CONTROLLING OF REAL INVERTED PENDULUM BY FUZZY LOGIC

Nenad Muškinja, Boris Tovornik
Faculty of Electrical Engineering and Computer Science Maribor
University of Maribor
Smetanova 17, 2000 Maribor, Slovenia
nenad.muskinja @ uni-mb.si

Abstract

The basic aim of our work was to swinging up a real pendulum from lower position to upper position and balance in this position in center of course. For this purpose we used fuzzy logic controller with two sets of rules: first for swinging up the pendulum, and second for balancing the pendulum in upper position. Because we controlled the real system and all inputs were noised, we had to use filters. The best results we achieved with combination of fuzzy and state controller.

1. Introduction

Controlling the inverted pendulum is a classic experiment that is used in control laboratories. Futura (1992) has developed minimum time controller that is unfortunately not very robust. Ästrom (1996) has proposed an energy control strategy that controls the energy of inverted pendulum in to the steady-state upright position. This strategy depends critically on the available acceleration of the real inverted pendulum. Our goal was to develop a fuzzy rule based controller with similar behavior. The problem is to swinging up a real pendulum from lower position to upper position and balance in this position in center of course.

2. Process description

The figure 1 shows a laboratory inverted pendulum that is constructed in Laboratory of Process Automation. Figure 2 shows the chart with the potentiometer and pole that can swing in vertical plane. The potentiometer measures the pole angle and in the same time it is used as a pole shaft. The chart is elastically connected to a DC motor (figure 3), where chart position is measured over second potentiometer.

Control algorithm, data acquisition and visualization was realized on PC-486 with AD/DA converter (*PCL 711*) and fuzzy processor module *OMRON FB-30AT*. All program was written in C code with *Borland C++*.

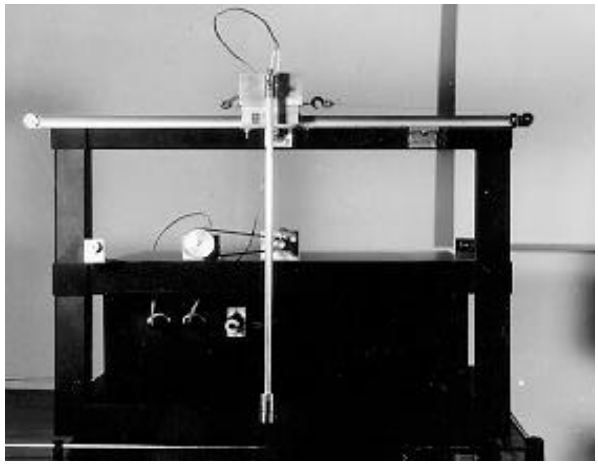

Figure 1: Inverted pendulum construction

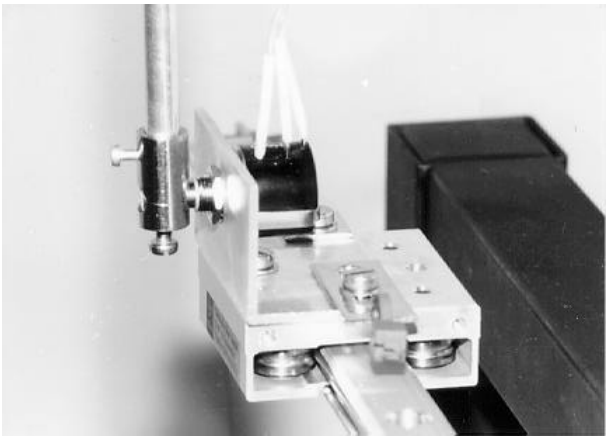

Figure 2: Chart with potentiometer and pole

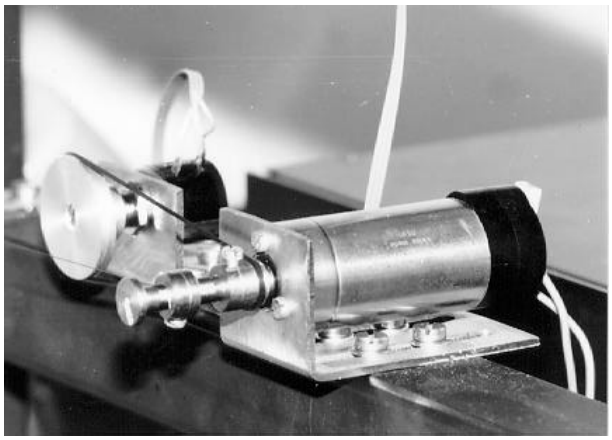

Figure 3: DC motor and potentiometer for chart position measurement

3. Swinging up of pendulum

From basic position after small impulse the pendulum should swing up to upper position without overstepping of limited course of motion. The principle of swinging up was (while an angle is small) based on increasing of energy, it means:

a) For small angles $|a| < 89^\circ$:

