Proceedings of the 2002 IEEE
International Conference on Robotics & Automation
Washington, DC • May 2002

Stabilization of Acrobat Robot in Upright Position on a Horizontal Bar

Masaki Yamakita†‡, **Toshiyasu Yonemura**†, **Yohei Michitsuji*** and **Zhiwei Luo**‡
†Dept. of Mechanical and Control System Eng., Tokyo Inst. of Tech., Japan
*Institute of Industrial Science, University of Tokyo, Japan
‡Bio-Mimetic Control Research Center, RIKEN, Japan
{yamakita,tyone}@ctrl.titech.ac.jp

Abstract - *In this paper, we propose a control algorithm for the problem of stabilization of Acrobat Robot in upright position on a horizontal bar, in an actual experimental environment. The dynamics of the closed loop is designed to match a stable closed loop dynamics around the equilibrium. Moreover, we apply a iterative learning control in order to deal with modeling errors. The validity of the proposed methods will be shown by numerical simulations and experiments.*

1 Introduction

Nonholonomic systems [4] can be controlled with less actuators than the number of its freedom. Acrobot is a simple example of two links planar robot which has a nonholonomic constraint. For this system, Spong [1] proposed a stabilizing method on upright position by using a zero dynamics and a pumping energy function. Nam [2] introduced a canonical form which is derived from the law of conservation of angular momentum, and stabilized Acrobot into an invariant manifold, where angular momentum is zero. However in this method, an approximate linear controller has to be applied repeatedly to stabilize Acrobot on upright position.

A recent report by Stojic [5] showed that the system is globally stabilized in an unstable equilibrium by zeroing a new output function instead of a linear controller. This method can be easily applied to an actual robot, though the switched linear controllers makes control input bigger.

Acrobat Robot used in this study, which we call Super Mechano-Boy (SMB), is like a human and more complex than Acrobat and we modify the control method to our experimental apparatus.

In this paper, we propose to determine parameters used in the control by Stojic so that the closed loop dynamics around the equilibrium (upright position) is identical to a closed loop dynamics controlled by an optimal linear controller. Also we apply a iterative learning control in order to deal with modeling errors using the dynamics of the output function. We will show numerical simulations followed by the experimental results using Acrobat Robot.

2 Acrobat Robot (SMB)

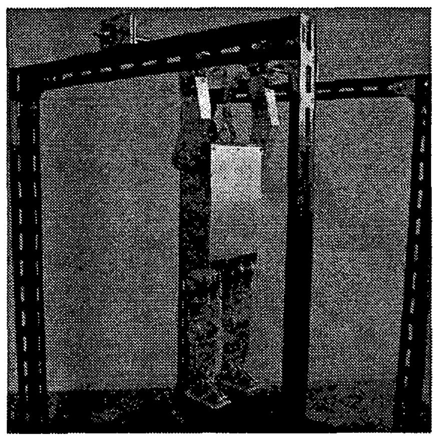


Figure 1: The model of a acrobat robot

Fig. 1 shows Acrobat Robot to be considered in this paper (we call simply this SMB below), which is a human like robot and consists of 9 links, one body, two arms, hams, cruses and foots. All actuators are inside the body, and the arm is connected with actuator by a chain, and legs are controlled by coupled drive mechanism [3]. In our research team, this human like robot is used for various purposes, e.g. dynamic walking, jumping, giant-swing, landing and so on [8]. Though SMB has 9 links, in this paper we constrain the legs with the body mechanically so that SMB can be considered a 2 links model with arm and body as in Fig. 2, and we can deal with the control problem simply. The arm is connected and rotates freely with a horizontal bar and the bar is not actuated so that SMB is an underactuated system. SMB is a nonholonomic system because the acceleration constraint of this system can not be integrable.

