SIMULATION, CONTROL DESIGN AND EXPERIMENTAL TESTING OF UNSTABLE MOBILE DEVICE

Švejda, P., Grepl, R., Kratochvíl, C., Vlachý, D., Zezula, P.

Laboratory of mechatronics, FME, Brno University of Technology

Abstract

This paper deals with modeling, simulations and experimental testing of inverted pendulum-like robotic mobile device with coaxial wheel arrangement. Design of simulation model, its verification and control design by using MATLAB environment will be mentioned. Further, an assemblage of a real laboratory model and realtime control from MATLAB Real Time Toolbox is introduced.

1 Introduction

This paper briefly introduces a robot named Keywatko which began as a student project. The Keywatko is two-wheeled unstable inverted pendulum-like robot with koaxial wheel arrangement. At the beginning it was inspired by two interesting projects:

nBot This robot is perhaps best known inverted pendulum-like robotic device and was built up by David P. Anderson in a home working room. The nBot uses for getting state informations a home-made wheel revolution sensor and for tilt angle sensing an accelerometer ADXL202 and piezoelectric rate-gyro. For getting better values of sensor signals is used complementary filter.

Joe le pendule This robot was built in EPFL laboratories in Switzerland as a size-reduced prototype of a real transportation device. Control system consists from two decoupled state space controllers and is implemented directly on the body of robot in DSP.

Elementary idea of such balancing robots is easy - to steer wheels in the direction of the body falling. In other words the aim is to hold up the center of gravity of whole robot's body above wheels. In practise it means two sensor groups for tilt angle and for angular position measuring. Robot doesn't have onboard chip for data evaluation and for feedback control. It contains only electronics for motor control, sensors and sensoric data collect boar which is used as interface between robot and MF624 I/O multifunction board. Algorithms for robot control and data evaluation is implemented in the MATLAB environment where are used the Real Time Toolbox with common Simulink tools for direct cooperation with robot.

2 Model building in SimMechanics

For building up a model of robot we used MATLAB SimMechanics toolbox. We supposed that in "revolute" block actuates gear torque to a wheel and robot's body too, but in an oposite direction. Further there was a damping implemented in a joint wheel-body, which is adequate to angular rate in pivot. Damping which substitute rolling resistence of wheel was implemented too. From easier modeling purpose there is a reduction from rotational to translation movement.

If we want to use this model for design control we should modify it a bit by using a MATLAB command `linmod`. This command extracts Simulink model to the MATLAB environment and create linearized state space model of the Simulink subsystem. Working point can be set explicitly otherwise it is considered as a zero. Before use of `linmod` we had to remove `Continuous Angle` block, which contains integrator and would get an extra state to a linearized system.

After we choose In signal (torque) and Out signals $y = [\varphi \ \ x]$. Then we can get a linearized SimMechanics model and consequently make its equivalent in the MATLAB environment.

```
[A,B,C,D] = linmod('keywatko_for_linearization_SimMech');
Plant     = ss(A,B,C,D);
```

To get state space vectors – state vector, input and output vector, we use sim command:

```
[t,x,y]     = sim('keywatko_for_linearization_SimMech')
StateVector = mech_gate_states(x(end,:),...
 'keywatko_for_linearization_SimMech/Plant/Machine Environment [0 0 -9.81]')
StateVector.StateNames
```

We have got SimMechanics printout in the MATLAB environment which is represented by command mech_gate_states. Acording the printout of the command we can formulate the variables of state space model:

$$\begin{aligned} \mathbf{x} &= [\varphi \quad x \quad \dot{\varphi} \quad \dot{x}]^T \\ \mathbf{y} &= [\varphi \quad x]^T \end{aligned}$$

2.1 Analytical model building

An analytical mathematical model was derivated by using Lagrange method of second kind. It was used for verification and easy Simulink model building as a second approach to the solved problem. By using Simulink model assembled from derivated analytical model was designed first feedback control via PD regulator by witch we could validate the model and simulate the basic behaviour of simulated problem.

3 LQR design control

In terms of previously builded state space model we designed LQR control of robot. At the first it is important to check the observability and controlability of the states of system. This was provided by using commands ctrb and obsv which generates appropriate matrixes. In our case, the system is fully controlable and observable.

Then was stepped to lqr design control. Since the lqr controll process is optimal from mathematical view it doesn't have to be optimal from user's point of view because it could have enhanced energetical requirements. Since these requirements could be unreachable at the moment it is possible to set the compromise between quality of regulation and requirements on the signal by seting a penalization matrix. It's values was set to

```
Q = [10 0 0 0; 0 500 0 0; 0 0 0 0; 0 0 0 0]
```

and value of control matrix was chosen for simplicity to $\mathbf{R} = [1]$. Now the optimal gain matrix can be evaluated:

```
K_lqr = lqr(A,B,Q,R);
```

and new closed system with feedback and gain matrix can be created $\mathbf{K}$:

```
Ac = [(A-B*K_lqr)];
Bc = [B];
Cc = [C];
Dc = [D];
sysCtrl_lqr = ss(Ac, Bc, Cc, Dc);
```

Because full-state feedback systems don't compare their outputs with reference the steady state output doesn't have desired values. This must be achieved by correction factor where desired output part is chosen to be balanced. Then a state reprezentation can be created by using command `ss`. By using `rscale` command we obtain a correction factor, which can be used for this final tune of controller. Then we built the resulting system by using `ss` command.

