

PALESTINE POLYTECHNIC UNIVERSITY



**College of Engineering and Technology**  
**Mechanical Engineering Department**  
**Graduation Project**  
**2-DOF Vertical Planner Manipulator**

*By*

*Tareq A.Melhem*

*Anas I.Abulehia*

Supervised by

*Dr.Yousef Al-Sweiti*

*Hebron-Palestine*

June 9, 2016

PALESTINE POLYTECHNIC UNIVERSITY  
COLLEGE OF ENGINEERING AND TECHNOLOGY  
MECHANICAL ENGINEERING DEPARTMENT

2-DOF Vertical Planner Manipulator

Project Team

Tareq Melhem

Anas Abulehia

According to directions of the project supervisor and by the agreement of all examination committee members, this project is presented to the department of Mechanical Engineering at College of Engineering and Technology, for partial fulfillment Bachelor of engineering degree requirements.

Supervisor Signature

Dr. Yousef Al-Sweiti

.....

Committee Member Signature

*Prof. Dr. Karim Tahboub*

*Dr. Iyad Hashlamon*

.....

.....

Department Head Signature

Eng. Jalal Salaimh

.....

# Abstract

**Our graduation project is 2-DOF Vertical Planner Manipulator, simply the manipulator has do two main functions, first one is perform any 2-D sketch in vertical plane, second one is to be able to repeat a predetermined motion (teach the required motion by moving the head, then it has to repeat that motion.**

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	The need of the project . . . . .	1
1.2	Specifications . . . . .	1
1.3	Robotics overview . . . . .	1
1.4	Robotic mechanical structure . . . . .	3
1.5	Robotic system . . . . .	3
1.6	Project overview . . . . .	3
1.7	Action plan . . . . .	3
<b>2</b>	<b>Geometrical Design and Dynamic Analysis</b>	<b>5</b>
2.1	Suggested designs and selection . . . . .	5
2.2	Sketch and dimensions with work space . . . . .	9
2.3	Forward kinematics . . . . .	10
2.4	Inverse kinematics . . . . .	11
<b>3</b>	<b>Kinetics of the robot</b>	<b>13</b>
3.1	Introduction . . . . .	13
3.2	Mathematical model . . . . .	13
<b>4</b>	<b>Trajectory generation</b>	<b>18</b>
4.1	Introduction . . . . .	18
4.2	Point-to-point motion . . . . .	18
4.2.1	Cubic polynomials . . . . .	18
<b>5</b>	<b>Mechanical design</b>	<b>24</b>
5.1	Stress analysis . . . . .	24
5.2	Deflection analysis . . . . .	26

<b>6</b>	<b>Control Design and Hardware Implementation</b>	<b>30</b>
6.1	Introduction . . . . .	30
6.2	Hardware components . . . . .	30
6.2.1	Microcontroller . . . . .	30
6.2.2	Rotary potentiometer . . . . .	31
6.2.3	Actuators . . . . .	32
6.2.4	Motor drivers . . . . .	33
6.3	Control design . . . . .	34
6.3.1	PID controller . . . . .	34
6.4	Tests and results . . . . .	35
	Bibliography . . . . .	39
	<b>Appendices</b>	<b>40</b>

# List of Figures

1.1	Two link robot arm . . . . .	2
2.1	The two designs . . . . .	5
2.2	considered work-space . . . . .	9
2.3	links length of the manipulator . . . . .	9
2.4	Different situations of the manipulator . . . . .	10
2.5	Two link robot arm . . . . .	11
2.6	Two link robot arm with explained angles . . . . .	12
3.1	Manipulator diagram . . . . .	15
3.2	The Non-linear model using matlab simulation . . . . .	17
4.1	(a) Cubic polynomial trajectory.(b)Velocity profile for cubic polynomial trajectory.(c)Acceleration profile for cubic polynomial trajectory. . . . .	20
4.2	work-pace . . . . .	20
4.3	Cubic polynomial trajectory . . . . .	21
4.4	Velocity profile for cubic polynomial trajectory . . . . .	21
4.5	Acceleration profile for cubic polynomial trajectory . . . . .	21
4.6	Torque curves for each joint for cubic polynomial trajectory . . . . .	22
4.7	Cubic polynomial trajectory . . . . .	22
4.8	Velocity profile for cubic polynomial trajectory . . . . .	22
4.9	Acceleration profile for cubic polynomial trajectory . . . . .	23
4.10	Torque curves for each joint for cubic polynomial trajectory . . . . .	23
5.2	Upper link of manipulator projections . . . . .	24
5.1	Isometric view for the CAD design of our manipulator . . . . .	25
5.3	The cross sectional area for moment of Inertia . . . . .	25
5.4	Lower link of manipulator projections . . . . .	26
5.5	Diagram of the beam for deflection analysis . . . . .	27

5.6	free body diagram with applied shear force . . . . .	27
5.7	lower link deflection . . . . .	28
5.8	upper link deflection . . . . .	29
6.1	arduino mega 2560 . . . . .	31
6.2	arduino software development tool . . . . .	31
6.3	internal connection of potentiometer . . . . .	32
6.4	wiper motor . . . . .	32
6.5	pitman motor with gear box . . . . .	33
6.6	Upper motor driving circuit . . . . .	33
6.7	Lower motor driving circuit . . . . .	34
6.8	equivalent block diagram for motor and attached link . . . . .	34
6.9	first trial . . . . .	35
6.10	first trial . . . . .	36
6.11	theta one move form 90 to 150 degree . . . . .	36
6.12	theta one move form 110 to 70 degree . . . . .	37
6.13	theta one move form 110 to 70 degree . . . . .	37
6.14	theta one move form 150 to 10 degree . . . . .	38

# List of Tables

1.1	Actions of the project for the first semester . . . . .	4
2.1	Comparison between serial and parallel manipulator . . . . .	8
5.1	All parameters of the manipulator . . . . .	29



# Chapter 1

## Introduction

### 1.1 The need of the project

We want to help teachers to draw sketches on the board (vertical plane) during the lecture to be more clear for student.

### 1.2 Specifications

- 1- Workspace 50X40 cm.
- 2- Precision 5mm.
- 3- Rigid.
- 4- Speed (cross the diagonal of workspace within 3 sec).
- 5- Quiet enough.
- 6- No programming needed from the point view of the user.
- 7- Easy to use.

