Gain Scheduling Based PID Controller for a Two Degree of Freedom Manipulator Robot

Gaber Elsaady ,Elnoby Ahmed and Mostafa Thabet
Electrical Engineering Department, Faculty of Engineering Assiut University, Asyut, Egypt
Most.thabet@gmail.com

ABSTRACT—This paper presents a PID control scheme for position control of a two degrees of freedom (DOF) Manipulator robot. A modified a gain scheduling structure used to identify the PID parameters the gain scheduling technique enables the controller to adapt on-line the PID parameters depending on the operation conditions (essentially the robot arm position)[7],[8] . It also compares between constant PID parameters system and on-line variable PID parameters system, The controlled target is an arm robot with two links, where the arm head of the robot is moved by permanent magnet DC motor with encoder controlled by Arduino. The whole system is simulated PROTEUS v8.3 and the results for position control show the effectiveness of the proposed approach.

Keywords- ARM Manipulator, Gain schedule, PID control Pulse width modulation.

I. INTRODUCTION

Robots are clear examples of mechanical systems run on certain instructions and several types of research work on providing simple and easy control system.

Robot manipulators are used in a lot of applications such as move materials, parts, and various types[1]. This paper presents a type of horizontal drive that is used as the arrangement of parts to a printed wiring board and a product assembly, and is usually controlled by PID compensator to get an exact moving in controlling the ARM robot, the arm head is moved by motors, so each link is required its accuracy and incoherency. In general, gains of PID controller are determined through trial and error processes by skilled experts, even though the adjustment of gains following to alterations of modeling errors and frictions are difficult. The difference in the output of the target and the model is feed-backed and the output of the targets good performance by reflecting the difference to the controller. The movement of the two link arm manipulator depends on two angles[3],[4], each angle gives by separate motor each motor has its pulse width modulation signal, it controls calculations and it`s encoder signal, the two motors don`t have the same speed[9].To get the best accuracy you need a motor with full control in speed and step (angle).Many types of research use a stepper motor or servo motor. Stepper motor performs well in angle control and Servo motor is great in speed control, but here we use PMDC motor, but PMDC motor only is not enough you can`t get full control in an angle so we use PMDC motor enhanced by angle encoder on motor`s shaft to achieve closed-loop control concept. Here we use C language for Arduino programming where it used to build the PID function and gain schedule calculation and control in the motor speed using pulse width modulation. First, the pulse width modulation starts with soft-start function to reduce the vibration on the arm, note increasing the acceleration time will increase the reach time. After soft-starting the pulse width modulation depends on the current position of the motor, current speed and the deceleration time. The current position is a signal came from motor`s encoder and it calculated by PID equations if the difference between set-point and current point is larger so the speed will faster. Also, the PID equation gets the current speed and calculate the deceleration time to make the arm stop in the exact angle. So our target is using the control system to reduce the settling time, steady-state error and deceleration error.

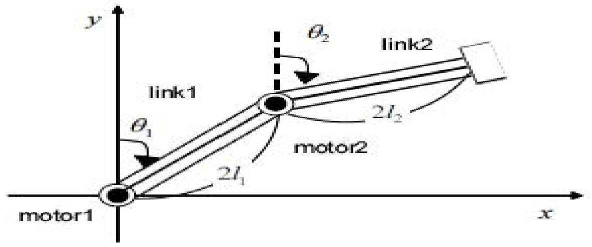
In this paper,3 types of PID control systems

1- PID control with fixed PID parameters that give smooth effect.
Constant values of Kp, Ki, and Kd that gives smooth effect means the variation of motor speed is smooth not fast it gives accurate results but with long reaching time
2- PID control with fixed PID parameters that give aggressive effect
Constant values of Kp, Ki, and Kd that gives aggressive effect means the variation of motor speed is aggressive not it does not give accurate results but with short reaching time
3- Gain schedule based PID control not only use both values smooth and aggressive but also use more values among them it depend on the difference between set-point and current angle it gives accurate results in short time [2],[5],[6].

We will prove that using system simulation using PROTEUS PROFESSIONAL V8.3 and get the HEX code from Arduino UNO

II. Arm robot motion equations

Fig. 1 shows a model of an arm robot with two links. In this model, parameters are defined as shown in Table 1,



Fig(1) Arm robot model

From Fig. 1, the coordinate of the set point for links is expressed as follows.

