The Fourteenth International Symposium on Artificial Life and Robotics 2009 (AROB 14th '09),
B-Con Plaza, Beppu, Oita, Japan, February 5 - 7, 2009

Swing Up Control of a 3-DOF Acrobot Using Evolutionary Approach

Ryo Fukushima* and Eiho Uezato**

*Graduate School of Engineering and Science, University of the Ryukyus
**Faculty of Engineering, University of the Ryukyus
1 Senbaru, Nishihara, Okinawa. 903-0213
(Tel:098-895-8638 ; Fax:098-895-3636)
(fukushima@mibai.tec.u-ryukyu.ac.jp)

Abstract: This paper discusses a control method for the acrobot. The acrobot is a model of a gymnast on a horizontal bar with three links and two active joints and a passive joint. This robot is a non-holonomic and underactuated system. We propose a control method for the acrobot where swing-up stage is performed by genetic programming (GP) and balancing stage is handled by a linear quadratic regulator (LQR). Here, GP searches for the optimum input torques for swing up so that the acrobot is able to reach the nearly-desired configuration. The LQR is then switched on to stabilize the system. Simulation results show that the proposed method could control the acrobot effectively.

Keyword: Underactuated robot, Nonlinear system, Genetic programming, Acrobot

1 Introduction

The acrobot [1] is a model of a gymnast on a horizontal bar. This robot is a non-holonomic and underactuated system. Various studies have demonstrated for a two-degree-of-freedom acrobot with an active joint and a passive joint[2], but little has been reported on a three-degree-of-freedom (3-DOF) acrobot[3][4]. The 3-DOF acrobot is studied in order to model more realistic system.

In this paper, we discuss a swing up control method for the 3-DOF acrobot with and two active joints and a passive joint. Swing up control is performed by genetic programming (GP)[5]. GP is an approach expanding from genetic algorithm (GA), and it can widely search for optimum input feedback function for design of the acrobot system. A motion control using GP is discussed in [6].

We propose a control method for the 3–DOF acrobot where swing up stage is performed by GP and balancing stage is handled by a linear quadratic regulator (LQR). It is appropriate to use GP for di cult control problem as the acrobot.

2 Model of the acrobot

Fig. 1 shows the model of the acrobot. m_i $(i = 1, 2, 3)$ and I_i denote mass and moment of inertia, l_i and l_{ci} denote length of link and distance to the center of mass, θ_i is angle, h_i is height to the top of link. Here, u_2 and u_3 are symbolized as actuated torques. The equation of motion of the acrobot system is

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

where,

$$\theta = [\ \theta_1,\ \theta_2,\ \theta_3\ ]^T,\ u = [\ u_2,\ u_3\ ]^T.$$

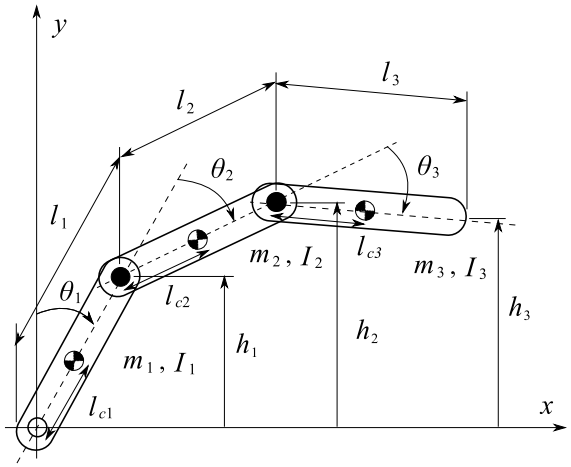

Fig. 1: Model of the acrobot

M is an inertial matrix, C is a coriolis term, G is a gravity term and H is constant matrix.

3 Control System

Fig. 2 shows the block diagram for closed-loop system. Su x d represents desired values. We define deviations as $e = \theta_d - \theta$, $\dot{e} = \dot{\theta}_d - \dot{\theta}$. The control goal is to swing the acrobot up to balancing point ($\theta_d = 0$, $\dot{\theta}_d = 0$). We propose a control method for the acrobot where swing up stage is performed by genetic programming (GP) and balancing stage is handled by a linear quadratic regulator (LQR).