### 1.3 Robotics overview

Robotics is concerned with the study of those machines that can replace human beings in the execution of task, as regards both physical activity and decision making.

The end-effector, or robotic hand, can be designed to perform any desired task such as welding, gripping, spinning etc., depending on the application. In our case the term robot will mean a computer controlled planner manipulator.

This type of robot is essentially a mechanical arm operating under computer control. figure 1.1 illustrate Kuka Arm robot model KR 360 R2830 (KR 360 FORTEC).



Figure 1.1: Two link robot arm

## **1.4 Robotic mechanical structure**

One of the most important things in any robot is the mechanical structure of that one. The robot structure consists basically of the robot body that includes links and joints. Some force such as electrical energy is required to make the links turn under command. One of the most interesting aspects of robot in general is its behavior. In our project we focus on arm robot with fixed base known as manipulator with two Degree of freedom (planner) to perform predetermined tasks, and in our case is drawing on vertical board.

Manipulators are composed of an assembly of links and joints. Links are defined as the rigid sections that make up the mechanism and joints are defined as the connection between two links, the joints can be rotary or prismatic in our case it is rotary joints and the device that attached which interacts with the environment is end effector.

## **1.5 Robotic system**

A robot manipulator is not just components connected with each other, it is consist of arm, external power source, end effector, sensors, actuators, drivers, computer controller and input device it may be embedded in the computer.

## **1.6 Project overview**

This work concerns the design and control of planner manipulator that has two degree of freedom. In the end it has to do two features, first one is the robot will be able to repeat a predetermined motion , the second one is it will be perform any 2-D sketch.

## **1.7 Action plan**

The time plan explains the stages in designing. The section includes the first one show what is done in the first semester.

Table 1.1: Actions of the project for the first semester

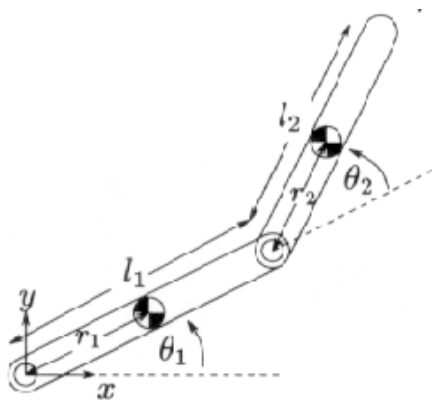
action \ week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Selection The Idea of the Project	■	■													
conceptual design			■												
study forward and invers kinematic			■	■											
drive the mathematical model of chosen design				■	■										
modular mathematical model using matlab						■	■	■							
Mehcnical Design									■	■					
detailed modular mathematical model											■	■			
control system design and Linearization													■	■	■
Simulation Using Matlab	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
Learning on Latex Edittor	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■

# Chapter 2

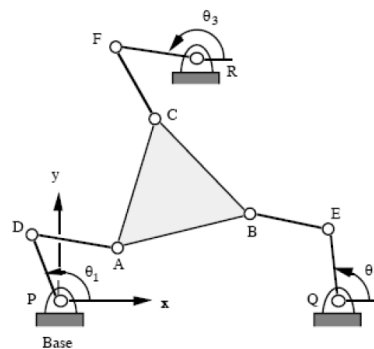
## Geometrical Design and Dynamic Analysis

### 2.1 Suggested designs and selection

To meet the needs of our project we think two ways to that first one is serial manipulator the second one is parallel manipulator the figure



(a) serial manipulator



(b) parallel manipulator

Figure 2.1: The two designs

A **serial robot** consists of several links connected in series by various types of joints, typically revolute and prismatic. One end of the robot is attached to the ground and the other end is free to move in plane. The fixed link is called base, and the free end where a gripper or a mechanical hand is attached, the end effector. For a robot to perform a specific task, the location of the end effector relative to the base should be established first. This is called position analysis problem. There are two types of position analysis problems: direct

position or direct kinematics and inverse position or inverse kinematics problems.

For direct kinematics, the joint variables are given and the problem is to find the location of the end effector. For inverse kinematics, the location of the end effector is given and the problem is to find joint variables necessary to bring end effector to the desired position. For a serial robot direct.

When a robot manipulator performs a given task, the end effector exerts some force and/or moment on its environment. This contact force and/or moment will cause the end effector to be deflected away from its desired location. Intuitively, the amount of deflection is a function of the applied force and the stiffness of the manipulator. Thus the stiffness of a robot manipulator has a direct impact on the position accuracy. Furthermore some advanced control strategies use the stiffness characteristics for feedback control of a robot manipulator. The overall stiffness of a robot manipulator depends on several factors, including the size of and material used for the links, the mechanical transmission mechanisms, the actuators and the controller. As the links become longer and more slender.

For some applications, such as arc welding, it is necessary to move the end effector of manipulator from point to point rapidly. The dynamics of the robot manipulator plays an important role in achieving such high-speed performance. The development of dynamical model is important in several ways. First, a dynamical model can be used for computer simulation of a robotic system. By examining the behavior of the model under various operating conditions, it is possible to predict how a robotic system will behave when it is built. Various automation tasks can be examined without the need of real system. Second, it can be used for the development of suitable control strategies. A sophisticated controller requires the use of a realistic dynamical model to achieve optimal performance under high-speed operations. Some control schemes rely directly on a dynamic model to compute actuator torques required to follow a desired trajectory. Third, the dynamic analysis of the manipulator reveals all the joint reaction forces (and moments) needed for the design and sizing of links, bearings and actuators. There are two types of dynamical problems: direct dynamics and inverse dynamics. The direct dynamic problem is to find the response of a robot arm corresponding to some applied torques and/or forces. That is, given a vector of joint torques or forces, we wish to compute the resulting motion of the robot manipulator as a function of time. The inverse dynamic problem is to find the actuator torques and/or forces

required to generate a desired trajectory of the manipulator. The problem can be formulated in joint space, or the end effector space. The two formulations are related by the Jacobian matrix and its time derivative. In general, the efficiency of computation for direct dynamics is not as critical since it is used primarily for computer simulations of a manipulator. On the other hand an efficient inverse dynamical model becomes extremely important for real-time feed forward control of a robot manipulator.[3]

**A parallel robot manipulator** is composed of two or more closed-loop kinematic chains in which the end-effector (mobile platform) is connected to the fixed base platform by at least two independent kinematic chains. Between the base and end effector platforms are serial chains (called limbs or legs). For parallel robot manipulators two position analysis problems have to be solved: direct kinematics and indirect kinematics.