Motor1 angle and link1 length we can get point1(x_1,y_1) coordinates

x_1= L_1*sinΘ1

y_1= L_1*cosΘ1

Motor2 angle and link2 length we can get point2(x_2,y_2) coordinates

x_2= 2*L_1*sin Θ1+ L_2*sin Θ2

y_2= 2*L1*cosΘ1+ L_2*cosΘ2

Also, coordinate of position of arm are (x,y) as follows.

x = 2*L_1*sin Θ1+ 2*L_2*sin Θ_2 (1)

y = 2*L_1*cosΘ_1+ 2*L_2*cosΘ_2 (2)

Table (1)Arm parameters

m_i	mass of link *I*[kg]
l_i	length of link *i* [m]
Θ_i	Motor i angle

III. System block Diagram

In this section, the control system is designed in the following steps starting with AC source 220v AC, the AC signal goes throw A **rectifier** that converts alternating current (AC), to direct current (DC), which flows in only one direction.

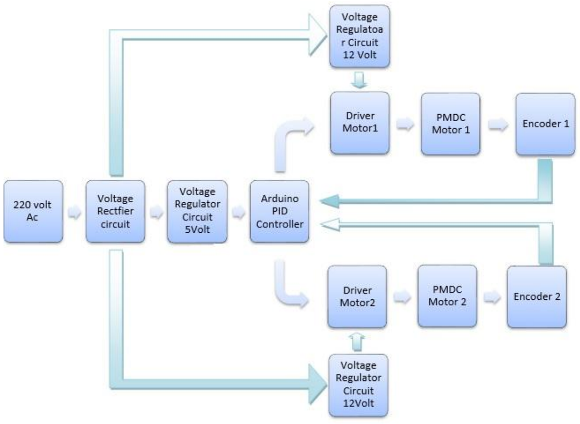


Fig (2) System Block diagram

The next step is the regulation**,** A voltage regulator is a device that maintains a fixed output voltage even variable input voltage Here there are two regulators

- 12 volt used for motor driver and motors
- 5 volt used for Arduino

The Arduino is the controller and it will perform the PID calculations .The calculation depends on

- **A set-point angle** is the desired angle (x,y) we can get it from (1), (2)
- **Current angle** it's the angle which will the motor start from and it represent the feedback signal, it comes from the encoder
- **PID parameters it** uses three basic control behaviors

Kp : P- constant gives an output which is proportional to current error, It compares set point with the actual value

Ki: I-constant is needed, use to eliminate the steady state error. It integrates the error over a period of time until the error value reaches zero

Kd: D-constant predicts the future behavior of the error. Its output depends on the rate of change of error with respect to time, multiplied by derivative constant. Driver Motor it receives the signal from Arduino to control the DC motor, Signals types:

1. **Direction's signal** its change due to the set-point angle and the current angle
2. **Pulse Width Modulation signals** depend on PID parameters, Angle error, and set-point, here there are two motors, motor1 control on theta1 and motor2 control on theta2,Each motor has it`s PWM signal (input signal)**,** Direction's signals (input signal) **,**Encoder signals (output signal)

Motors parameters

• L: Motor inductance	• R: Motor armature resistance
• kv : Back-emf constant	• ka: Torque constant.
• f: Friction coefficient.	• J: Motor inertia.

IV. system built up, Simulation and Results

1- System Components

1.1Arduino Uno

The Arduino Uno is a microcontroller board based on the ATmega328.It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button.

Table (2**)** Arduino UNO properties

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V	50 mA
Clock Speed	16 MHz

1.2 Driver motor

DC Driver motor is a rather simple circuit, containing 3 switching element, 2 relays, and 1 transistor.

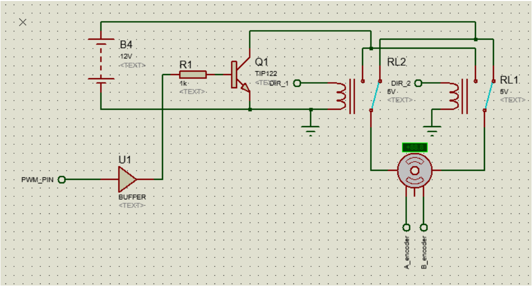


Fig (3) DC Driver motor circuit diagram

The switching elements (Q1) are usually bi-polar or MOSFET transistors, the most part for this discussion. It is responsible for pulse width modulation (speed control).The NC pin in relay1 connected to NC pin in relay2 are connected to a power supply (battery for example) terminal 1, and The NO pin in relay1 connected to NO pin in relay2 are connected to a power supply terminal 2.The two relays use in direction control of the DC motor. Though the load can, in theory, be anything you want, by far the most pervasive application.

1.3 DC-Motor

When a voltage UA(t) is applied on the motor, current i (t) accelerates the motor. As the angular velocity w(t) = Ө (t) increases, the friction f w(t) in motor and drive shafts, and back-emf (electromotive force) kv w(t), reduces the acceleration ˙w(t) =Ӫ (t).