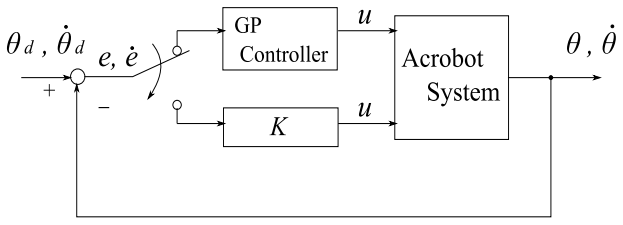

Fig. 2: Block diagram for closed-loop system

3.1 Swing up control of GP

Evolutionary approach is the model of natural selection of genes. As evolution progresses, it is idea of the individuals adapt to a given environmental. GP, which is one of evolutionary approaches, searches for the optimum input torques (feedback input function $u(e, \dot{e})$) for swing up. Each individual of GP represents tree structure which stands for a function. Fig. 3 shows an example of a tree structure. Now, we explain the terminology of GP.

Each elements of tree structure are called a node. The node has a function node and a terminal node. According to the tree structure as shown in Fig. 3, the function nodes are "+", "tanh" and "×", the terminal nodes are "x_1", "x_3", "5.2". The function nodes have a branch, for example "+" has two branch as "tanh" and "×", and "tanh" has one branch as "x_1". The rank of tree structure is called depth. For example, in Fig. 3, depth of tree structure is three. The tree structure shown in Fig. 3 represents function as "$\tanh(x_1)+5.2x_3$". We operate the tree structure with genetic crossover and mutation, so that tree structure adapts to the given environment. The elements for design of GP have the function nodes, the terminal nodes, fitness, parameters (crossover rate, mutation rate, population size), termination condition. We will get the desired feedback input function by the five elements set effectively.

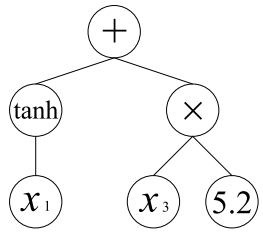

Fig. 3: An example of a tree structure

The following are calculation procedure.

Step 1

Set the number of generation G, and population size is N. Generate a initial population, and evaluate the population with fitness function below.

Step 2

Performs the crossover and mutation operation to the population with mutation rate α.

Step 3

Generate new population up to $N \times \beta$, where, β is crossover rate.

Step4

Evaluate generated population, and individual of the high fitness value is brought down to next generation.

Step5

Until generation reach G, repeat from Step 2. We adopt the excellent individual as the input function of control system.

The fitness function for evaluating the individuals is

$$E = \min_t \; w_1(l_1 - h_1)^2 + w_2(l_1 + l_2 - h_2)^2 + w_3(l_1 + l_2 + l_3 - h_3)^2 + w_4(\dot{\theta}_{r1} - \dot{\theta}_1)^2 + w_5(\dot{\theta}_{r2} - \dot{\theta}_2)^2 + w_6(\dot{\theta}_{r3} - \dot{\theta}_3)^2 \quad , \quad (2)$$

where, t represents time of swing up motion, t_f represents finish time of swing up motion, w_i represents weight coe cient. The lower fitness value, the closer the acrobot locates in the desired position. The "min" in the Eq. (2) is the minimum fitness value of each step from the range of $0 < t \leq t_f$. Finally, this minimum fitness value is of the individual. Here, the first, three term in the Eq. (2) are function considering the highest marks of the each link. For considering height of the each link, the farther distance from the balancing point causes high fitness value. θ_1 and θ_2 are restricted in the range of $-\pi < \theta_{2,3} < \pi$ to limit rotated the link 2 and link 3. If θ_1 and θ_2 exceed the range of limit while searching for the desired input function using GP, we add 10^6 to the fitness value of the individual for penalty.

3.2 Stabilize control at balancing point

Stabilizing control uses the LQR at the near balancing point. If θ_i is su ciently small ($\theta_i \approx 0$), we can approximate $\sin\theta_i \approx \theta_i$, $\cos\theta_i \approx 1$ and neglect $\dot{\theta_i}^2$. Thus, the Eq. (1) is simplified

$$\tilde{M}\theta + \tilde{G}\theta = Hu, \quad (3)$$

and eliminate C which is coriolis term from the Eq. (1). Here, state variable defines $x = [\, \theta_1, \theta_2, \theta_3, \dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3 \,]^T$, and the Eq. (3) is

$$\dot{x} = Ax + Bu. \quad (4)$$

$$A = \begin{matrix} 0_{3\times3} & I_{3\times3} \\ -\tilde{M}^{-1}\tilde{G} & 0_{3\times3} \end{matrix} \quad , \quad B = \begin{matrix} 0_{3\times2} \\ \tilde{M}^{-1}H \end{matrix} \quad .$$

