Tytul naszego artykulu brzmi nastepujaco.....

Paweł Parulski * **Patryk Bartkowiak** **
Krzysztof Kozłowski *** **Dariusz Pazderski** ****

* *Institute of Automation and Robotics, Poznan University of Technology, Poznan, Poland (e-mail: pawel.parulski@put.poznan.pl).*
** *Institute of Automation and Robotics, Poznan University of Technology, Poznan, Poland (e-mail: author@put.poznan.pl)*
*** *Institute of Automation and Robotics, Poznan University of Technology, Poznan, Poland (e-mail: author@put.poznan.pl)*
**** *Institute of Automation and Robotics, Poznan University of Technology, Poznan, Poland (e-mail: author@put.poznan.pl)*

Abstract: The aim of this paper is to test the usefulness of a new approach presented in [Li et al. (2019)] to control the Pendubot. The control problem stated in the article is to stabilize the Pendubot in the upright position.

Keywords: Robotics, Robot dynamics, Pendubot, nonlinear control, feedback linearization, nonlinearity,

1. INTRODUCTION

Acrobot and Pendubot are well-known fundamental examples of underactuated planar mechanical systems in the presence of the gravity. Although their kinematic structures are relatively simple, still they can be considered as important benchmark systems which are useful for the development of advanced control strategies.

A standard procedure at the stabilization process for the Pendubot system, as well as for the twin structure which is the Acrobot, is usually divided into 2 subproblems, i.e. the task of swing-up (when the robot hangs down) and the balancing task around the equilibrium point.

Basically, many solutions to the first problem are based on the work of Spong (1994); Xin and Kaneda (2004), the second, is still being developed by many authors.

The paper is organized as follows: Section 2 describes the mathematical model for the Acrobot system, together with constrains imposed on the model. In section 3 the analyzed control algorithms along with the state transformation for the stabilization task is introduced. Section 4 describes the simulation results of analyzed algorithms, while the Section 5 gives final remarks.

The main attempt of this paper is to verify via simulations the the new approach to control the Pendubot that has recently appeared in the literature [Li et al. (2019)]. Authors are also interested in the applicability of such an algorithm to the physically existing root.

The considered algorithm has recently been published, thus it does not have any simulation or experimental verification. In this paper authors attempt to fill this gap and extend preliminary results recently discussed by Bartkowiak (2019).

2. ROBOT'S MODEL

The Pendubot robot is a connection of $N = 2$ rigid bodies coupled in a tree structure, supported on ground via an actuated frictionless revolute joint. Both links have non-zero mass and the revolute joint connecting them is unactuated. As a result, the system has one degree of underactuation (2 DOF with 1 independent actuator). In Fig. 1 the reference frame has been attached at the

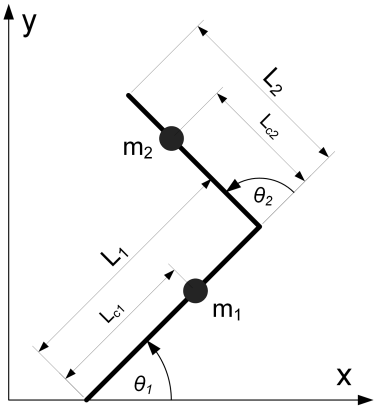

Fig. 1. Two link inverted pendulum, with one actuator – Pendubot

pivot point, and coordinates are indicated as $\theta = (\theta_1, \theta_2)$. The kinetic energy is quadratic, $K = \frac{1}{2}\dot{\theta}^T D(\theta)\dot{\theta}$, with D being a positive definite matrix. The potential energy, V, depends only on the configuration variables. Denote the Lagrangian by $L = K - V$, and impose the actuation on a system as below:

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}_k} - \frac{\partial L}{\partial \theta_k} = \begin{cases} \tau_k, & k = 1 \\ 0, & k = 2 \end{cases} \tag{1}$$

with τ_k taking values in $\mathcal{R}$. The mathematical model of the system dynamics thus takes the form

$$D(\theta)\ddot{\theta} + C(\theta,\dot{\theta})\dot{\theta} + G(\theta) = B\tau \tag{2}$$

where $B = [\,I\ \ 0\,]^T$ and

$$\begin{aligned} d_{11}(\theta_2) &= a_1 + a_2 + 2a_3\cos\theta_2 \\ d_{12}(\theta_2) &= a_2 + a_3\cos\theta_2 \\ d_{21}(\theta_2) &= d_{12} \\ d_{22}(\theta_2) &= a_2 \end{aligned}$$