- if angle a is positive (negative) and if angle velocity da is positive (negative) than the action value is positive (negative),
- if angle a and angle velocity da has different sign than the action value is equal 0.

b) For medium angles $\pm(88^\circ;145^\circ)$ the action value is equal 0.

c) For big angles close to the upper position $\pm(145^\circ;180^\circ)$:

- if the angle velocity is small, $|da| < 1{,}47/s$ and it has same sign like angle a than energy must be increased like in case a),
- if angle velocity is big, $|da| > 1{,}47/s$ and it has the same sign like angle a than we break the swinging.

All this conditions respect position of pendulum *x* to not overstep limits.

The control algorithm of increasing the energy of inverted pendulum was realized with fuzzy logic system. We use three inputs variables (a ,da in x) and one output variable *u*. Input variable a has six membership functions symmetrical by center.

. In case a) we don't need to know accurate value of angle a , so we used rectangle shape. Definition ranges of variables are:

chart position: $x \in [-16\text{cm},16\text{cm}]$,
pole angle: $a \in [-3.14\text{rad},3.14\text{rad}]$,
angle velocity: $da \in [-6\text{rad}/s, 6\text{rad}/s]$ and
DC motor voltage: $u \in [-10\text{V},10\text{V}]$

All this values are calculated in to the range 1÷4095 in C program because of 12 bit unsigned logic fuzzy processor board. Figure 4 shows all membership functions.

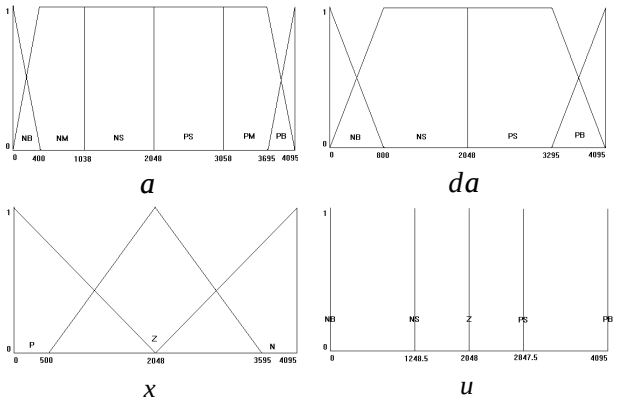

Figure 4: Membership functions for swinging algorithm

4. Balance of pendulum

Initial for this algorithm is upper position, in that case it is ($a = 0, da = 0, x = 0 \text{in} dx = 0$). Our aim was to keep the pendulum in or close upper position during activity of defects. Since we should control the position of the pendulum, we had to use four input variables (a ,da ,x and dx). We used membership functions with triangle and trapezoidal shape. Because the balancing is very sensitive we used five membership functions for angle a and angle velocity da and three membership functions for chart position *x* and chart velocity *dx*. We compensated:

- chart position *x*, if angle a and angle velocity da were small and
- angle a , if angle a and angle velocity da were big

We determine the rules with respect to position of the pendulum. In order to use all rules in table we adjusted ranges for:

chart position: $x \in [-16\text{cm},16\text{cm}]$,
chart velocity: $dx \in [-0.5\text{m/s},0.5\text{m/s}]$,
pole angle: $a \in [-0.2,0.2]$,

pole angle velocity $da \in [-0.5/s, 0.5/s]$ and
DC motor voltage: $u \in [-10\text{V}, 10\text{V}]$

Output membership functions were seven discrete values. The small values ± 2.44V we used for tuning of position and medium and big (±4.88V, ±10V) for compensation of big deviations and also for inertia.

Figure 5 shows membership functions for balance algorithm.

