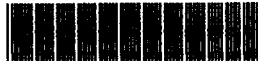


raf

TJ223.P55.E74 2009.



0000065834

PID of an inverted pendulum / Eric Chee Sai Hoo.



UNIVERSITI TEKNIKAL MALAYSIA MELAKA

FAKULTI KEJURUTERAAN ELEKTRIK

PROJEK SARJANA MUDA

(PSM 2)

BEKU 4983


PID OF AN INVERTED PENDULUM

ERIC CHEE SAI HOO

B010510002

May 2009

“I hereby declare that I have read through this report and found that it has comply the partial fulfillment for awarding the degree of Bachelor of Electrical Engineering (Control, Instrumentation and Automation)”

Signature : 

Supervisor's name : En. Syed Najib bin Syed Salim

Moderator's name : Shahrieel bin Mohd Aris

Date : 13 May 2009

PID OF AN INVERTED PENDULUM


ERIC CHEE SAI HOO

**This Report Is Submitted In Partial Fulfillment Of Requirements For The
Degree of Bachelor In Electrical Engineering (Control, Instrumentation and
Automation)**

**Fakulti Kejuruteraan Elektrik
Universiti Teknikal Malaysia Melaka**

May 2009

"I hereby declared that this report is a result of my own work except for the excerpts that have been cited clearly in the references."

Signature : 

Name : ERIC CHEE SAI HOO

DATE : 13 May 2009

To my dearly loved father and mother

To all teachers and supervisor

ACKNOWLEDGEMENT

I wish to express my deepest gratitude to my supervisor, Mr. Syed Najib bin Syed Salim for guiding me in this project. This success of this project is highly influenced by his information, suggestions and ideas.

Also, I wish to thank my parent who is very supportive to me. Without them, this project would not be completed.

I also would like to thank my friends who had helped me and to those either directly nor indirectly had helped me.

ABSTRACT

Inverted pendulum has been a popular platform for control engineers to research and test their controller. Controller for an inverted pendulum is widely ranged; from Proportional-Integral-Derivative (PID) to Fuzzy Logic, to Fuzzy PID and etceteras. This project will implement a PID controller to an inverted pendulum. Controller implementation can be done in both hardware and software. A software controller is selected in this project and is written with Microsoft Visual C#. Communication between software (computer) and inverted pendulum hardware (motor and rotary encoder) linked with serial cables and a decoder/driver. Microprocessor (PIC) from MICROCHIP is used to build the decoder/driver and its function limited to passing data. For software controller, it is equipped with a graph plotting tool to monitor the position of the inverted pendulum's pole. A simulation is done using Matlab to simulate responses of the balancing system. Outcomes from few experiments, the observations clearly show that the inverted pendulum is able to be controlled and balanced. Results from conventional PID and cascade PID are analyzed and suggestions are given so that in future, better results could be obtained.

ABSTRAK

Pendulum songsang merupakan perkakasan yang selalu digunakan oleh jurutera kawalan untuk penyelidikan dan ujian terhadap sesuatu pengawal. Pengawal untuk pendulum songsang terdiri daripada pelbagai jenis iaitu dari Pengawal Perkadaran-Pengamiran-Kerbezaan atau PID kepada Logik Samar atau 'Fuzzy Logic', seterusnya gabungan Pengawal Perkadaran-Pengamiran-Kerbezaan dan logik samar atau 'Fuzzy PID' dan sebagainya. Bagi projek ini ianya dilaksanakan menggunakan pengawal 'PID'. Pelaksanaan pengawal ini melibatkan perkakasan dan perisian. Perisian computer yang dipilih dalam projek ini ditulis dengan menggunakan perisian 'Microsoft Visual C#'. Komunikasi di antara perisian (komputer) dan pendulum songsang adalah melalui kabel RS-232 serta penyahkod/pengerak. Mikro-pemproses daripada MICROCHIP digunakan untuk membina litar penyahkod yang mana fungsinya terhad kepada penghantaran data. Perisian untuk pengawal juga disertakan bersama ruangan graf yang boleh memaparkan kedudukan sudut pendulum songsang tersebut. Simulasi pula akan dibuat menggunakan MATLAB untuk melihat sambutan keluaran merujuk kepada sistemimbangan. Hasil yang diperolehi daripada beberapa eksperimen yang dibuat akan dinyatakan dengan jelas menerusi sambutan keluaran yang diperolehi yang mana melalui pemerhatian yang buat menunjukkan bahawa ianya mampu mengawal kedudukan pendulum songsang serta dapat mengimbangnya. Keputusan yang diperolehi merujuk kepada dua konfigurasi iaitu 'conventional PID' dan 'cascade PID' akan dianalisa dan cadangan penambahbaikan turut dinyatakan didalam laporan ini.