The dynamic analysis of parallel manipulators is complicated by the existence of multiple closed loop chains. Several approaches have been proposed, including the Newton-Euler formulation the Lagrangian formulation and the principle of virtual work. The traditional Newton-Euler formulation requires the equations of motion to be written once for each body of the manipulator, which inevitably leads to a large numbers of equations and results in poor computational efficiency. The Lagrangian formulation eliminates all of the unwanted reaction forces and moments at the outset. It is more efficient than the Newton-Euler formulation.[3]

Table 2.1 gives comparison between main characteristics of serial and parallel robots. So serial on one hand good advantages are available on serial manipulator and on the other hand there is disadvantages can not be neglect.

Our one requires are first large workspace (all point on the board must be accessible), second little force just to be able to afford its own and end-effector masses also and this related with stiffness so the third one reasonable stiffness, fourth simple calibration process finally is the seed we want high speed relatively. So that the serial is more appropriate for our specification.

Table 2.1: Comparison between serial and parallel manipulator

Feature	Serial robot	Parallel robot
Workspace	Large	Small and complex
Solving forward kinematics	Easy	Very difficult
Solving inverse kinematics	Difficult	Easy
Position error	Accumulates	Averages
Force error	Averages	Accumulates
Maximum force	Limited by minimum actuator force	Summation of all actuator forces
Dynamics characteristics	Poor, especially with increasing the size	Very high
Modelling and Solving dynamics	Relatively simple	Very complex
Areas of application	A great number in different areas, especially in industry	Currently limited, especially in industry
Payload/weight ratio	Low	High
Speed and acceleration	Low	High
Accuracy	Low	High
Uniformity of components	Low	High
Calibration	Relatively simple	Complicated
Workspace/robot ratio	High	Low



## 2.2 Sketch and dimensions with work space

we will design a manipulator for a work-space with  $50\text{cm} \times 40\text{cm}$  length, see figure 2.2. The base-joint of the robot is shifted  $5\text{cm}$  from the origin of the work-space to face it.

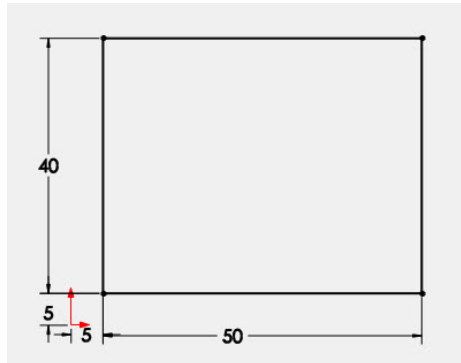


Figure 2.2: considered work-space

Now, to evaluating the length of manipulator's links, the distance between base-joint and the farthest point on the work-space is  $71.06\text{cm}$ , then we consider that  $l_1 = 40\text{cm}$  and the angle  $\theta = 17^\circ$ , (see figure 2.3) then using cosine law :

$$l_2^2 = l_1^2 + d^2 - 2l_1d\cos\theta$$

$$l_2 = 35\text{cm}$$

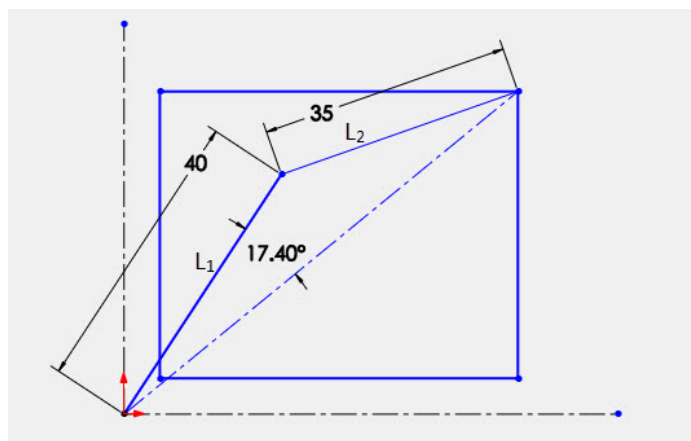


Figure 2.3: links length of the manipulator

Different situations of the manipulator, we can conclude that it able to access any point on our work-space (see figure 2.4).