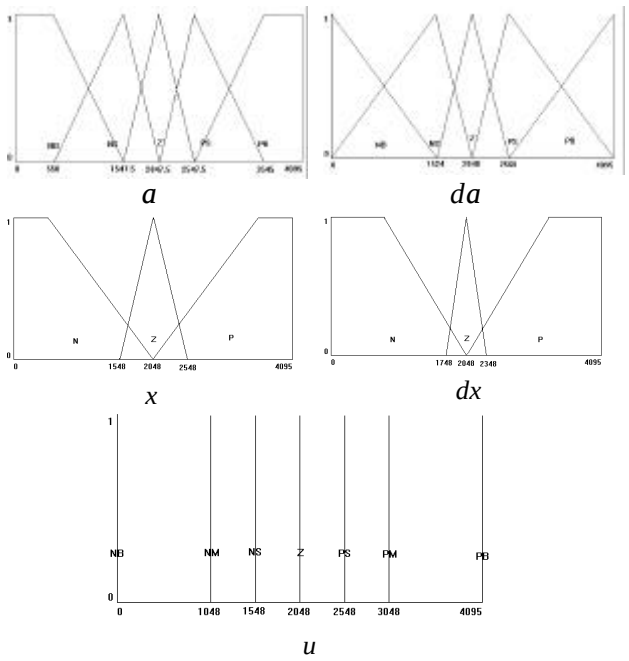

Figure 5: Membership functions for balance algorithm

5. Filtration and friction compensation

Because we controlled the real system and all inputs were noised, we had to use filters. Pole angle a and chart position x was measured over potentiometer voltage. All other variables must be calculated. We filtered pole angle a, angle velocity da and chart velocity dx by 1-st order filter. Variables da and dx were computed from measured values a and x.

Computation and filtration of chart velocity dx:

$$\begin{aligned} dx &= \frac{s}{T_{dd}s+1}x \\ dxT_{dd}s + dx &= sx \\ T_{dd}[dx(k)-dx(k-1)] + T_s dx(k) &= x(k)-x(k-1) \\ dx(k) &= \frac{1}{T_{dd}+T_s}[x(k)-x(k-1)+T_{dd}dx(k-1)] \end{aligned} \quad (1)$$

Filtration of pole angle velocity da :

$$da(k) = \frac{1}{T_d+T_s}[a(k)-a(k-1)+T_d da(k-1)] \quad (2)$$

Computing and filtration of pole angle a :

$$\begin{aligned} \frac{a_f}{a} &= \frac{1}{T_f s+1} \\ a_f[T_f s+1] &= a \\ T_f\frac{a_f(k)-a_f(k-1)}{T_s} + a_f(k) &= a(k) \\ T_f a_f(k) - T_f a_f(k-1) + T_s a_f(k) &= T_s a(k) \\ a_f(k)(T_f+T_s) &= T_s a(k) + T_f a_f(k-1) \\ a_f(k) &= \frac{T_s a(k) + T_f a_f(k-1)}{T_f+T_s} \end{aligned} \quad (3)$$

The sampling period was $T_s = 0.003\text{s}$, filtration constant for derivation of chart position was $T_{dd} = 0.03\text{s}$, filtration constant for angle deviation was $T_d = 0.04\text{s}$ filtration constant for angle was $T_f = 0.005\text{s}$.

A friction caused that output voltage for DC motor between ± 0.3V had no effect. Therefore we add output nonlinear element shown on figure 6.

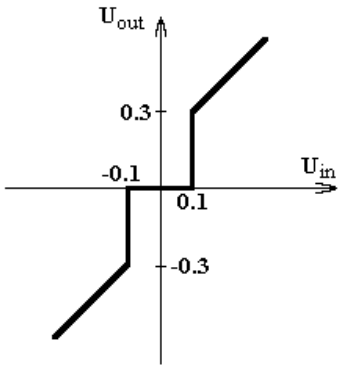

Figure 6: Output non-linear element for friction compensation

This element also solves the problems with oscillations close to zero pole angles.

6. Switching between swinging and balance algorithm

a) *Fuzzy switching algorithm with state controller of balance*

We switched between algorithms if angle $|a| > 3.1$.

In this moment we switched the absolute zero of pole angle a to zero value as it has in upper position. The swinging value was chosen with respect to pendulum inertia.

b) *Both fuzzy algorithms*

The switching was the same like in case a). Because each algorithm uses different shapes of membership function, we needed seven inputs for fuzzy processor board, two tables of rules and two outputs. In the switching moment the table of rules is switched and other output for calculating of action value is used.

7. Experiment results

All experiments were don in real time on laboratory inverted pendulum shown on figures 1,2 and 3. For swinging up the pendulum we used the algorithm from section 3. Figure 7 shows that the pendulum swings up in only two swings. This for the real inverted pendulum is near of the optimum.