Table (3) Motor Parameters

Jm = 3.2284E-6	moment of inertia of the rotor
Bm = 3.5077E-6	motor viscous friction constant
K = 0.0274	electromotive force constant
Ra = 4	electric resistance
La= 2.75E-6	electric inductance

1.4 Rotary Encoder

The dc motor is equipped with a simple encoder to measure the rotation of the motor shaft. The encoder uses n = 12 holes called quadrature design, meaning it has a resolution of 48 pulse changes per revolution. The motor shaft is connected via a gear with transmission ratio 15, meaning the encoder will deliver 48·15 = 720 pulse changes per motor shaft revolution

1.5 Graphical LCD

The Graphical LCDs are thus used to display customized characters and Shapes. This LCD has a display format of 128x64 dots we use it to display the movement and the position of the two link arm robot

2. Numerical Simulation and results

Numerical simulation is performed by using the values of the parameters in table 1, table 2 and table 3

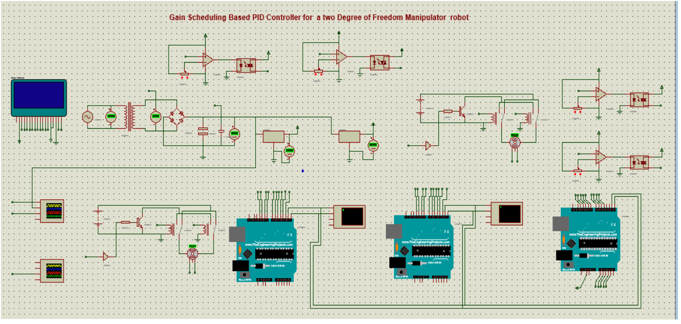
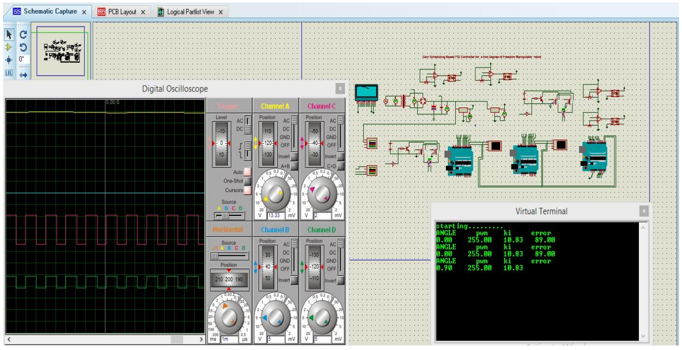
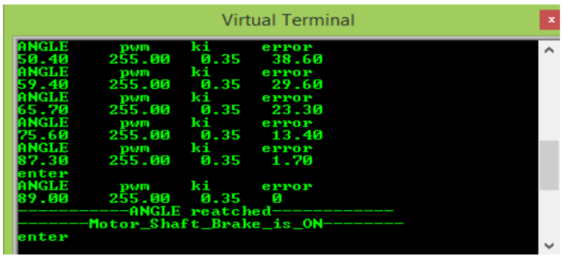

Gain Scheduling Based PID Controller for a two Degree of Freedom Manipulator robot

Fig (4) System schematic diagram



```
starting..........
ANGLE      pwm    ki     error
0.00    255.00   10.83   89.00
ANGLE      pwm    ki     error
0.00    255.00   10.83   89.00
ANGLE      pwm    ki     error
0.90    255.00   10.83
```

Fig (5) System on simulation runtime



```
ANGLE      pwm    ki      error
50.40    255.00   0.35    38.60
ANGLE      pwm    ki      error
59.40    255.00   0.35    29.60
ANGLE      pwm    ki      error
65.70    255.00   0.35    23.30
ANGLE      pwm    ki      error
75.60    255.00   0.35    13.40
ANGLE      pwm    ki      error
87.30    255.00   0.35    1.70
enter
ANGLE      pwm    ki      error
89.00    255.00   0.35    0
-----------ANGLE reatched------------
-------Motor_Shaft_Brake_is_ON--------
enter
```

