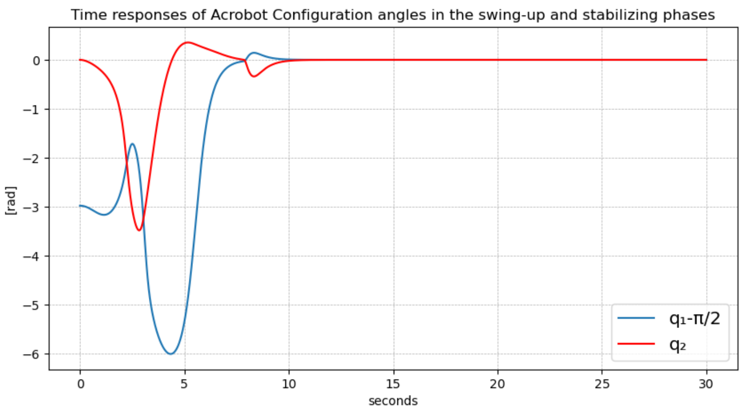

Figure 2: Configuration angle q wrt time

In the "swing-up" maneuver of an Acrobot, the joint speeds play a crucial role, namely that of increasing the kinetic energy of the system, lifting the Acrobot towards the vertical position. The joint speeds must be coordinated so that link 1 is accelerated upward while link 2 is pushed in a direction that contributes to the overall kinetic energy buildup. This coordination is essential to ensure that the system gains energy efficiently.

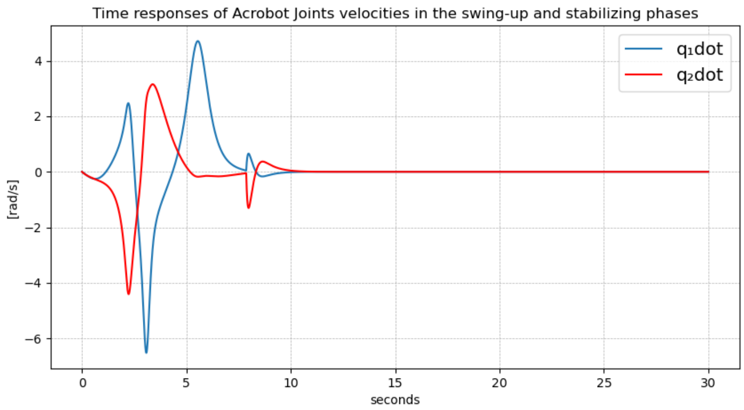

Figure 3: Joint velocity dq wrt time

Then, from the following figure, can be observed the primary goal of the Acrobot's "swing-up" phase is to raise the system's overall energy from a lower starting point to a desirable level. The achievement of this goal is indicated by the error in energy going to zero, which occurs when the system's total energy approaches or reaches the target level perfectly, and the system is ready for the following balancing phase in the vertical position. The control is used to apply torque to the joints to increase the total kinetic energy by swinging the Acrobot upward to the vertical position with the aim to achieve the error in energy of 0.

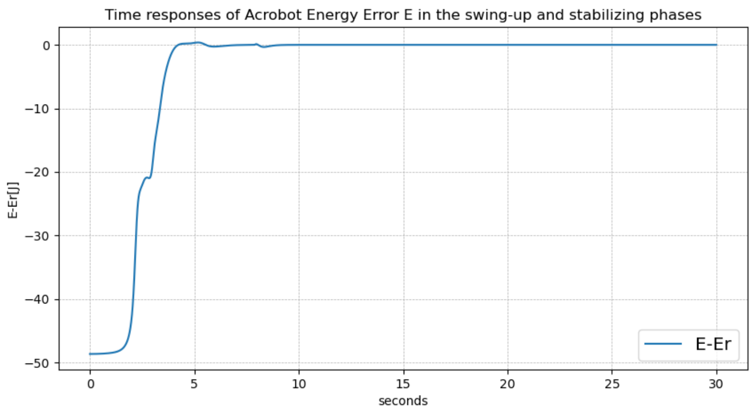

Figure 4: Energy Error wrt time

The following figure explain the phase involves applying torque to the system's joints to boost the system's kinetic energy, which raises the Acrobot to the vertical position. The objective is to build up potential energy in the system, which may later be used to balance it vertically or carry out other tasks.