4 Simulation

We carry out simulation as sampling time of 10[ms] and $t_f = 3.0$[s]. Initial values for state variable is $[\ \theta_1, \theta_2, \theta_3, \dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3\ ] = [\ \pi, 0, 0, 0, 0, 0\ ]$. We set the terminal nodes as e, $\dot{e}$ and random real numbers from the range of $[-10, 10]$. The function nodes are shown in Table 1. We hope that the term tanh in the function node will get better result by inhibiting input torque. We search for the individual (input function) which minimize E by a combination of the terminal nodes and the function nodes. Table 2 shows parameters of GP, and Table 3 shows Parameters of the acrobot system.

Using GNU Octave, the LQR controller was designed with weighting matrices

$$\begin{aligned} Q &= \mathrm{diag}(1,1,1,1,1,1), \\ R &= \mathrm{diag}(1,1), \end{aligned}$$

and the state feedback controller $u = -Kx$, where,

$$K = -\begin{bmatrix} 224.39 & 124.15 & 58.06 & 82.26 & 50.04 & 24.15 \\ 171.39 & 98.91 & 40.69 & 62.73 & 39.05 & 17.82 \end{bmatrix}. \quad (5)$$

Table 1: Nodes of function

Function	Number of arg.	Description
$+$	2	arg.1 + arg.2
$-$	2	arg.1 − arg.2
$*$	2	arg.1 × arg.2
tanh	1	tanh(arg.1)

Table 2: Parameters of GP

Parameter	Value
Number of generation G	100
Population N	200
Mutation rate α	0.60
Crossover rate β	0.80

Table 3: Parameters of the acrobot

Parameter	Value	Parameter	Value
m_1 [kg]	0.5	l_{c1} [m]	0.25
m_2 [kg]	0.5	l_{c2} [m]	0.25
m_3 [kg]	1.0	l_{c3} [m]	0.5
l_1 [m]	0.5	I_1 [kgm^2]	0.01
l_2 [m]	0.5	I_2 [kgm^2]	0.01
l_3 [m]	1.0	I_3 [kgm^2]	0.083

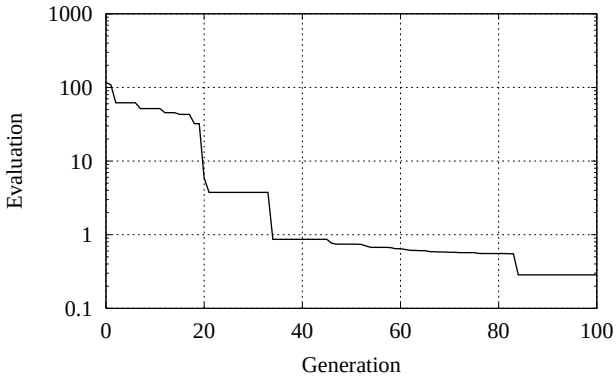

Fig. 4: Evaluation at each generation

5 Results and Discussions

Fig. 4 shows the fitness value of excellent individual at each generation. As the generation proceed, the fitness value tends to the desired position gradually. The fitness function is very important for getting the optimum input torques. Especially, the setting of the weight coe cients of the fitness function affects results strongly, but there is no effective way to determine of the weight coe cients, we have to decide those by trial and error. We evaluate the position of the acrobot considering the highest marks of each link. That is, fitness value become lower at the near balancing point and higher at the far balancing point.

We evaluate the angular velocity of the acrobot considering reaction forces. Each link of the acrobot receives reaction forces from the next link. Thus, if we restrain the angular velocity of the link 2 previously, the angular velocity of the link 1 and link 3 will decrease with decline of the angular velocity of link 2. As a result, the weight coe cients of the fitness function are $w_1 = 40$, $w_2 = 20$, $w_3 = 10$, $w_4 = 1$, $w_5 = 10$, $w_6 = 1$.