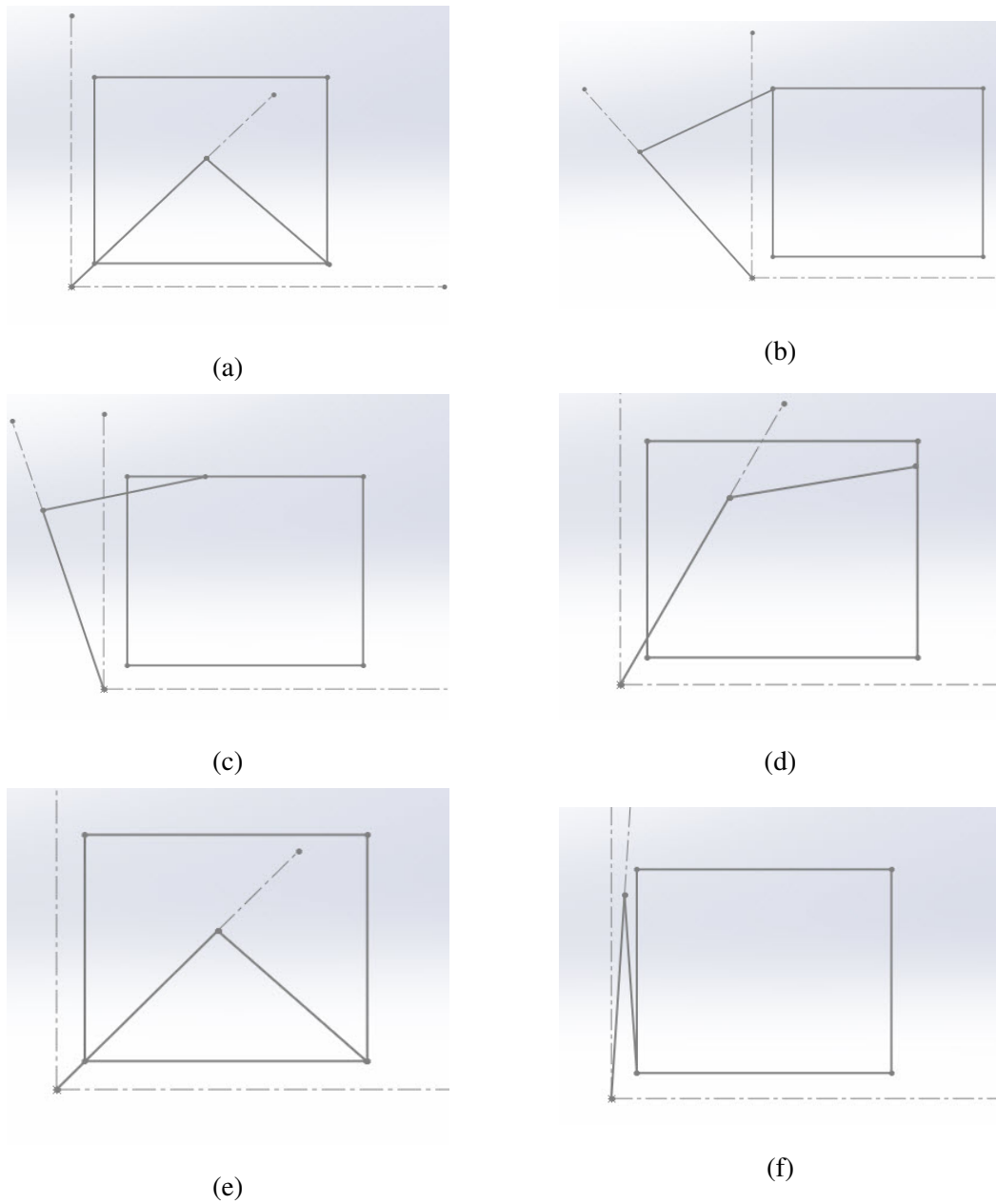


Figure 2.4: Different situations of the manipulator

## 2.3 Forward kinematics

In this section we show how we get the plan position, velocity and acceleration from the angles  $\theta_1$  and  $\theta_2$  of links that got from the actuators and encoders. Notice Figure 2.1.

$$f(\theta_1, \theta_2) \Rightarrow (x, y)$$

$$f(\theta_1, \dot{\theta}_1, \theta_2, \dot{\theta}_2) \Rightarrow (v_x, v_y)$$

$$f(\theta_1, \dot{\theta}_1, \ddot{\theta}_1, \theta_2, \dot{\theta}_2, \ddot{\theta}_2) \Rightarrow (a_x, a_y)$$

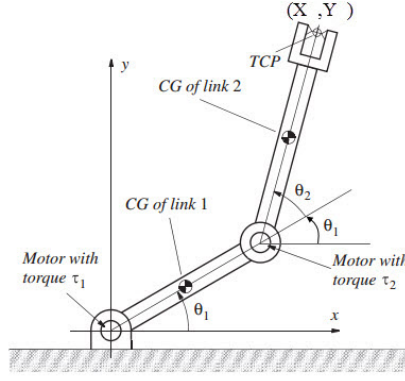


Figure 2.5: Two link robot arm

**The positions** From the figure it is clear that

$$P_x(t) = l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) \quad (2.1)$$

$$P_y(t) = l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2)$$

Now to get **velocity** functions we differentiate (2.3), then.

$$v_x(t) = -l_1 \dot{\theta}_1 \sin(\theta_1) - l_2 (\dot{\theta}_1 + \dot{\theta}_2) \sin(\theta_1 + \theta_2) \quad (2.2)$$

$$v_y(t) = l_1 \dot{\theta}_1 \cos(\theta_1) + l_2 (\dot{\theta}_1 + \dot{\theta}_2) \cos(\theta_1 + \theta_2)$$

In the same manner we differentiate (2.2), then we got **acceleration** equations.

$$a_x(t) = -l_2 \cos(\theta_1 + \theta_2) (\dot{\theta}_1 + \dot{\theta}_2)^2 - l_1 \sin(\theta_1) \ddot{\theta}_1 - l_1 \dot{\theta}_1^2 \cos(\theta_1) - l_2 \sin(\theta_1 + \theta_2) (\ddot{\theta}_1 + \ddot{\theta}_2) \quad (2.3)$$

$$a_y(t) = l_1 \cos(\theta_1) \ddot{\theta}_1 - l_2 \sin(\theta_1 + \theta_2) (\dot{\theta}_1 + \dot{\theta}_2)^2 + l_2 \cos(\theta_1 + \theta_2) (\ddot{\theta}_1 + \ddot{\theta}_2) - l_1 \dot{\theta}_1^2 \sin(\theta_1)$$

## 2.4 Inverse kinematics

We will find the angles  $\theta_1$  and  $\theta_2$  from given plane coordinates.

$$f(X_p, Y_p) \Rightarrow (\theta_1, \theta_2)$$

Start from cosines law

$$C^2 = l_1^2 + l_2^2 - 2l_1 l_2 \cos(\gamma) \quad (2.4)$$

and we know that the length of C is the summation of  $X_p^2$  added to  $Y_p^2$  reconstruction the equation 2.4 to get the  $\gamma$  value.