CONTENT

ACKNOWLEDGEMENT	iv
ABSTRACT	v
ABSTRAK	vi
LIST OF TABLE	x
LIST OF FIGURE	xii
LIST OF SYMBOLS AND TERMS	xvii
LIST OF APPENDIX	xix
INTRODUCTION	1
1.1 Background	1
1.2 Concept	2
1.3 Problem Statement	3
1.4 Project Objective	4
1.5 Project Scope	4
1.6 Outline	6
LITERATURE REVIEW	7
2.1 Journal	7
2.1.1 An Integration Design Approach in PID Controller by Jen-Yang Chen [4]	8
2.1.2 A General Practical Design Method for Fuzzy PID Control from Conventional PID Control by Yongli Hua and Seiji Yasunobu[5]	9
2.1.3 A Proportional Integral Derivative Feedback Control without a Subsidiary Speed Loop by M. Aboelhassan[6]	12

2.1.4	Journal Comparison	14
2.2	Previous Project	20
2.3	Contribution	21
2.4	Proportional-Integral-Derivative	21
2.4.1	Tuning PID	25
2.4.1.1	Ziegler-Nichols	27
2.4.1.1.1	Open loop tuning	28
2.4.1.1.2	Closed loop tuning	28
2.4.1.2	Manual tuning	29
2.4.2	Anti-Windup Controller	30
2.4.3	Cascade Configuration	33
2.4.4	PID Conclusion	34
2.5	Theory of motor modeling	35
	METHODOLOGY	39
3.1	Methodology Overview	40
3.2	Simulation	42
3.3	Hardware procurement	44
3.4	Decoder Circuit	44
3.5	Motor driver	47
3.6	PC-based PID	50
3.7	Knowledge base	51
3.8	Inverted pendulum mechanical structure	53
3.8.1	Motor modeling	53
3.8.2	Sensor	55
3.8.3	Mechanical Drawing	56
3.9	Software and Hardware Integration	56
3.10	Troubleshooting	56

RESULTS AND DISCUSSIONS	57
4.1 Motor modeling	57
4.2 Conventional PID	60
4.3 Cascade PID	69
4.4 Fuzzy Logic	88
4.5 Disturbances towards controllers	91
4.6 Evaluation	99
4.7 Final results	104
4.8 Issues	104
CONCLUSION	106
REFERENCE	108
APPENDIX A	111
APPENDIX B	114
APPENDIX C	139
APPENDIX D	142
APPENDIX E	143
APPENDIX F	145

LIST OF TABLE

No	Title	Page
2.0	Comparison between journals	14
2.1	Comparison between previous project and current project	20
2.2	Ziegler- Nichols tuning method	28
2.3	Effects of increasing parameters	29
2.4	Advantages and disadvantages of available tuning method	34
3.0	Comparison between 16F877A and 18F4331	46
3.1	Comparison between programming languages	50
3.2	Sensors comparison	55
4.0	Speed(rpm) vs input voltage of M586TE	57
4.1	Current(A) vs input voltage of M586TE	58
4.2	Tachometer voltage(V) vs input voltage	58
4.3	Summary of speed(rpm), current(A) and input voltage(V)	58
4.4	M586TE DC Motor Specification	59
4.5	Comparison of parameters between simulation and real time results for PID, cascade PI & P and Fuzzy Logic controller	100