$a_1 = m_1L_{c_1}^2 + m_2L_1^2 + I_1$, $a_2 = m_2L_{c_2}^2 + I_2$, $a_3 = m_2L_1L_{c_2}$, $a_4 = m_1L_{c_1} + m_2L_1$, $a_5 = m_2L_{c_2}$,

$$C(\theta,\dot{\theta}) = \begin{bmatrix} -2a_3\sin\theta_2\dot{\theta}_2 & -a_3\sin\theta_2(\dot{\theta}_1+\dot{\theta}_2) \\ a_3\sin\theta_2\dot{\theta}_1 & 0 \end{bmatrix}, \tag{3}$$

$$G(\theta) = \begin{bmatrix} ga_5\cos(\theta_1+\theta_2) + a_4\cos\theta_1 \\ ga_5\cos(\theta_1+\theta_2) \end{bmatrix}. \tag{4}$$

Assuming some simplifications and ignoring dynamics of the motor, one can treat τ as a real torque generated by the motor.

2.1 Robot's parameters

The choice of the robot parameters (Table 1) in the simulation was selected in order to adapt them to the physically existing mechanism, i.e. to the one legged robot presented in Fig. 2. This one-legged mechanism can be successfully studied as a double inverted pendulum. Considered robot was built in the Institute of Automation and Robotics at Poznan University of Technology, as a testbed for a four-legged walking robot [Kozłowski et al. (2008, 2013)]. The

Table 1. Robot's parameters

Link	Mass $[kg]$	Centre of mass $[m]$	Length $[m]$	Inertia $[kg\,m^2]$
1	1.593	0.074	0.15	0.0119
2	0.405	0.134	0.295	0.0117

physical model depicted in Fig. 2 is driven by Maxon's 200W EC-Powermax 30 brushless motors with planetary gearhead of $n = 53$ reduction. Such drives provides the maximum torque of approximately 6 Nm. This value is considered as a saturation of the control input.

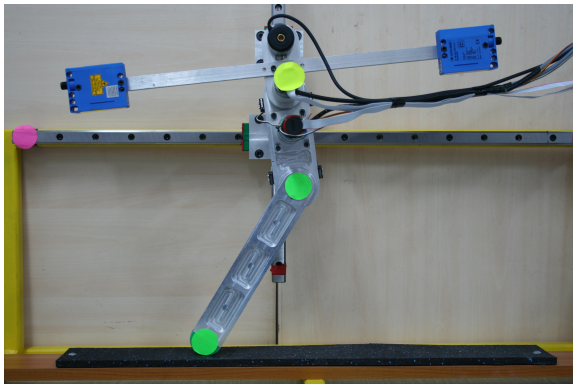

Fig. 2. Experimental test-bed

3. STABILIZATION PROBLEM

The system (1) for $\tau_1 = 0$ has four equilibrium points: first of them are two unstable positions (top and mid position) where $(q_1, q_2, \dot{q}_1, \dot{q}_2) = (\frac{\pi}{2}, 0, 0, 0)$ and $(q_1, q_2, \dot{q}_1, \dot{q}_2) = (-\frac{\pi}{2}, \pi, 0, 0)$, respectively. The other two are trivial $(\frac{\pi}{2}, \pi, 0, 0)$, and $(-\frac{\pi}{2}, 0, 0, 0)$ and they are not being analyzed.

The aim of this work is to verify considered algorithm whether it is capable of stabilizing the system around its top unstable position, taking into account the limitations and constraints resulting from practical or physical conditions (existing robot).