$$C = \sqrt{X_p^2 + Y_p^2} \quad (2.5)$$

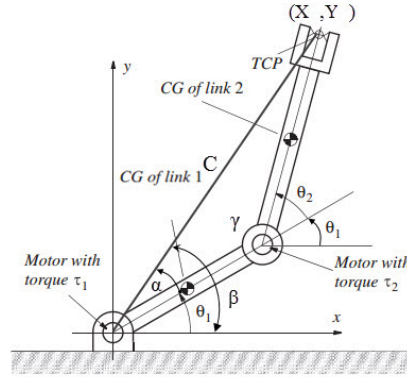


Figure 2.6: Two link robot arm with explained angles

$$\gamma = \cos^{-1} \left( \frac{l_1^2 + l_2^2 - C^2}{2l_1l_2} \right) \quad (2.6)$$

substitute equation 2.5 in 2.6

$$\gamma = \cos^{-1} \left( \frac{l_1^2 + l_2^2 - X_p^2 - Y_p^2}{2l_1l_2} \right) \quad (2.7)$$

and it is clear from figure 2.6 that  $(\theta_2 = \pi - \gamma)$  then

$$\theta_2 = \pi - \gamma \quad (2.8)$$

Now we will find the formula to get  $\theta_1$

$$\beta = \tan^{-1} \left( \frac{Y_p}{X_p} \right) \quad (2.9)$$

from sines law

$$\frac{\sin(\gamma)}{C} = \frac{\sin(\alpha)}{l_2} \quad (2.10)$$

evaluating  $\alpha$

$$\alpha = \sin^{-1} \left( \frac{l_2 \sin(\gamma)}{C} \right) \quad (2.11)$$

i.e

$$\theta_1 = \beta - \alpha \quad (2.12)$$

# Chapter 3

## Kinetics of the robot

### 3.1 Introduction

To be able to deal with the manipulator forward and to complete the mission is conversion all things we talk about in the chapter 2 to mathematical model to go forward toward with it. First we will do a simple mathematical model to know the general behaviour of it, then we derive the mathematical model with addition of more thing to concluded.

### 3.2 Mathematical model

We use Lagrange method to find the equation of motion for the given system. Notice the both figures 2.5 and 3.1. The two robot's links have the following characteristics: Link 1: length  $l_1$ , total mass  $m_1$ , moment of inertia  $J_1$  and center of gravity  $G_1$ . Link 2: length  $l_2$ , total mass  $m_2$ , moment of inertia  $J_2$  and center of gravity  $G_2$ .

It is assumed that the two links have a symmetric mass distribution so that the centre of gravity is in the middle of the link. In order to formulate the equations of motion the so-called Lagrange method can be used. This method defines a set of generalized coordinates that specify the position of the system uniquely. In this case the generalized coordinates are the two angles  $\theta_1$  and  $\theta_2$ . The Lagrangian is defined as the difference between the kinetic and potential energies,

$$L(\theta_1, \dot{\theta}_1, \theta_2, \dot{\theta}_2) = K - P \quad (3.1)$$

Where K is the kinetic energy of the system and P is the potential energy of the system.

The equations of motion can now be derived as the following Lagrange equations,

$$\frac{\partial}{\partial t} \left( \frac{\partial L}{\partial \dot{\theta}_1} \right) - \frac{\partial L}{\partial \theta_1} = \tau_1, \quad (3.2)$$

$$\frac{\partial}{\partial t} \left( \frac{\partial L}{\partial \dot{\theta}_2} \right) - \frac{\partial L}{\partial \theta_2} = \tau_2,$$

The kinetic energy of the total system is the sum of the kinetic energy of each link and this in turn is the of the kinetic energy due to linear motion of the centre of gravity and the kinetic energy due of the rotation of the link around the center of gravity, i.e.,

$$K = \frac{1}{2} m_1 v_{G_1}^2 + \frac{1}{2} J_1 \dot{\theta}_1^2 + \frac{1}{2} m_2 v_{G_2}^2 + \frac{1}{2} J_2 (\dot{\theta}_1 + \dot{\theta}_2)^2 \quad (3.3)$$

If it is assumed that the center of gravity is positioned in the middle of each of the links, the cartesian coordinates of the two centers are

$$(x_{G_1}, y_{G_1}) = \left( \frac{l_1}{2} \cos(\theta_1), \frac{l_1}{2} \sin(\theta_1) \right) \quad (3.4)$$

$$(x_{G_2}, y_{G_2}) = \left( l_1 \cos(\theta_1) + \frac{l_2}{2} \cos(\theta_1 + \theta_2), l_1 \sin(\theta_1) + \frac{l_2}{2} \sin(\theta_1 + \theta_2) \right)$$

The two velocity vectors will be

$$v_{G_1} = \frac{d(x_{G_1}, y_{G_1})}{dt} \quad (3.5)$$

$$v_{G_2} = \frac{d(x_{G_2}, y_{G_2})}{dt}$$

then by evaluating  $x_g, y_g$  we differentiate it with respect to time and substitute it 3.3 for both two links we got

$$K = \left( \frac{1}{2} m_1 \frac{l_1^2}{4} + \frac{1}{2} J_1 + \frac{1}{2} m_2 l_1^2 \right) \dot{\theta}_1^2 + \left( \frac{1}{2} m_2 \frac{l_2^2}{4} + \frac{1}{2} J_2 \right) (\dot{\theta}_1 + \dot{\theta}_2)^2 + \frac{1}{2} m_2 l_2 l_2 \cos(\theta_2) \dot{\theta}_1 (\dot{\theta}_1 + \dot{\theta}_2) \quad (3.6)$$

The other term in the Lagrangian is the potential energy of the robot links due to gravity,

$$P = m_1 g \frac{l_1}{2} \sin(\theta_1) + m_2 g \left( l_1 \sin(\theta_1) + \frac{l_2}{2} \sin(\theta_1 + \theta_2) \right) \quad (3.7)$$

substitute equation 3.3 and equation 3.7 on equation 3.2 the resultant is the dynamic of the system expressed in 3.8

$$M_{11} \ddot{\theta}_1 + M_{12} \ddot{\theta}_2 + K_1(\theta_2, \dot{\theta}_1, \dot{\theta}_2) + G_1(\theta_1, \theta_2) = \tau_1, \quad (3.8)$$