4.6	SSE, OS%, Tr, Tp and Ts for PI, PID, cascade PI and P and cascade PID and P for 90 degree amplitude disturbance	101
4.7	SSE, OS%, Tr, Tp and Ts for PI, PID, cascade PI and P and cascade PID and P for 180 degree amplitude disturbance	101
4.8	SSE, OS%, Tr, Tp and Ts for PI, PID, cascade PI and P and cascade PID and P for -90 degree amplitude disturbance	102
4.9	SSE, OS%, Tr, Tp and Ts for PI, PID, cascade PI and P and cascade PID and P for -180 degree amplitude disturbance	102
4.10	Progress of the activities	104

LISTS OF FIGURE

No	Title	Page
1.0	Inverted pendulum architecture	02
1.1	C# platform structure	03
1.2	Scope Chart	05
2.0	Step response for third order system with Ziegler-Nichols PID controller, 3-step grey prediction and 150-step grey-prediction	09
2.1	PID and fuzzy PD+I control results on the three-order system	10
2.2	PD and fuzzy PD control results on the three-order system	10
2.3	PI and fuzzy PI control results on the third-order system	11
2.4	PID control, PD control and Fuzzy PD control results on the nonlinear system	11
2.5	Step disturbance of P with secondary loop, PD and PID without secondary loop	12
2.6	Step response of P with secondary loop, PD and PID without secondary loop	13
2.7	PV vs. time, for three values of K_p (K_i and K_d held constant)	24
2.8	PV vs. time, for three values of K_i (K_p and K_d held constant)	24

2.9	PV vs. time, for three values of Kd (Kp and Ki held constant)	24
2.10	Open loop reaction curve	27
2.11	Closed loop oscillation	27
2.12	Simulink model for antiwindup	31
2.13	System output without anti-windup	32
2.14	PID output without anti-windup	32
2.15	System output with anti-windup	32
2.16	PID output with anti-windup	32
2.17	Cascade configuration for dc motor	33
2.18	DC motor schematic	37
2.19	Block diagram of motor mechanical part	38
2.20	Block diagram of motor electrical part	38
3.0	Methodology Flow Chart	40
3.1	General overview of methodology	41
3.2	Simple quadrature decoder using flip-flop	44
3.3	Input and output of a quadrature decoder	45
3.4	Typical H-bridge (First design)	47
3.5	Isolation measure	47
3.6	H-bridge from motor driver chip (Second Design)	48
3.7	Motor Braking	49
3.8	Motor forward	49
3.9	Knowledge base architecture	52
4.0	Plot V_{dc}/I_a vs w/I_a and linear fitting	58

4.1	Position(rev) output of PI ($K_p = 66, K_i = 25$)	61
4.2	Speed(rad/sec) output of PI ($K_p = 66, K_i = 25$)	60
4.3	Control Signal of PI ($K_p = 66, K_i = 25$)	60
4.4	Real time position output of PI ($K_p = 66, K_i = 25$)	63
4.5	Real time speed output of PI ($K_p = 66, K_i = 25$)	63
4.6	Real time control signal of PI ($K_p = 66, K_i = 25$)	64
4.7	Position(rev) output of PID ($K_p = 160, K_i = 10, K_d = 2$)	65
4.8	Speed(rad/sec) output of PID ($K_p = 160, K_i = 10, K_d = 2$)	65
4.9	Control Signal of PID ($K_p = 160, K_i = 10, K_d = 2$)	66
4.10	Real time position output of PID ($K_p = 160, K_i = 10, K_d = 2$)	67
4.11	Real time speed output of PID ($K_p = 160, K_i = 10, K_d = 2$)	67
4.12	Real time control signal of PID ($K_p = 160, K_i = 10, K_d = 2$)	68
4.13	Simulink model of inverted pendulum for cascade PI and P	69
4.14	Position(rev) output for 2 cascade loop	70
4.15	Speed(rev/min) output for 2 cascade loop	71
4.16	Torque(Nm) output for 2 cascade loop	71
4.17	Position(rev) output for 3 cascade loop	72
4.18	Speed(rev/min) output for 3 cascade loop	72
4.19	Torque(Nm) output for 3 cascade loop	73
4.20	Position(rev) output of cascade P and P	74
4.21	Speed(rad/sec) output of cascade P and P	75
4.22	Control Signal of cascade P and P	75
4.23	Real time position output of cascade P and P	76