3.1 Control algorithm

The control of double inverted pendulum is still an open problem. As the fully actuated system is trivial in control, the underactuated is not. The [Li et al. (2019)] presented new approach the the underactuated case. In the recently published article, authors try to find the largest feedback linearizable subsystem for mechanical systems that are not feedback linearizable. Moreover, they consider the problem of choosing a partially linearizing output that renders the zero dynamics asymptotically stable, and illustrate the theoretical results by mathematical formulas developed for double inverted pendulum with one actuator (Acrobot and Pendubot). The paper presents an attempt to use such algorithm to control the Pendubot from any initial condition to equilibrium pose. This method can be briefly characterize as follows. At first, rewrite the Eq. (2) in the following form:

$$\begin{aligned} d_{11}\ddot{\theta}_1 + d_{12}\ddot{\theta}_2 + \mu_1 + \phi_1 &= \tau \\ d_{21}\ddot{\theta}_1 + d_{22}\ddot{\theta}_2 + \mu_2 + \phi_2 &= 0 \end{aligned} \tag{5}$$

where: d_{ij} are the entries of matrix $D(\theta)$, and $\mu_1 = C_{11}(\theta,\dot{\theta})\dot{\theta}_1 + C_{12}(\theta,\dot{\theta})\dot{\theta}_2$, $\mu_2 = C_{21}(\theta,\dot{\theta})\dot{\theta}_1$, $\phi_1 = G_1(\theta)$, $\phi_2 = G_2(\theta)$, where C_{ij}, G_i are entries of Eq. (3) and Eq. (4), respectively, and apply the feedback transformation according to Spong (1994). The overall robot's dynamics model after this transformation can be written in the following form:

$$\Sigma_{pend}: \quad \begin{aligned} \dot{\theta}_1 &= w_1 \\ \dot{w}_1 &= u \\ \dot{\theta}_2 &= w_2 \\ \dot{w}_2 &= -d_{22}^{-1}\mu_2 - d_{22}^{-1}\phi_2 + J_2(\theta_2)u. \end{aligned} \tag{6}$$

where: $J_2(\theta_2) = -d_{22}^{-1}d_{21}$. After introducing new state variables (7)

$$\begin{aligned} q_1 &= \theta_1 - I_2(\theta_2) \\ v_1 &= w_1 - J_2^{-1}(\theta_2)w_2 \\ q_2 &= \theta_1 \\ v_2 &= w_1, \end{aligned} \tag{7}$$

where $I_2(\theta_2) = \int_0^{\theta_2} J_2^{-1}(s)ds$, and transforming into normal form, one obtains the Pendubot's dynamics in new variables $x = [q_1, v_1, q_2, v_2]^T$:

$$\begin{aligned} \dot{q}_1 &= v_1 \\ \dot{v}_1 &= \alpha v_1^2 + \beta v_1 v_2 + \gamma v_2^2 + \eta \\ \dot{q}_2 &= v_2 \\ \dot{v}_2 &= u. \end{aligned} \tag{8}$$

or in a form of

$$\dot{x} = f(x) + g(x)u, \tag{9}$$

where

$$f(x) = \begin{bmatrix} v_1 \\ \alpha v_1^2 + \beta v_1 v_2 + \gamma v_2^2 + \eta \\ v_2 \\ 0 \end{bmatrix}, \; g(x) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$\alpha(\theta_2) = \frac{a_3 \sin\theta_2}{a_2}$, $\beta(\theta_2) = -\frac{a_3 \sin\theta_2}{2a_2}$, $\gamma(\theta_2) = \frac{a_3^2 \sin\theta_2 \cos\theta_2}{a_2(a_2+a_3\cos\theta_2)}$, $\eta(\theta_1,\theta_2) = \frac{a_5\cos(\theta_1+\theta_2)}{a_2+a_3\cos\theta_2}$.

Finding the control signal u is identical to obtaining such an output function h that maximally linearizes considered dynamics. According to Li et al. (2019), the output function can be selected as:

$$h = q_1 - q_{1R} \tag{10}$$

The control function u becomes

$$u = \frac{1}{L_g L_f^2 h}(-L_f^3 h + X), \tag{11}$$

where

$$X = -(\omega_0^3 h + 3\omega_0^2 L_f h + 3\omega_0 L_f^2 h). \tag{12}$$

and $L_f h$, $L_f^2 h$ are Lie derivatives along vector field $f(x)$.