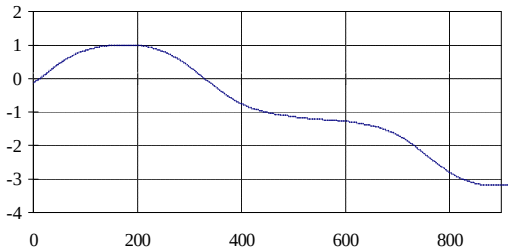

Figure 7: Pole angle $a[\text{rad}]$ in swinging up the inverted pendulum

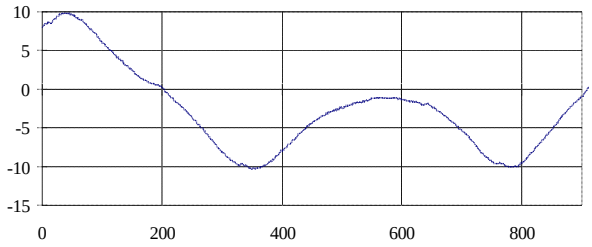

Figure 8: DC motor voltage $u[\text{V}]$ in swinging up the inverted pendulum

Balancing the pendulum was controlled with fuzzy state controller. Figures 9 and 10 shows fuzzy logic algorithms. For small pole angle deviation the controller was successful, but for big deviations from upper right position we achieved better results with linear state controller.

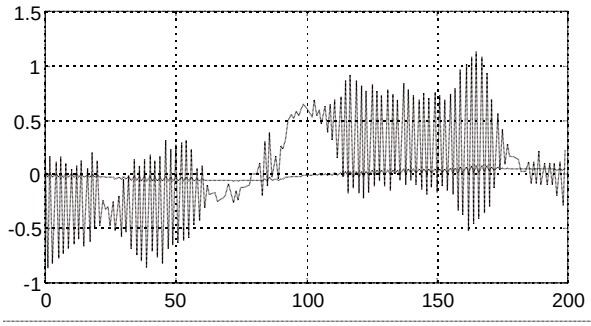

Figure 9: Pole angle and pole angle velocity in balancing the inverted pendulum

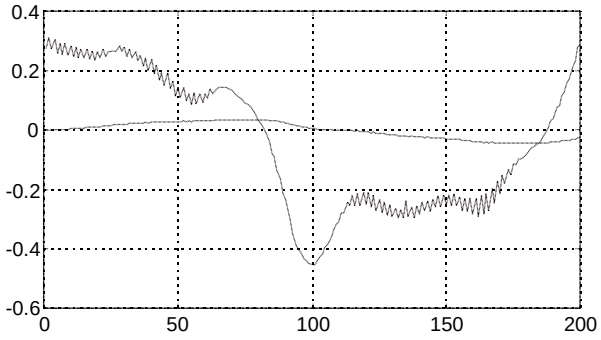

Figure 10: Chart position and chart velocity in balancing the inverted pendulum

8. Conclusion

The best result we achieved with swinging algorithm. These algorithms don't need so big initial impulse unlike energy swinging algorithm. It's also closer to time optimal swinging. Our fuzzy algorithm for balance control is not able to compensate bigger deviations unlike state controller. This is the reason why it isn't often able to stabilize pendulum if velocity is big in moment of switching. The best results we achieved with combination of fuzzy and state controller. Because setting of zero pole angles has big influence to stability of system, inaccurate setting of zero was the most frequent cause of desirability after switching of algorithms.

To improve the inverted pendulum control we have some recommendations:

- increase the number of membership functions of *dx*, because we found out the big influence on system,
- improve the sensibility of an pole angle sensor,
- take off sliding during transfer between a motor and a position sensor and
- include to fuzzy rules influence of motor moment to derivation of position by various angle.

Literature

[1] Furuta, M. Yamakita, and S. Kobayashi, Swinging up control of inverted pendulum using pseudo-state feedback. J. Systems and Control Eng. 206, pp. 263-269, 1992.

[2] K. J. Ästrom, and K. Furuta, Swinging up a pendulum by energy control. Proc. 13th IFAC Triennial World Congres, San Francisco, USA, 1996.

[3] M. Kabat and R. Ondrejkova, Project report: Inverted pendulum, Process Control Laboratory, Faculty of Electrical Engineering and Computer Science, Maribor, 1997.