4.24	Real time speed output of cascade P and P	76
4.25	Real time control signal of cascade P and P	77
4.26	Position(rev) output of cascade PD and P	78
4.27	Speed (rad/sec) output of cascade PD and P	78
4.28	Control Signal of cascade PD and P	79
4.29	Real time position output for cascade PD & P	79
4.30	Real time speed output of cascade PD & P	80
4.31	Real time control signal of cascade PD and P	80
4.32	Angle (rev) output of cascaded PI and P simulation	81
4.33	Speed (rad/sec) output of cascaded PI and P simulation	82
4.34	Control Signal of cascaded PI and P simulation	82
4.35	Real time position output of cascade PI and P	83
4.36	Real time speed output of cascade PI and P	83
4.37	Real time control signal of cascade PI and P	84
4.38	Position(rev) output of cascade PID and P	85
4.39	Speed(rad/s) output of cascade PID and P	85
4.40	Control Signal of cascade PID and P	86
4.41	Real time position output of cascade PID & P	87
4.42	Real time speed output of cascade PID & P	87
4.43	Real time control signal of cascade PID and P	88
4.44	Angle (rev) output of a 5x5 fuzzy system	89
4.45	Speed (rad/sec) output of a 5x5 fuzzy system	89
4.46	Real time angle output of 5x5 fuzzy logic	90

4.47	Real time speed output of 5x5 fuzzy logic	90
4.48	Response of PI toward 90 degree amplitude pulse	91
4.49	Response of PID toward 90 degree amplitude pulse	92
4.50	Response of cascade PI and P toward 90 degree amplitude pulse	92
4.51	Response of cascade PID and P toward 90 degree amplitude pulse	93
4.52	Response of PI toward 180 degree amplitude pulse	93
4.53	Response of PID toward 180 degree amplitude pulse	94
4.54	Response of cascade PI and P toward 180 degree amplitude pulse	94
4.55	Response of cascade PID and P toward 180 degree amplitude pulse	95
4.56	Response of PI toward -90 degree amplitude pulse	95
4.57	Response of PID toward -90 degree amplitude pulse	96
4.58	Response of cascade PI and P toward -90 degree amplitude pulse	96
4.59	Response of cascade PID and P toward -90 degree amplitude pulse	97
4.60	Response of PI toward -180 degree amplitude pulse	97
4.61	Response of PID toward -180 degree amplitude pulse	98
4.62	Response of cascade PI and P toward -180 degree amplitude pulse	98
4.63	Response of cascade PID and P toward -180 degree amplitude pulse	99
4.64	CPU usage	105

LIST OF SYMBOLS AND TERMS

PID	-	Proportional-Integral-Derivative
PI	-	Proportional-Integral
PD	-	Proportional-Derivative
PWM	-	Pulse Width Modulation
PCB	-	Printed Circuit Board
DC	-	Direct Current
PIC	-	Programmable Intelligent Computer
PC	-	Personal Computer
SP	-	Setpoint
PV	-	Present value
GUI	-	Graphical User Interface
GC	-	Garbage Collector
Kp	-	Proportional Gain
Ki	-	Integral Gain
Kd	-	Derivative Gain
SSE	-	Steady State Error
OS%	-	Overshoot
Ts	-	Settling time
Tp	-	Peak Time

Tr - Rise Time
ZOH - Zero Order Hold

LIST OF APPENDIX

No	Title	Page
A	Gantt Chart	111
B	C# Source Code	114
C	PIC Source Code	139
D	Pendulum Pole Structures	142
E	PCB Circuit	143
F	Paper	145

CHAPTER 1

INTRODUCTION

1.1 Background

Inverted pendulum is a popular platform for control engineers to test the controller. Conventionally, a pendulum is a ball tied to an end of a string. The ball is left freely. Another end of the string is fixed on a same location. Whenever an external force is applied to the ball, it will swing, for example, to the left. Due gravitational force, damping factor and air friction, the swing will eventually stop, and then it will swing back in the reverse direction (right). This sequence will repeat itself but each cycle, the swing angle will decrease. The pendulum will stop when the angle reach zero (with respect to surface perpendicular).