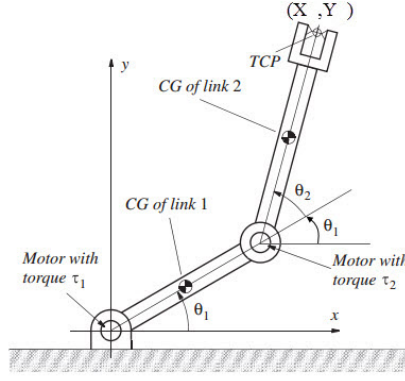


Figure 3.1: Manipulator diagram

$$M_{11}\ddot{\theta}_1 + M_{12}\ddot{\theta}_2 + K_1(\theta_2, \dot{\theta}_1, \dot{\theta}_2) + G_1(\theta_1, \theta_2) = \tau_1,$$

where the inertial components are,

$$M_{11} = \left(\frac{1}{4}m_1 + m_2\right)l_1^2 + \frac{1}{4}m_2l_2^2 + J_1 + J_2 + m_2l_1l_2\cos(\theta_2), \quad (3.9)$$

$$M_{12} = M_{21} = \frac{1}{4}m_2l_2^2 + J_2 + \frac{1}{2}m_2l_1l_2\cos(\theta_2),$$

$$M_{22} = \frac{1}{4}m_2l_2^2 + J_2,$$

$$M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}$$

The centrifugal and Coriolis force components are,

$$K_1(\theta_2, \dot{\theta}_1, \dot{\theta}_2) = -\frac{1}{2}m_2l_1l_2\sin(\theta_2) \cdot \dot{\theta}_2(2\dot{\theta}_1 + \dot{\theta}_2), \quad (3.10)$$

$$K_2(\theta_2, \dot{\theta}_1, \dot{\theta}_2) = \frac{1}{2}m_2l_1l_2\sin(\theta_2) \cdot \dot{\theta}_1^2$$

Finally, the terms due to gravity are,

$$G_1(\theta_1, \theta_2) = m_1g\frac{l_1}{2}\cos(\theta_1) + m_2g\left(l_1\cos(\theta_1) + \frac{l_2}{2}\cos(\theta_1 + \theta_2)\right), \quad (3.11)$$

$$G_2(\theta_1, \theta_2) = m_2g\frac{l_2}{2}\cos(\theta_1 + \theta_2). \quad (3.12)$$

Actually the mathematical model of the manipulator does not consider any hanging mass on any of the links, but there is some actuator for controlling -there is applied torque- the upper link and this actuator has a mass, that mean it will change the mathematical model

so we will consider this point and modifying the simple one to get the detailed one. The changing on the equation 3.3 is that there new kinetic energy for the motor mass must included.

$$K^* = \frac{1}{2}m_M v_M^2 \quad (3.13)$$

Now the position of the motor is depend on the angle of the first actuator, equation 3.14 express that position.

$$(x_M, y_M) = (l_1 \cos(\theta_1), l_1 \sin(\theta_1)) \quad (3.14)$$

After the differentiation of it with respect to time and substituting it on equation 3.13 the equation will be

$$K^* = \frac{1}{2}m_M l_1^2 \dot{\theta}_1^2 \quad (3.15)$$

So that we consider the kinetic energy changing from motor onto the model.

So far now we must consider the changing in potential energy.

$$P^* = m_M g l_1 \sin(\theta_1) \quad (3.16)$$

Now for the new Kinetic and potential energy caused by the motor weight and moving we apply Lagrange for that, on other words we will substitute equation 3.16 and equation 3.16 in 3.2 so

$$L^* = (K^* - P^*) \quad (3.17)$$

$$L^* = \frac{1}{2}m_M l_1^2 \dot{\theta}_1^2 - m_M g l_1 \sin(\theta_1)$$

$$\frac{\partial L^*}{\partial \dot{\theta}_1} = m_M l_1^2 \dot{\theta}_1$$

$$\frac{\partial L^*}{\partial \theta_1} = -m_M g l_1 \cos(\theta_1)$$

$$\frac{\partial}{\partial t} \left( \frac{\partial L^*}{\partial \dot{\theta}_1} \right) = m_M l_1^2 \ddot{\theta}_1 \tau^* = m_M l_1^2 \ddot{\theta}_1 + m_M g l_1 \cos(\theta_1)$$

so that

$$\tau_1^* = m_M l_1^2 \ddot{\theta}_1 + m_M g l_1 \cos(\theta_1)$$

$$\tau_2^* = 0$$



It is clear that the hanging motor on the upper joint will not affect the required torque for the upper link, the lower motor torque will increase because that one afford the hanging mass.

Now the full mathematical model expressed by so equation 3.8 will not effected only values of  $M_{11}$  and  $G_1(\theta_1, \theta_2)$  will change and became as equation 3.18.

$$M_{11} = \left(\frac{1}{4}m_1 + m_2\right) l_1^2 + \frac{1}{4}m_2 l_2^2 + J_1 + J_2 + m_2 l_1 l_2 \cos(\theta_2) + m_M l_1^2 \quad (3.18)$$

$$G_1(\theta_1, \theta_2) = m_1 g \frac{l_1}{2} \cos(\theta_1) + m_2 g \left( l_1 \cos(\theta_1) + \frac{l_2}{2} \cos(\theta_1 + \theta_2) \right) + m_M g l_1 \cos(\theta_1)$$