Fig. 5 shows successful simulation results of swing up and balancing. θ_1, θ_2 and θ_3 converged on the balance point in about 3.5 seconds. $\dot{\theta}_1$, $\dot{\theta}_2$, $\dot{\theta}_3$, u_2 and u_3 are similar results. A switching time to swing up control from balance control is about 2.15 seconds. The acrobot reaches to the balancing point quickly. We determine the switching time as the step when the lowest fitness value on the simulation.

We obttained the optimum feedback input function using GP. The tree structure of u_2 has number of the nodes of 47 and its depth is 14. The tree structure of u_3 has number of the nodes of 161 and its depth is 27. The input functions are very complex because the depth and number of nodes is large. therefore, we can see that it was very hard to perform swing up control of the 3-DOF acrobot.

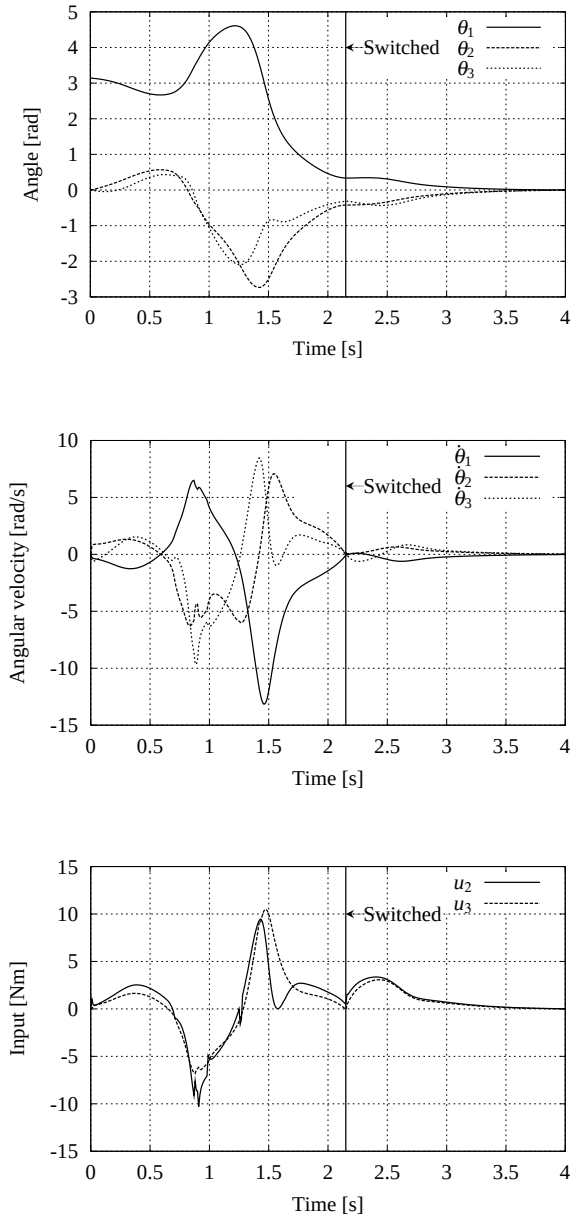

Fig. 5: Simulation results

6 Conclusion

In this paper, we have discussed swing up control for the acrobot. We obtained the optimum input torque using GP. As the simulation results, the acrobot could swing up to the desired position, and the proposed method could control the 3-DOF acrobot effectively.

References

[1] M. W. Spong (1995), The Swing Up Control Problem For The Acrobot. IEEE Control System Magazine, Vol.15, No.2, pp.49–55

[2] K. Kawada, M. Obika, et al (2005), Creating Swing-Up Patterns of a Acrobot Using Evolutionary Computation. Proc. 2005 IEEE Int. Symposium on Computational Intelligence in Robotics and Automation, pp.261–266

[3] X. Xin and M. Kaneda (2007), Swinging–Up Control for a 3–DOF Gymnastic Robot With Passive First Joint: Design and Analysis. IEEE TRANSACTION ON ROBOTICS, Vol.23, No.6, pp.1277–1285

[4] T. Watabe, M. Ymakita and T, Mita (2002), Stabilization of 3 Link Acrobat Robot in Upright Position. Proc. SICE Ann. Conf. 2002 in Osaka (SICE2002), pp.2996–3001

[5] J. Koza (1992), Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press

[6] T. Ogawa, N. Oshiro and H. Kinjo (2008), Backward Movement Control with Two-Trailer Truck System Using Genetic Programming. Proc. of the 13th Int. Symposium on Artificial Life and Robotics, pp.597–600