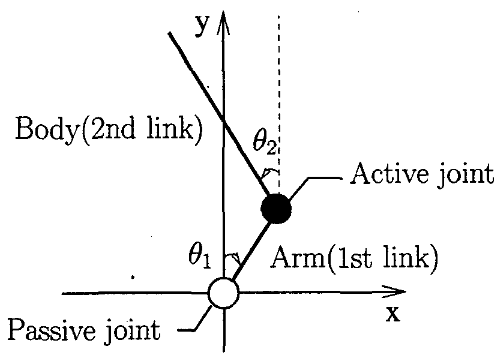


Figure 2: 2 links model for control law

3 Proposed Control Algorithm

3.1 The control strategy

First, we consider a control law to stabilize SMB to the upright position. Usually linear controllers are used for this problem. But such controllers are effective locally. Due to strongly nonlinearity of the system, the system can not be stabilized globally by such simple controllers. Therefore, in this paper we define an output function such that the system is minimum phase and control it to zero.

The dynamic equation of the 2 links model (Fig. 2) is given by

$$\boldsymbol{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \boldsymbol{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + \boldsymbol{G}(\boldsymbol{\theta}) = \boldsymbol{u}, \tag{1}$$

where $\boldsymbol{\theta} = [\theta_1, \theta_2]^T$ is a vector of the absolute joint angles from y axis, $\boldsymbol{M} = [2 \times 2]$ is an inertia matrix, $\boldsymbol{C} = [2 \times 2]$ is a colioris matrix and $\boldsymbol{G} = [2 \times 1]$ is a gravity matrix. u is control input, and u can be represented by

$$\boldsymbol{u} = \begin{bmatrix} -1 \\ 1 \end{bmatrix} \tau, \tag{2}$$

where τ is a control torque at the active joint.

Design of the proposed control algorithm consists of the following 3 steps:

- Definition of the output function

  First, we design the output function given by the angular momentum and an auxiliary function which depends on the angular momentum.

- Zeroing the output function

  We determine a controller which makes the output function to zero. Actually our control law stabilizes the dynamics of the output function.

- Optimization of parameters

  We determine the control parameters in such a way that dynamics of the closed loop system is locally optimal around the equilibrium.

Finally we can get the nonlinear control law to stabilize the system. This is locally optimal around the equilibrium.

We give the detail of the design procedure in the next section.

3.2 Output function

The inertia matrix ($\boldsymbol{M} = [2 \times 2]$) of the system (1) has the following structure

$$\boldsymbol{M}(\boldsymbol{\theta}) = \begin{bmatrix} a_{11} & a_{12}\cos(\theta_1 - \theta_2) \\ a_{12}\cos(\theta_1 - \theta_2) & a_{22} \end{bmatrix}, \tag{3}$$

where a_{11}, a_{12}, a_{22} are constant parameters. The angular momentum L around the horizontal bar is

$$L = J_{\theta_1}\dot{\theta}_1 + J_{\theta_2}\dot{\theta}_2, \tag{4}$$

where θ_1, θ_2 are the angle of links, and $J_{\theta_1}, J_{\theta_2}$ are defined as

$$J_{\theta_1} = a_{11} + a_{12}\cos(\theta_1 - \theta_2) \tag{5}$$
$$J_{\theta_2} = a_{12}\cos(\theta_1 - \theta_2) + a_{22}. \tag{6}$$

Then we define an auxiliary function p [5] as

$$p = \int_0^t \frac{L}{J_p} dt \quad , \quad J_p = (J_{\theta_1} + J_{\theta_2}). \tag{7}$$

Using this function p, the angular momentum can be expressed as

$$L = J_p \dot{p}, \tag{8}$$

and we define the output function y as a summation of L and p

$$y = L + a_1 p = J_p \dot{p} + a_1 p \ , \ (a_1 > 0), \tag{9}$$

where a_1 is a weighting parameter for the function p. If the output function y given here becomes zero, the dynamics of the function p satisfies :

$$J_p \dot{p} + a_1 p = 0. \tag{10}$$

Hence this dynamics is always stable as J_p is positive, p converges to 0. On the other hand, an integral constant of p of (7) is chosen so that SMB is upright on the horizontal bar, when p is zero. As the result, zeroing the output function leads to only one equilibrium of the system (1). Notice that if we choose only the angular momentum as the output function (i.e. $a_1 = 0$), zeroing the output function will not attain the upright position. Let r_g represents the position of center of mass on x axis. Since the manifold is characterized $r_g = 0$, the upright posture of the robot can not be guaranteed.

3.3 Dynamics of output function

The dynamics of the output function is determined as follows. For mechanical systems in (1), derivate of the angular momentum is known to be equal to its gravity term. So the angular momentum of the system of eq. (1) has relative degree 3.