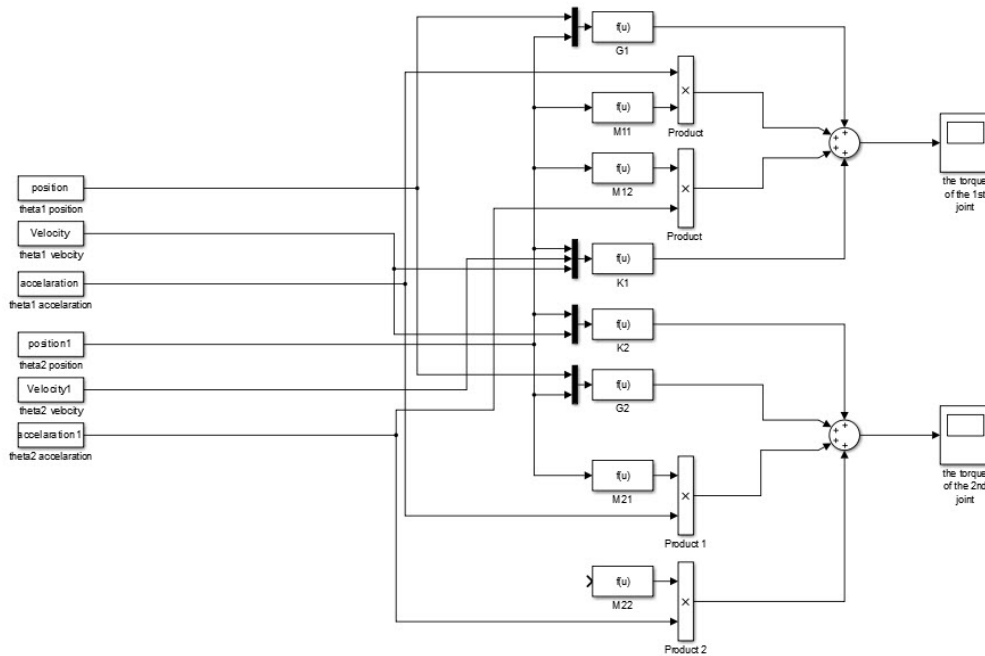


Figure 3.2: The Non-linear model using matlab simulation

# Chapter 4

## Trajectory generation

### 4.1 Introduction

the basic problem of this chapter is to move the manipulator arm from initial position to some desired final position with respect to reference frame. so what want to plan in this chapter is the trajectory; the trajectory refers to a time history of position, and acceleration for each of the DOF; planning each of DOF independently and then assume that the motion is realizable as hole.[2]

### 4.2 Point-to-point motion

In point-to-point motion, the manipulator has to move from an initial to an final joint configuration in a given time  $t_f$ . In this case, the actual end-effector path is of no concern. in our approach we consider Cubic Polynomial function to create the trajectory for point-to-point motion then substitute into the dynamics equation of robot to the maximum torque om each joint (to move between farthest points in highest moment of inertia).[1]

#### 4.2.1 Cubic polynomials

Consider the problem of moving the end-effector from initial position to final position in a particular time. the set of goal joint angles can be calculated using the inverse kinematics. the initial position of manipulator is also know in the form of a set of the joints angle. To make a smooth motion between the initial and final position of the manipulator. we wish first to generate a polynomial joint trijectory between tow configurations, and we wish specify the start and end velocity for the trajectory. this gives four constrains that the trajectory

must satisfy. two constrains on the function's value come from the selection of initial and final values:

$$\theta(0) = \theta_0 \quad (4.1)$$

$$\theta(t_f) = \theta_f$$

in the other construction come from the velocity. if we need the function to be continuous in velocity, the initial and final velocity must be zeros:

$$\dot{\theta}(0) = 0 \quad (4.2)$$

$$\dot{\theta}(t_f) = 0$$

these four constrains can be satisfied by a polynomial of at least third degree. since a cubic polynomial has four independent coefficients, it can be made to satisfy these constrains. A cubic has the form

$$\theta(t) = a_0 + a_1t + a_2t^2 + a_3t^3 \quad (4.3)$$

and so the joint velocity and acceleration is given

$$\dot{\theta}(t) = a_1 + 2a_2t + 3a_3t^2 \quad (4.4)$$

$$\ddot{\theta}(t) = 2a_2 + 6a_3t$$

Combine the equation (4.3) and (4.4) with the constraints yields four equations in four unknowns:

$$\theta(0) = a_0 \quad (4.5)$$

$$\theta(t_f) = a_0 + a_1t_f + a_2t_f^2 + a_3t_f^3$$

$$\dot{\theta}(0) = a_1 = 0$$

$$\dot{\theta}(t_f) = a_1 + 2a_2t_f + 3a_3t_f^2 = 0$$

solving these equations for the  $a_i$  we obtain

$$a_0 = \theta_0 \quad (4.6)$$

$$a_1 = 0$$

$$a_2 = \frac{3}{t_f^2}(\theta_f - \theta_0)$$

$$a_3 = -\frac{2}{t_f^3}(\theta_f - \theta_0)$$

using (4.6) we can calculate the cubic polynomial that connects any initial joint angle position with any desired final position. this solution is for the case when the joint start and finish at zero velocity. figure 4.1 show cubic polynomial trajectory.

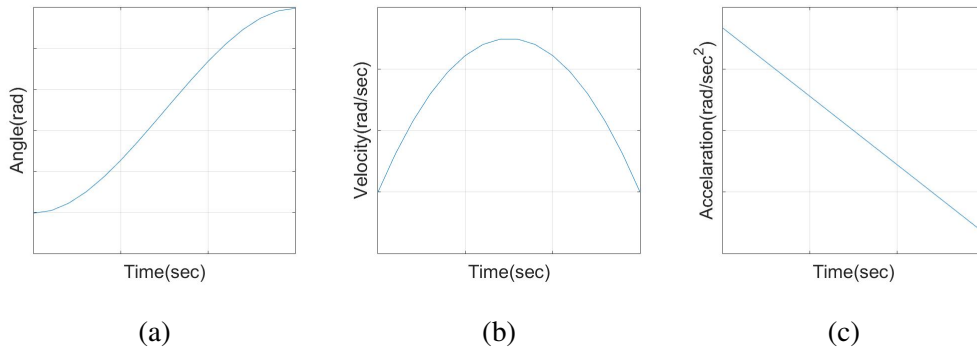


Figure 4.1: (a) Cubic polynomial trajectory.(b)Velocity profile for cubic polynomial trajectory.(c)Acceleration profile for cubic polynomial trajectory.

now we will apply this method to generate the trajectory for two paths (1 → 3 and 2 → 4, see figure 4.2) to calculate the maximum torque.