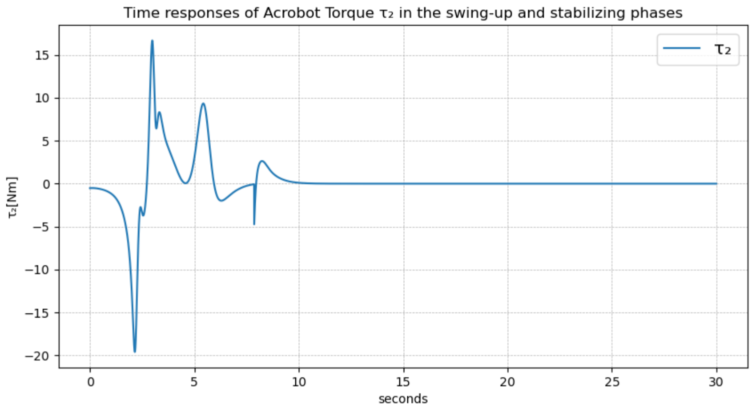

Figure 5: Torque τ wrt time

The torque must be precisely and dynamically measured in order to achieve 0 mistake in energy throughout this "swing-up" period. To ensure that the system has exactly the amount of energy needed to move to the target location, the torque control must be adjusted in real-time.

Finally, the following figure explain the phase portrait of $(q_1 - \pi/2, \dot{q_1})$, where $(q_1, \dot{q_1})$ approaches to homoclinic orbit

$$\dot{q_1}^2 = \frac{2E_r}{\alpha_1 + \alpha_2 + 2\alpha_3}(1 - \sin q_1). \tag{30}$$

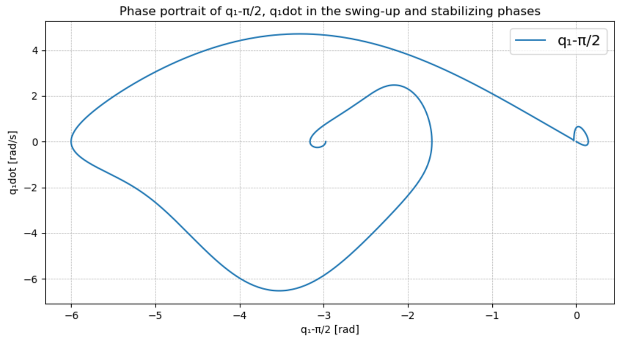

Figure 6: Phase portrait of $(q_1 - \pi/2, \dot{q_1})$ in the swing-up phase

A trajectory that continuously approaches an equilibrium point or steady state without ever converging entirely into it is commonly referred to in this context as a homoclinic orbit. In the case under consideration, the vertical position that the Acrobot reaches without ever fully reaching it is of particular interest. It is essential to understand how the system could be effectively guided to approach a particular vertical position and stay stable there with it.

Another test was carried out with the same parameter (21), but with a starting and goal state defined as follows:

$$q_{start} = [-1.4, 0, 0, 0] \tag{31}$$

$$q_{goal} = [\pi/2, 0, 0, 0] \tag{32}$$

It can be seen that, although the energy error converges towards the value zero, the controller fails to converge, resulting in an incorrect swing-up.