If the auxiliary function p given in (7), is differentiated three times, the term $\ddot{\boldsymbol{\theta}}$ appears as:

$$p^{(3)} = \frac{(\ddot{L} - \ddot{J}_p \dot{p} - \dot{J}_p \ddot{p}) J_p - \dot{L} + \dot{J}_p{}^2 \dot{p}}{J_p^2}, \tag{11}$$

since $\ddot{J}_p$ contains the term $\ddot{\boldsymbol{\theta}}$. Obviously if p is output, the system has also relative degree 3. As a result, our output function y, which is summation of the angular momentum and p, has also a relative degree 3, and we can control the output so that its dynamics satisfies

$$y^{(3)} + a_2 \ddot{y} + a_3 \dot{y} + a_4 y = 0, \tag{12}$$

where a_2, a_3, a_4 are parameters. We need to choose three parameters a_2, a_3, a_4 as well as a_1 in the previous section. A method to determine the parameters is discussed in the next section.

3.4 Locally optimal nonlinear feedback law

In this section, we explain the method how to determine the control parameters, a_1, a_2, a_3, a_4.

Since there is no way to find a set of parameters that is globally optimal, in this paper we design the parameters such that the nonlinear feedback gives an optimal dynamics around the upright position for a criterion function. This way is effective, since the performance of the control law around the upright is very important. Any linear control design method can be applied for design of such locally optimal controller.

The closed loop system (1) is given by an affine system,

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x}) \cdot u_1(\boldsymbol{x}, \boldsymbol{y}) \ , \ \ \boldsymbol{x} = \begin{bmatrix} \boldsymbol{\theta} \\ \dot{\boldsymbol{\theta}} \end{bmatrix}, \tag{13}$$

where $u_1(\boldsymbol{x}, \boldsymbol{y})$ is the output zeroing feedback controller obtained from (12). Then, a linearized system of eq. (13) around the equilibrium is given by

$$\dot{\boldsymbol{x}} = \boldsymbol{A_c x}, \tag{14}$$

where $\boldsymbol{A_c}$ depends on the control parameter a_i. Let assume that a linearized system of the eq. (1) around the equilibrium is

$$\dot{\boldsymbol{x}} = \boldsymbol{Ax} + \boldsymbol{B} u_2, \tag{15}$$

then we can easily design a linear optimal state feedback law, e.g. LQR, H_∞ optimal controllers, as

$$u_2 = -\boldsymbol{Fx}. \tag{16}$$

Thus the closed loop system is given by

$$\dot{\boldsymbol{x}} = (\boldsymbol{A} - \boldsymbol{BF})\boldsymbol{x}. \tag{17}$$

Matching the closed loop dynamics of eq. (14) and eq. (17) around the equilibrium, we can obtain the design parameters a_1, a_2, a_3, a_4 so that coefficient of the characteristic polynomial of eq. (14) are equal to those of eq. (17). Since the number of the coefficients is 4 and the number of design parameters is also 4, the assignment is possible in general.

It is important that this nonlinear controller u_1 can stabilize globally the system (14) because there is no linearlization.

4 Numerical simulation

First we design a LQR optimal controller u_2 (in eq. (16)) in order to calculate parameters a_1, a_2, a_3, a_4. The used matrices for the LQR controller u_2 are as follows :

$$\boldsymbol{Q} = diag(1, 1, 10, 10) \ \ , \ \ \boldsymbol{R} = 1000. \tag{18}$$

Table 1: Notations and numerical settings

m_1	mass of arm	0.52	[kg]
m_2	mass of body	3.17	[kg]
l_1	length of arm	0.18	[m]
l_2	length of body	0.41	[m]
r_{1x}	COG on x of arm	0.11	[m]
r_{1y}	COG on y of arm	-0.04	[m]
r_{2x}	COG on x of body	0.16	[m]
r_{2y}	COG on y of body	0.008	[m]
I_1	inertia moment of arm	1.4×10^{-3}	[kg m^2]
I_2	inertia moment of body	4.4×10^{-2}	[kg m^2]

Thus, the input is $u_2 = -\boldsymbol{F}\boldsymbol{x}$, where $\boldsymbol{F}$ is given by

$$\boldsymbol{F} = [-81.3, -65.7, -14.2, -14.1]. \qquad (19)$$

The design parameters calculated by matching two closed dynamics (14) and (17), are $a_1 = 1.74, a_2 = 37.13, a_3 = 379.05, a_4 = 1304.96$, respectively.