Fig (6) System on-line Parameters monitoring

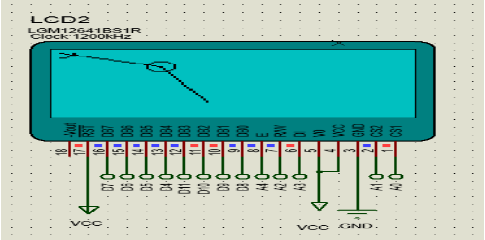
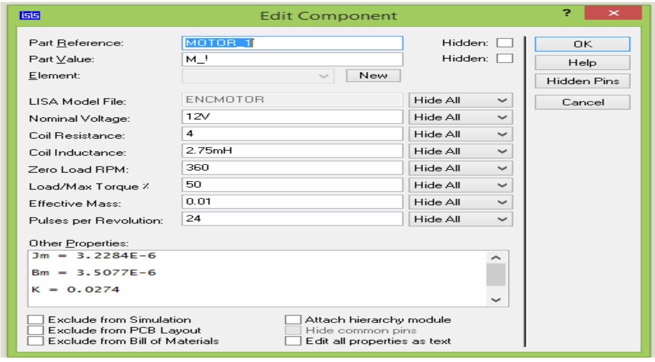


Fig (7) Two link movement Display on Graphic LCD



Edit Component

Part Reference: MOTOR_1 Hidden: □
Part Value: M_! Hidden: □
Element: New

LISA Model File: ENCMOTOR Hide All
Nominal Voltage: 12V Hide All
Coil Resistance: 4 Hide All
Coil Inductance: 2.75mH Hide All
Zero Load RPM: 360 Hide All
Load/Max Torque % 50 Hide All
Effective Mass: 0.01 Hide All
Pulses per Revolution: 24 Hide All

Other Properties:

```
Jm = 3.2284E-6
Bm = 3.5077E-6
K = 0.0274
```

□ Exclude from Simulation
□ Exclude from PCB Layout
□ Exclude from Bill of Materials
□ Attach hierarchy module
□ Hide common pins
□ Edit all properties as text

Fig (8) Setting of motor parameters as given in table (1)

2.1 System response of position control using Fixed PID values

2.1.1 fixed PID parameters with smooth values

The simulation result is shown in Figure (9) and (10)

Kp=0.20 Ki=0.150 Kd=-0.3, Set-point=89 `

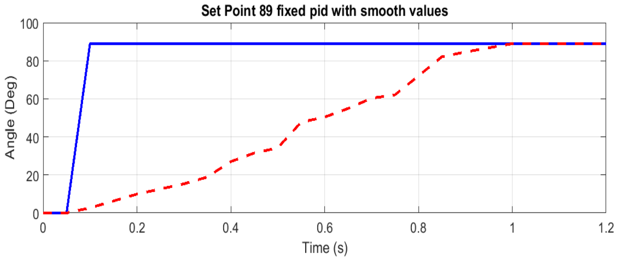
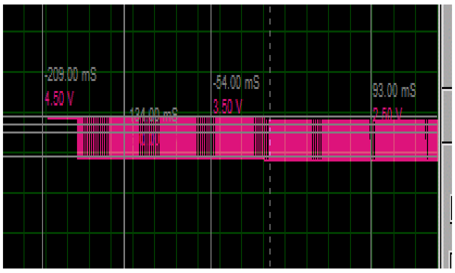


Fig (9) System response for fixed **smooth** PID



Fig(10)PWM variation for Fixed **smooth** PID

2.1.2 fixed PID parameters with Aggressive values

The simulation result is shown in Figure (11) and (12)

Kp=0.286 Ki=0.3505 Kd= -0.306,Set-point=89 `

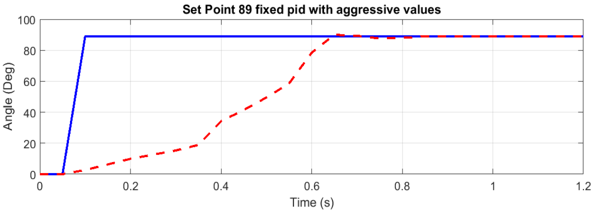


Fig (11)System response for Fixed **Aggressive** PID

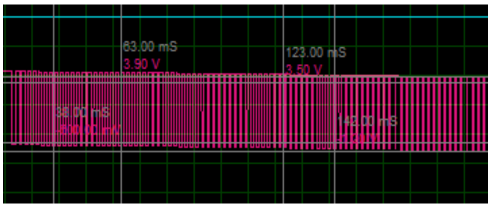


Fig (12) PWM variation for Fixed **Aggressive** PID

2.2 System response for position control using Gain Scheduling Based PID Controller

Here the system adapt on-line the PID parameters depending on

- Angle Set-point
- Angle error correction
- The Last Pulse width modulation value
- Motor Direction