According to Li et al. (2019) relative degree of h is not well defined around equilibria, thus the equilibria of Eq. (9) are not regular. Moreover, the control signal u explodes to plus or minus infinity. This means that the Pendubot can not be part-linearizable, with a 3-dimensional linear subsystem around an equilibrium point. As a result, one needs to use a different control algorithm around equilibrium, to settle the robots state vector at required upright position. In this article authors used the LQR (*Linear Quadratic Regulator*) controller to settle the robot at the equilibrium pose.
As discussed above, the Pendubot control consists of two regulators. The first one is a swing-up-like type, whose task is to bring the manipulator near the equilibrium point. When the robot is in the error tunnel defined by the δ parameter, it switches to the LQR controller (as in Moysis (2016)). This controller is designed to keep the system at an equilibrium pose. It consists of a state-feedback part together with the K-gain vector. Finally, the control rule for Pendubot can be written in the following form:

$$u = \begin{cases} \frac{1}{L_g L_f^2 h}(-L_f^3 h + X), & \text{for } \gamma(\theta_2) < |\delta| \\ -K(x_r - x), & \text{for } \gamma(\theta_2) > |\delta| \end{cases} \tag{13}$$

and $x = [q_1, v_1, q_2, v_2]^T$ is a state vector a $x_r = [q_{1_r}, v_{1_r}, q_{2_r}, v_{2_r}]^T$ is a reference state vector, ω_0 i $K = [k_1, k_2, k_3, k_4]$ are controllers' gains.

Tuning the controllers' parameters To acquire the best parameter ω_0, the integral of time-weighted absolute error (ITAE, Eq. (14)) performance index was used [Martins (2005)]

$$ITAE = \int_0^\infty t|e(t)|dt, \tag{14}$$

where t is the time and $e(t)$ is the difference between reference value and controlled variable. The best index value coincided with the $\omega_0 = 11$. In the following step the K-vector was obtained, which is the LQR controller's gain. In order to calculate them, the cost function J (Eq. 15), dependent on state and control signal, was defined.

$$J = x^T Q x + u^T R u, \tag{15}$$

where Q and R are the weight matrices for states and inputs, respectively. The optimal K-gains ($K = [1, 520, 1050, 50]$) were obtained with the use of Matlab's `lqr(sys,Q,R)` function, where `sys` is a linearization of Eq. (2) around equilibrium point.

4. SIMULATION RESULTS

This section provides a simulation results of implementing control method (13) to the system in a form of (9). The aim of presented simulations was to stabilize the robot in upright position. Authors also want to verify the convergence area of this new approach, which means that the set of initial conditions are going to be checked to see whether they are appropriate to stabilize the robot under certain boundary conditions. One of them is defined by zero initial angular velocities in the joints. Contrary to appearances this situation is not favorable as there is no any initial energy in the plant at the beginning of the pendulum movement. This causes that the algorithm has to "inject" a lot of energy to move the robot from the initial condition and not to fall at the very beginning. Moreover, the robot is not fixed to the ground and the reaction forces are considered preventing the robot to liftoff.

The algorithm verification is carried out in two ways. Firstly, one wants to know how the algorithm behaves when no restrictions on the motor torque are imposed (Case I). Secondly, the motor torque saturation is imposed on the model (Case II). Thus, the algorithm is being checked whether is capable of controlling the model of physical robot.

The algorithm is not able to stabilize the Pendubot from any initial conditions, with the adopted assumptions (like e.g maximal simulation time). It is interesting to see how far can the Pendubot be moved away from the equilibrium pose, to be stabilized in the upright position. The desired stabilization pose was the upright position for which the angles θ_{1r} and θ_{2r} were equal $\frac{\pi}{2}$ and 0, respectively. To determine the convergence area of analyzed method, a series of tests were conducted. The results are depicted in Fig. **??** for Case I and Fig. **??** for second case. The blue colour indicates the successful trial, while the red one indicates the initial conditions that lead to failure (robot collapse).

Simulations in this paper were conducted with a step size equal 5°, within range of θ_{1_0} from 0 to 180° and θ_{2_0} from −90° to 90°, which gave more than 1300 simulation per approach. These initial conditions mean that the swing-up problem is not considered, i.e. only situations where the robot is above the ground are taken into account. The trial was being taken as successful while the error between actual and desired angle was smaller than $\frac{1}{180}\pi$ rad within simulation time $t = 15$ s.

To see how the algorithm drives the robot to the reference upright position one of the successful trails from Fig **??** and Fig **??** was chosen. The common exemplary successful initial condition was chosen as $\theta_{1_0} = 70°$ and $\theta_{2_0} = 90°$. The obtained angular trajectories are shown in Fig. **??** and Fig. **??** for the first and second approach, respectively. One can observe that the algorithm for Case I drives the joints to theirs reference values about ??? times faster than this

for the Case II (it is obvious, as there is no restriction on the control signal.).