In simulations, we use the 2 links model (recall Section 2), to fit the condition of SMB. Parameters of the model, which are the same as those of real robot, are shown in Table 1. In simulations, the controller is assumed to be implemented as a digital controller where the control interval is 1.0 [msec], and the limit of input is ± 12.0 [Nm], which is also the same as one of the real robot. SMB is controlled from the stable equilibrium $\boldsymbol{\theta} = [\pi, \pi]^T$ by another controller, so we assume that the initial posture is $\boldsymbol{\theta}_0 = [\frac{\pi}{3}, -\frac{\pi}{2}]^T$.

Fig. 3 shows the result of simulations. As in the figure, the output function actually becomes zero and SMB approaches to the upright position. After the output function become zero, the motion to be upright depends on the dynamics of the function p. J_p is automatically given by the model parameters, therefore we can change the motion by arranging the value a_1.

Next, we check how robust this controller is. Here we consider robustness for the region of initial state. Actually the controller introduced in the previous section can stabilize the system globally. But if the input is saturated, it is possible to fail to stabilize the system. We set the initial angle θ_1 and angular velocity $\dot{\theta}_1$ around the initial state in the previous simulation. In Fig. 4, the hatching region shows a region of attractor to the upright position by the controller. Fig. 4 shows that there is a robust enough region for stabilizing SMB around the initial state.

If the poles of the closed loop are assigned to real numbers, we can reduce the vibration of the system around the equilibrium. In next simulations, we assign the poles to two case, one (A) is $[-5, -5, -5, -5]$, another (B) is $[-3+4j, -3-4j, -3+4j, -3-4j]$. To compare the behavior with each poles, we set initial position around the equilibrium $\boldsymbol{\theta} = [0.17, -0.17]^T[rad]$. Fig. 5 shows that the former pole assignment reduces the vibration than that of the latter one.

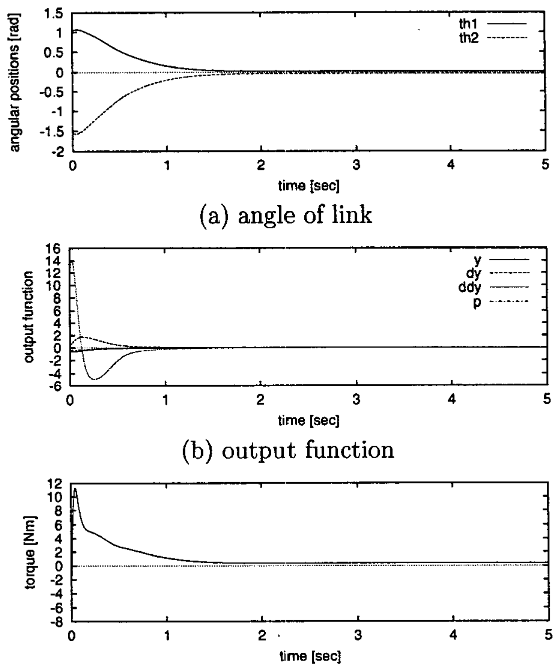


Figure 3: Simulated results I

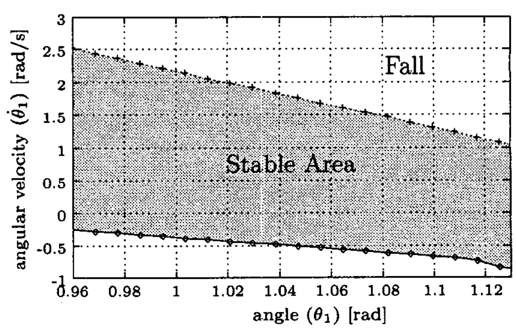
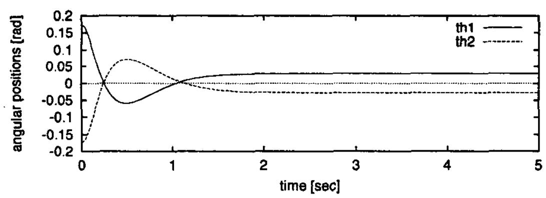