Inverted pendulum is revolutionized from pendulum hence inverted pendulum applies the same theory. But, instead of pendulum hanging downward, inverted pendulum is hanging upward. With no external force, the pole which replaces string, unable to maintain at the perpendicular position (zero angle). Any external force is considered extra for the balancing process. In order to balance the pole, controller must be implemented.

Proportional-Integral-Derivative (PID) controller is used to maintain the pole's position. PID is a conventional way of controlling a system. Practically, PID is the simplest, easiest and cheapest method comparing to other method. It is also

the most developed method in which various tuning, setting and developing method available.

1.2 Concept

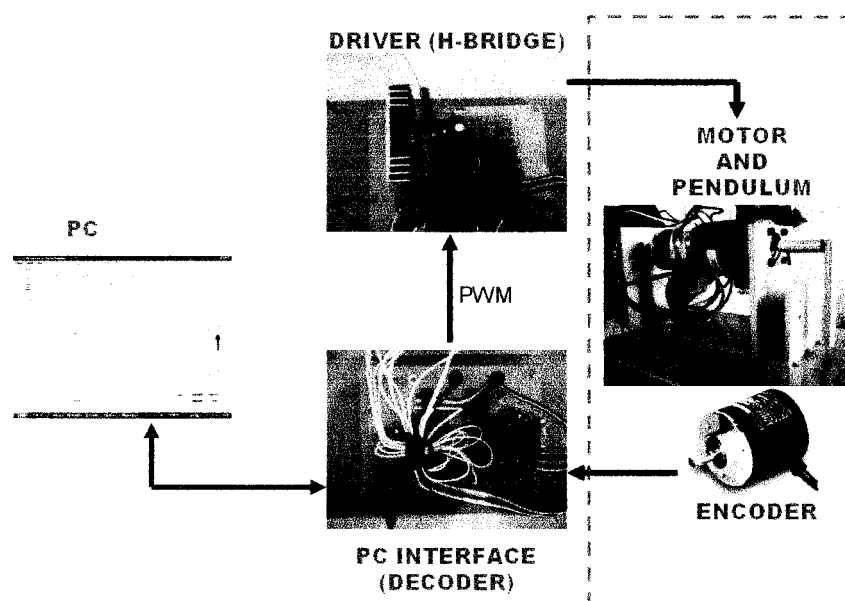


Figure 1.0: Inverted pendulum architecture

Figure 1.0 shows the architecture of this inverted pendulum system. Motor and encoder are coupled together. Mechanical structure which is the pendulum pole is mounted directly to the motor shaft. Interface is consisted of electronic circuit in which will responsible to pass data between pc, encoder and motor driver. Data will be received at pc side and it will be process to determine what kind of response the actuator should be exhibited.

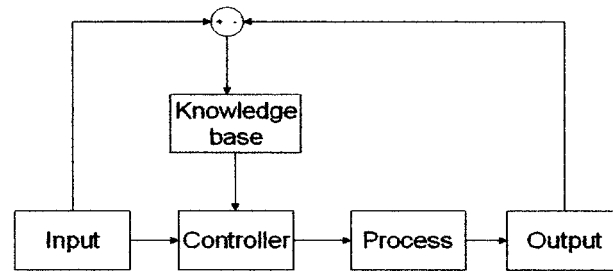


Figure 1.1: C# platform structure

C# platform structure is shown in Figure. Input is the setpoint and output is the angle. Flow 'Input to output' and flow 'Input & Output to Knowledge Base' are operating separately and independently. Using this approach, processing time is reduced compared to the conventional method where the data collection, signal processing and actuator driving happen serially.

1.3 Problem Statement

First challenge of an inverted pendulum system is the balance the pole. It is instable for the system to maintain the pole in a perpendicular position relative to the surface. The system pole initially pointing downward, then rotate upward to balance it. Another problem is the closed loop system (feedback). It is unable to produce good results in term of balancing. Even if it able to balance the pole, there may be some unwanted effect such as long settling time, overshoot, steady state error and etcetera may occur and this will affect the quality of the system. After initial balancing, there is another challenge which is the system resistance towards disturbance. Disturbance may cause the system to instability.