A wide range of robotic application reguire an energy savings or restrictions on its expenditure (through e.g. motors' finite power). As a consequence, the total energetic cost of driving the robot to the upright position was calculated during the simulation. The dynamic effort criterion (17) was used as an indicator of energy used by the algorithm. The effort was defined as

$$T = \int_0^{t_{max}} \tau^2 dt \tag{16}$$

which is the integration of squares of all join torques over time (Abdel-Malek and Arora (2013)). Figure **??** and Fig. **??** depict the motor's torque expended during stabilization process for exemplary initial conditions.

The drawings depicting the total torque consumption during simulation are presented in Fig. **??** and Fig. **??** for the Case I and Case II, respectively.

4.1 Case I

A series of tests were conducted to determine the convergence area. The results are depicted in Fig. 8 for the first case, where there is no saturation on the control signal. The results depicted in Fig **??** are for the case were motor torque is restricted to 6Nm, as in the section 2.1. The blue colour indicates the successful trial, while the red one indicates the initial conditions that lead to failure (robot collapse).

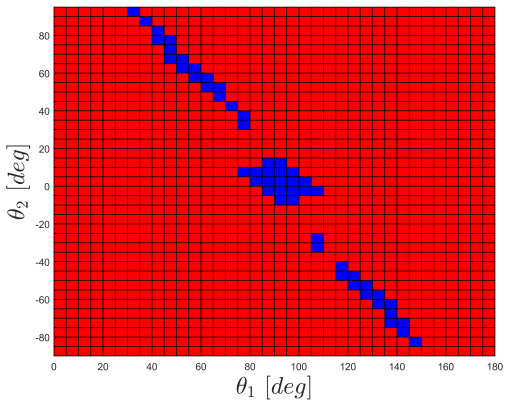

Fig. 3. Convergence area – Case I

The obtained angular trajectories are shown in Fig. 4

Figure 5 show vertical component of ground reaction forces for exemplary successful trial.

Figure 6 depict the motor's torque expended during stabilization process for exemplary initial conditions.

The dynamic effort criterion (17) was used as an indicator of energy used by the algorithm. The effort was defined as

$$T = \int_0^{t_{max}} \tau^2 dt \tag{17}$$

which is the integration of squares of all join torques over time Abdel-Malek and Arora (2013).

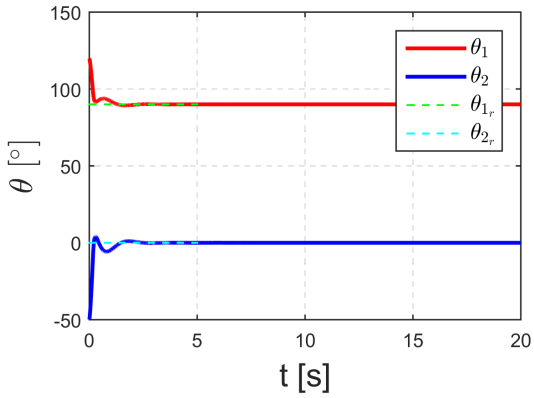

Fig. 4. Angular positions of links – Case ?

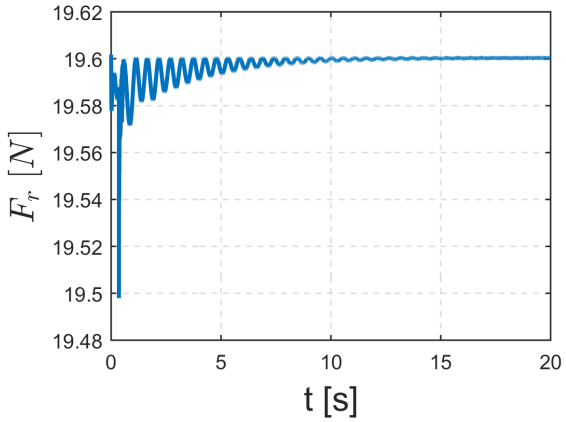

Fig. 5. Ground reaction forces – Case ?