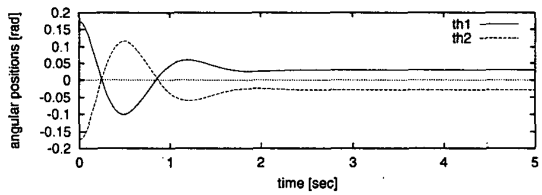
Figure 4: Robustness of the controller(I)

5 Iterative learning Control

If we try to apply the proposed method to an real robot, the modeling errors may become the important

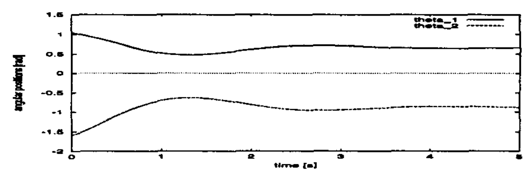


(a) poles assignment (A)

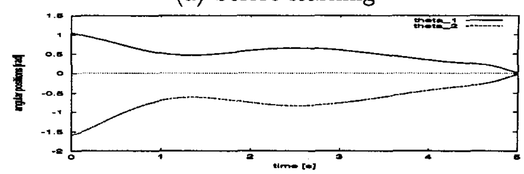


(b) poles assignment (B)

Figure 5: Simulated results II



(a) before learning



(b) after learning (10 times)

Figure 6: Angle of learning control

problem. The modeling error may deteriorate the performance of the zeroing controller. Therefore SMB can not become upright because there are no guarantee that the function p converges to zero. For this problem, we apply an iterative learning control for the dynamics of the output function (12) as,

$$y^{(3)} + a_2\ddot{y} + a_3\dot{y} + a_4 y - u_y = 0, \qquad (20)$$

where u_y is the learning input and this input is updated as a criterion function J decreases:

$$J = \int_0^{t_f} \exp[-W_y(t_f - t)]\; y^2 dt, \qquad (21)$$

where t_f is a control terminal time and W_y is a weighting parameter. The algorithm of updating this input as follow, (see [6] for details)

$$u_y = u_i \quad , \quad u_{i+1} = u_i + \gamma_i z_i, \qquad (22)$$

where γ_i, z_i are given by

$$z_i(t) \;=\; T_p^* y_i \;, \qquad (23)$$

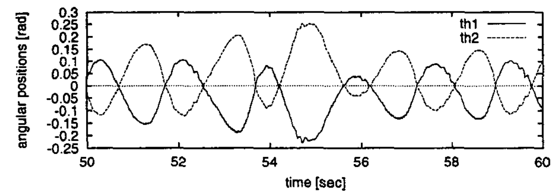
$$\gamma_i \;=\; \frac{\|z_i\|^2}{\|T_p z_i\|}, \qquad (24)$$

where T_p represents input-output relationship of the system (1). The output function y is made zero or very small by the iterative learning control, and the dynamics of the function p is stable for any modeling error because J_p is always positive so that the system can be stabilized at the equilibrium.

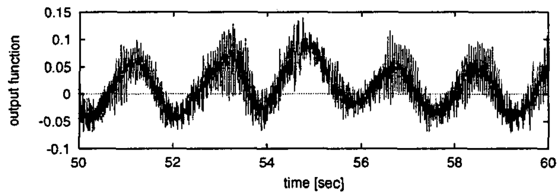
We compare the result with the learning control with that without the learning control. We added the parameter error by 20 % for the position of center of gravity and inertia, which are big modeling errors. We repeated the learning 10 times and the weight parameter was set as $W_y = 100$. Without the learning control, the output function can not become zero due to the modeling error, but after learned 10 times trial, output can be made almost zero.