Table (4) PID parameters for Gain schedule method

Set point Angle Range	PID parameters
0:5	Kp=0.01Ki=0.002Kd=0.09
10:15	Kp=0.017Ki=0.0028Kd=0.98
20:25	Kp=0.20Ki=0.105Kd=-0.3
30:45	Kp=0.82Ki=.13Kd=-0.29
55:70	Kp=0.88 Ki=0.34Kd=-0.1
75:90	Kp=0.92 Ki=0.48Kd=-0.1

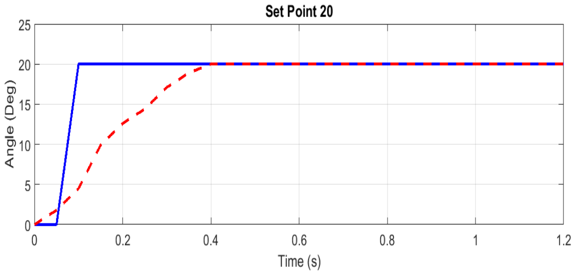


Fig (13)

System response for Gain Schedule PID, Set-point =20`

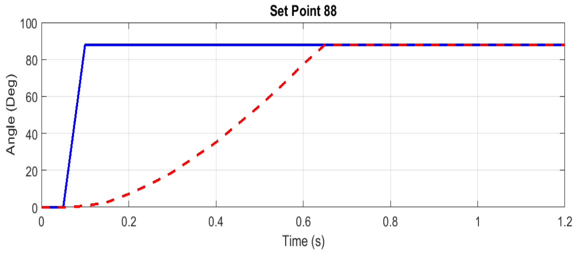
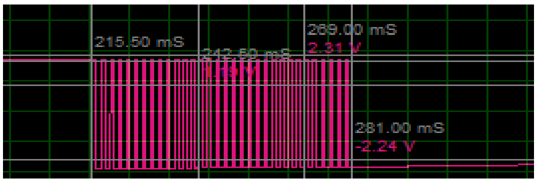


Fig (14)

System response for Gain Schedule PID, Set-point =88`



Fig(15)
PWM variation for Gain Schedule PID

According to simulation results

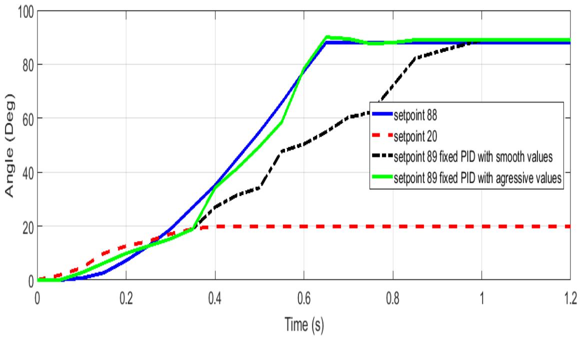


Fig (16) System response for all control systems

Table (5)Compare the results

	Gain Schedule PID	**Aggressive** PID	**smooth** PID
Settling Time	Very Short	short	long
Reaching set-point time at set-point=89`	0.61 msec Faster	0.76 msec Fast	0.988 msec Slow
PWM type	Non-linear	Non-linear	Non-linear
Deceleration error	No –error	Error	Error

5- CONCLUSIONS

Gain Schedule PID is used to attain good performance in designing the controller of the two link manipulator robot and then the control is determined by a simple manner. The Gain Schedule PID is compared with that of PID numerically, and the superiority of Gain Schedule PID is confirmed.
Research along this line will continue to implement the control algorithm in real time and to further test it in a laboratory environment as well as consider the two-link manipulator with an unknown load

6- References :

[1] K.Yoneda, T.Tsubouchi, and H.Ohkuma: The first robot creation design, Koudansya Scientific (2003).
[2] The Electrical Engineering Handbook, 2005
[3] T.Mita and K.Osuka: Introduction to robot control engineering, Corona Publishing(1989), 8-22.
[4] S.Kawamura: Introduction to robot control engineering, Ohmsha (1997), 57-59.
[5] K.J. Åström and T. Hägglund, PID Controllers: Theory, Design, and Tuning. Research Triangle Park, NC: Instrum. Soc. Amer. 1995.
[6]KARL J. ÅSTRÖM and TORE HÄGGLUNDK.J. Åström and T. Hägglund, Automatic Tuning of PID Controllers. Research Triangle Park, NC: Instrum. Soc. Amer.1988.
[7] Astrom, K.J. and B. Wittenmark. Adaptive control. Addison Wesley. (1989)
[8] Craig, J.J. Introduction to robotics, mechanics, and control. Addison Wesley. (1989)
[9] Sciavicco, L. and B. Siciliano. Modeling and control of robot manipulators. McGraw Hill. (1996).