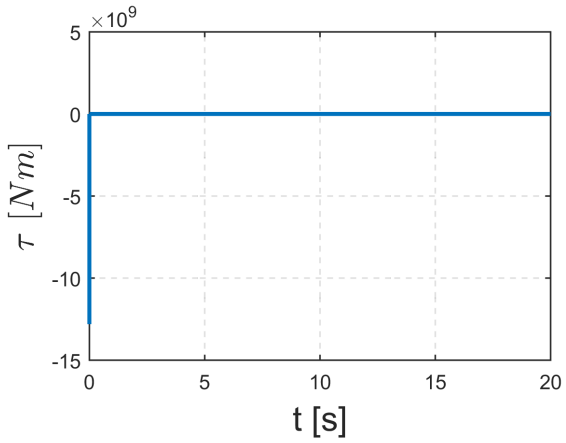

Fig. 6. Motor torque – Case ?

The drawings depicting the total torque consumption during simulation are presented in Fig. 7.

4.2 Case II

Figures for the second case

5. CONCLUSION

The main objective of this paper was to present a simulation results of the new approach of controlling the Pendubot system.

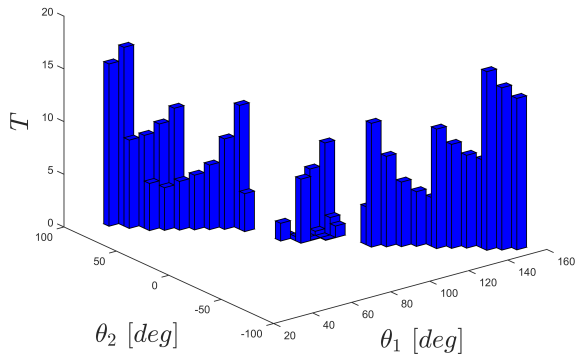

Fig. 7. Energetic cost of successful trials – Case ?

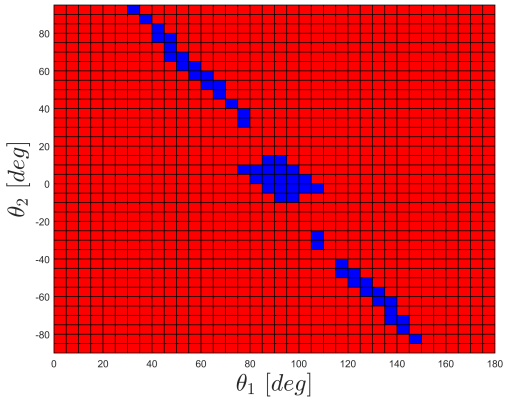

Fig. 8. Convergence area – Case II

ACKNOWLEDGEMENTS

Place acknowledgments here.

REFERENCES

Abdel-Malek, K. and Arora, J. (2013). *Human Motion Simulation: Predictive Dynamics*. Elsevier Inc.

Bartkowiak, P. (2019). *Control of 2 DOF planar underactuated manipulator*. Master's thesis, Poznan University of Technology. In Polish.

Kozłowski, K., Kowalski, M., Michalski, M., and Parulski, P. (2013). Universal multiaxis control system for electric drives. *IEEE Transactions on Industrial Electronics*, 60(2), 691–698.

Kozłowski, K., Pazderski, D., Michalski, M., and Kowalski, M. (2008). Sensory system of four-legged walking robot WR-06. *Elektronika – konstrukcje, technologie, zastosowania*, 188–190. In Polish.

Li, S., Moog, C., and Respondek, W. (2019). Maximal feedback linearization and its internal dynamics with applications to mechanical systems on R^4. *International Journal of Robust and Nonlinear Control*, 29(9), 2639–2659. doi:10.1002/rnc.4507.

Martins, F. (2005). Tuning PID controllers using the ITAE criterion. *International Journal of Engineering Education*, 21.

Moysis, L. (2016). Balancing a double inverted pendulum using optimal control and Laguerre functions. doi: 10.13140/RG.2.1.2948.6486.

Spong, M.W. (1994). Partial feedback linearization of underactuated mechanical systems. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'94)*, volume 1, 314–321 vol.1.

Xin, X. and Kaneda, M. (2004). New analytical results of the energy based swinging up control of the acrobot. In *2004 43rd IEEE Conference on Decision and Control (CDC) (IEEE Cat. No.04CH37601)*, volume 1, 704–709 Vol.1. doi:10.1109/CDC.2004.1428728.