4 Electronics

Keywatko is driven from MATLAB via multifunction I/O card MF624, which provides an interface between MATLAB Real Time Toolbox and the robot itself. MF624 offers outputs and inputs of an analog and digital signals, timers and counters, inputs for encoders, etc. Signals is sent to or received through the card between robot and MATLAB. Robot electronics is consists from module of motor control and sensor's modules sensing tilt angle of the robot and a steering angle. Robot uses only single motor drive which means movements only in direction outgoing from pitch axis.

4.1 Servo modification

While robot was built, we used only one actuator – an old Hitec servo HS-475HB. Servo contains a small DC motor, which is in common use in audiotechnics, gear unit and an electronics providing an inner servo logic. Electronics consists form comparator and a feedback unit, which is presented by a potentiometer connected to a gear of servo. When servo receive control signal the inner logic analysis an actual angular position and begin to turn with gear shaft. Received signal is constantly compared with feedback signal until these both are equal. Then movement is stopped. Further there was a mechanical stopper in the box of servo. Both, electronics and stopper were removed from the servo. Finally, there is only motor with gear unit which can rotate without angular limits.

As actuators for KeywatkoII were chosen two "robotic" servos HSR-5995TG, which offer larger torque. After the same modification as before are used as common motor that have sufficient torque on a low weight.

4.2 Motor driving

Direction and speed of motor movement is controlled by command blocks in MATLAB Real Time Toolbox, which cooperates with MF624 board where hardware support generated output signals is implemeted. This toolbox with MF624 card allows to connect an outer hardware device into the "hardware in the loop" configuration. The outer device then uses computational power of a desktop computer and can be controlled without undesirable model simplifications. For motor control are used digital signals for enabling motor function and rotary direction command. Further PWM signal is generated by using Real Time Toolbox and MF624 hardware support.

The signals mentioned previously are send to the motor driver unit board which contains two main parts. A full bridge driver L6203 is used for power performance and a current regulator

L6506 is used as a logic part processing the signals from MF624 board. The bridge driver allows current consuption up to 4 A which is more than sufficient condition for our purposes. Motor can be controlled only by this IC, but we used a recommend make from L6203 datasheet for bidirectionl DC motor controll with L6506 IC where only a few changes were made. We can comfortably control the movement direction of motor from MATLAB Simulink now and separately the angular velocity which is given by PWM signal from MF624 PWM generator.

4.3 Sensoric part

For measuring of angular position of wheel was used a two channel incremental encoder Sharp GP1A71R. This encoder has resolutuion 120 cycles per revolution and gives information about direction of movement. It was used only in the first version of robot Keywatko. New version of Keywatko uses two incremental encoders HEDS-9100 which is two channel too but gives more accurate information about angular position.

The tilt angle is measured by 3-axis accelerometer LIS3L02AS4 placed in the centre of rotation of robot's body and only one axis for tilt sensing is used. The signal from accelerometr is filtred by cumulative filter. Sensoric signals are send to MF624 board and consequently processed by algorithm programed in MATLAB and Real Time Toolbox.

5 Conclusion

The paper briefly described project of two-wheeled unstable inverted pendulum-like robot. Although it was focused on the first version of the robot most of the outputs of the work were used while next generation of robot was designed. KeywatkoII was improved from the sensoric and feedback control point of view and testing is still in progress.

Acknowledgement

Published results were acquired using the subsidization of the project AV0Z20760514 and GACR 101/06/0063.

References

[1] Böhm, J., Prokop, R., Fessl, J.: *Praktické aspekty samočinně se nastavujících regulátorů: algoritmy a implementace*, VUTIUM, 1999

[2] HUMUSOFT: *Real Time Toolbox for use with Simulink, version 4.0 – User's Manual*, , 2006

[3] Grepl, R.: *Modelování mechatronických systémů v* MATLAB–*SimMechanics*, BEN - technická literatura, 2007

[4] Grasser, Arrigo, Colombi, Rufer: *JOE: A Mobile, Inverted Pendulum*, Proceedings of the IASTED International Conference on Modelling, Identification and Control, Innsbruck, Austria, 2001

[5] nBot web: *http://geology.heroy.smu.edu/ dpa-www/robo/nbot/*, 2007-05,

Pavel Švejda, Robert Grepl, Ctirad Kratochvíl, David Vlachý, Pavel Zezula
Laboratory of mechatronics, FME Brno University of Technology
e-mail: `ysvejd00@stud.fme.vutbr.cz`