6 Experimental Result

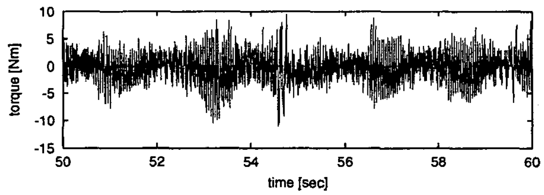
In order to check the validity of the proposed method, we made experiments. For experiments, we used SMB as shown in Fig. 1, which is restricted mechanically as 2 links robot. Each parameter of SMB is the same as the parameter using in the simulations. The arms of SMB are fixed to a horizontal bar and they can turn together freely, since the bar has no actuator, and mimics a non-actuated joint. The angle of the arm θ_1 can be measured from an encoder fixed to the bar. The angular velocity of the arm $\dot{\theta}_1$ is calculated from the angle θ_1. The absolute angle of the body θ_2 is calculated from the angular velocity of the body $\dot{\theta}_2$ which can be measured from a gyroscope. To implement the control law, we used $R_T M_A TX$ [7] and, sampling time was 5 [msec]. In this experiment, the initial position of SMB was set near the upright position and the controlled parameters was determined empirically, $(a_1, a_2, a_3, a_4) = (3, 100, 300, 800)$. We succeeded in stabilizing SMB on the bar until the controller was stopped in 60 [sec]. Fig. 7 shows the result of the experiment in last 10 [sec]. First, we tried to stabilize SMB from the same initial position as simulations (section 4), but the modeling error and uncertainly of input prevented from stabilizing from that initial po-



(a) angle of link (θ_1, θ_2)



(b) output function (y)



(b) torque (τ)

Figure 7: Results of experiment

sition. The iterative learning control as mentioned in section 5 is actually effective for this modeling error. Now, we are trying to experiment with the iterative learning control.

7 Conclusion

In this paper, we proposed a method to design a output function and a control law for an experimental stabilizing control of SMB, which gives an optimal dynamics around the upright position. Moreover, we proposed an iterative learning control for modeling errors and we examined the validity of the control laws by numerical simulations and experiments (partially). Now, we are trying to stabilize SMB experimentally in upright position from the stable equilibrium $\boldsymbol{\theta} = [\pi, \pi]$, by combination proposed control law in this paper with another way. The iterative learning control will be also effective for this combination.

Acknowledgement This work was supported in part by the Grant-in-Aid for COE Research #09CE2004 of the Ministry of Education, Science and Culture, Japan. We would like to especially thank P. Prautsch and N. Kamamichi for their help.

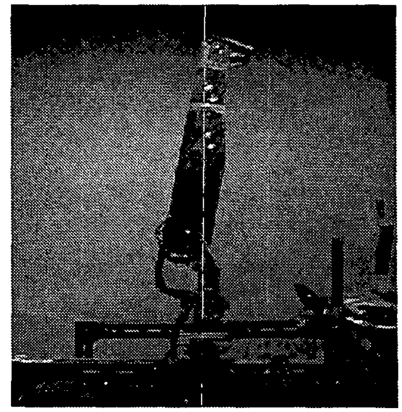


Figure 8: SMB in upright position

References

[1] M. W. Spong "The swing up control problem for the acrobot", *IEEE control systems*, pp. 49-55, 1995.

[2] T. Nam and T. Mita. "Swing-up Control and Singular Problem of a Acrobot Sytem", *Journal of Robotics Society. Japan*, to appear, 2001.

[3] Shigeo Hirose and Mikio Sato. "Coupled Drive of the Multi-DOF Robot", *Proc. ICRA*, pp.1610-1616, 1989.

[4] R. M. Murray, Z. Li and A. S. Sastry. *A Mathematical Introduction to Robotic Manipulation*, CRC Press, 1993.

[5] L. Cambrini, C. Chevallereau, C. H. Moog and R. Stojic. "Stable Trajectory Tracking for Biped Robots", *Proc. of CDC*, pp. 4815-4820, 2000.

[6] M. Yamakita and K. Furuta. "A Design of Learning Control System for Multivariable Systems" *Asia-Pacific Engineering Journal*, Vol. 2, No. 1, pp. 97-118, 1992.

[7] M. Koga. "MaTX/RtMatx: A freeware for intergrated CACSD", *Proc. of Int. Symp. on Computer-Aided Control System Design*, pp. 451-456, 1999.

[8] Y. Michitsuji, H. Sato and M. Yamakita. "Giant Swing via Forward Upward Circling of the Acrobat-robot", *Proc. of ACC*, pp. 3262-3267, 2001.