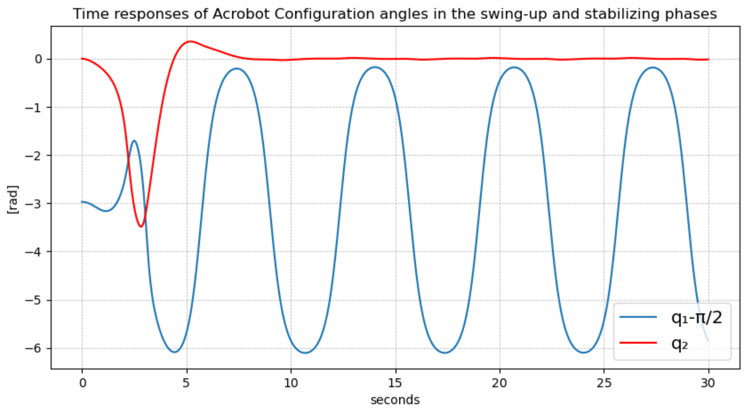

Figure 7: Configuration angle q wrt time

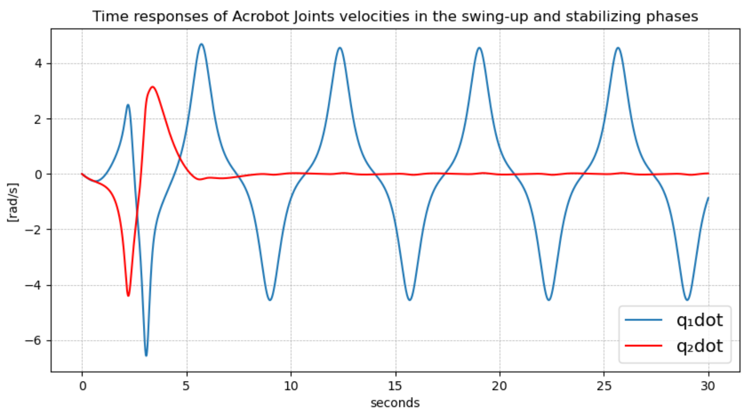

Figure 8: Joint velocity dq wrt time

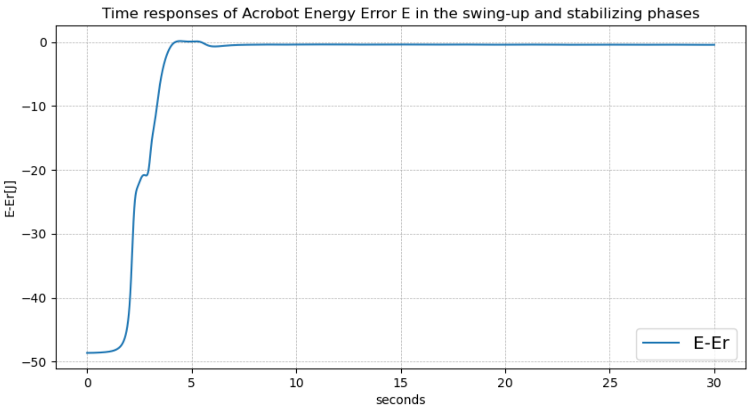

Figure 9: Energy Error wrt time

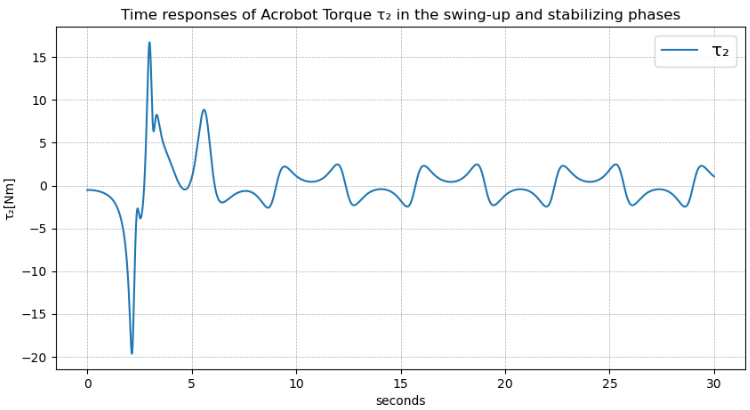

Figure 10: Torque τ wrt time

Unlike what is present in the reference report [1], i.e. that the system should function independently of the chosen initial condition, setting the starting state equal to $(-1.4, 0, 0, 0)$, we obtain an error that converges to zero but in this case the controller does not seem to converge as the trend of the joint q_1 does not fully reach zero and this involves the non-activation of the LQR controller in order to stabilize the system . Having tried different solutions, changed the dynamics and the way of carrying out the integration, we believe that the problem is intrinsic to the simulation environment used, as different approximations of the values are obtained, not following a univocal law.

6 Conclusion

In conclusion, the development of an energy-based controller for the "swing-up" process of an Acrobot is a key step in the control of complex dynamic systems. Through analysis and management of the system's energy, we were able to design a controller that allowed the Acrobot to transition from a low-energy state to a higher-energy state, preparing it for subsequent balancing and control steps.

The successful implementation of this "swing-up" controller underscores the potential of energy as a crucial metric for controlling physical and robotic systems. By purposely setting gains within the controller following appropriate constraints, we were able to swing an Acrobot from a low energy position to a vertical position following a desired energy and subsequently stabilize it by applying an LQR controller. The simulation results were provided to demonstrate the validity of the theoretical results.

References

[1] X.Xin and M.Kaneda, Analysis of the energy-based swing-up control of the Acrobot, 5 March 2007.

[2] Oriolo G. and De Luca A., Stabilization of the Acrobot via Iterative State Steering, May 1998.

[3] Mark W. Spong, The Swing Up Control Problem For The Acrobot, February 1995.

[4] X.Xin and M.Kaneda, A Robust Control Approach to the Swing up Control Problem for the Acrobot, 3 November 2001.