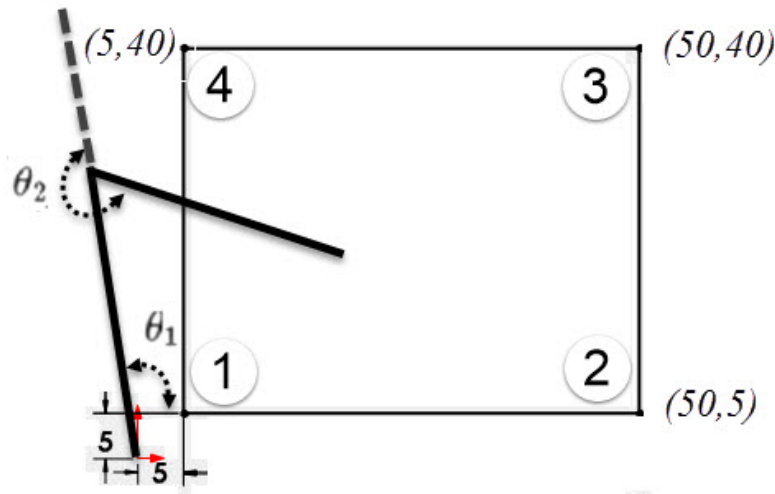


Figure 4.2: work-pace

To calculating the torque on each joint we have to move the end-effector from point to another on the work-space, starting with move it form 1 to 3, by using inverse kinematics we found ( $\theta_1 = 1.506rad$ ,  $\theta_2 = 3.275rad$ ) at position one and ( $\theta_1 = 1.183rad$ ,  $\theta_2 = 5.185rad$ ) at position three, (see section 2.4), so  $\theta_{10} = 1.506rad$ ,  $\theta_{1f} = 1.183rad$  and  $\theta_{20} = 3.275rad$ ,  $\theta_{2f} = 5.185rad$ , substitute them in equation 4.6 with  $t_f = 3.0sec$  then in 4.3 and 4.4 that will give

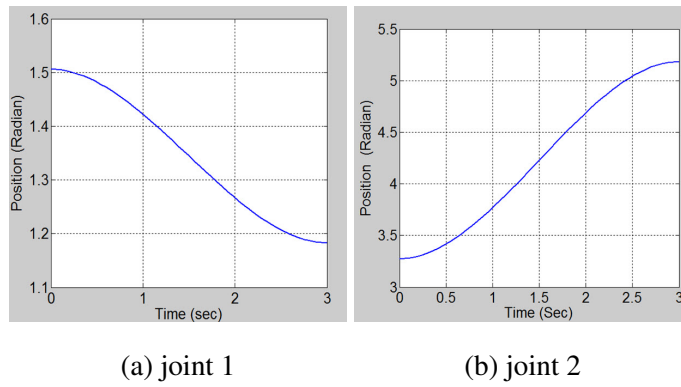


Figure 4.3: Cubic polynomial trajectory

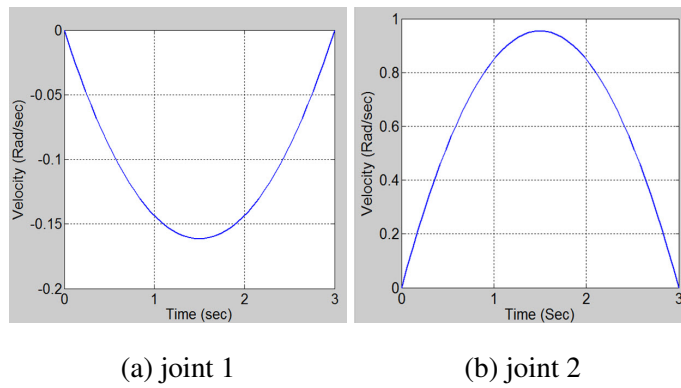


Figure 4.4: Velocity profile for cubic polynomial trajectory

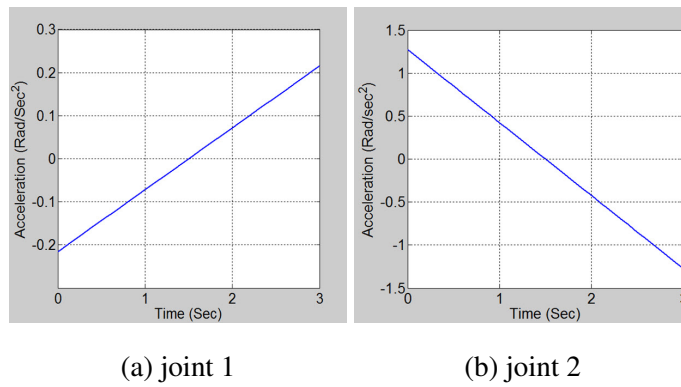


Figure 4.5: Acceleration profile for cubic polynomial trajectory

substitute these trajectories into the model by using matlab simulation of non-linear model (see figure 3.2), we found the torque on each joint as shown in figure 4.6.

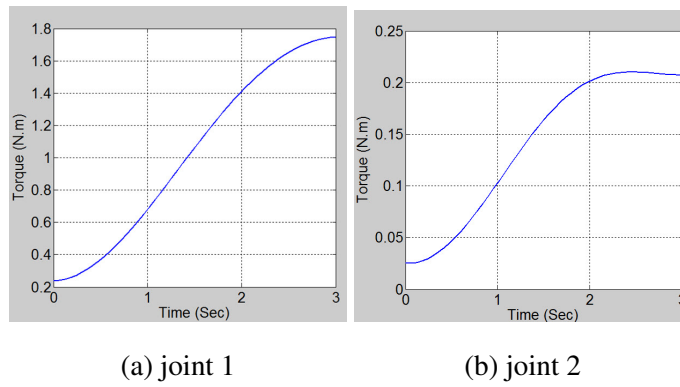


Figure 4.6: Torque curves for each joint for cubic polynomial trajectory

Now, we will repeat the previous steps for second path which is from point 2 to 4 (see figure 4.2),  $\theta_{10} = .865rad$ ,  $\theta_{1f} = 2.348rad$  and  $\theta_{20} = 4.605rad$ ,  $\theta_{2f} = 4.269rad$ , substitute them in equation 4.6 with  $t_f = 3.0sec$  also, then in 4.3 and 4.4 that will give

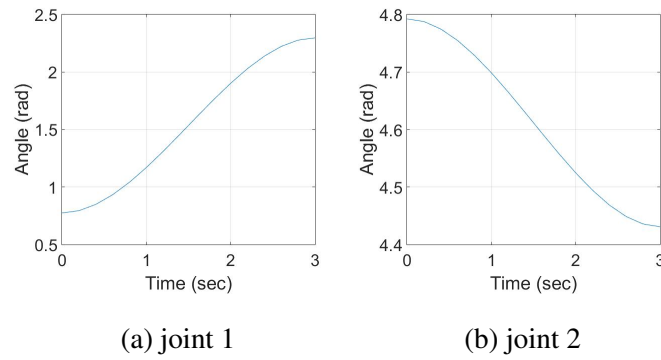


Figure 4.7: Cubic polynomial trajectory

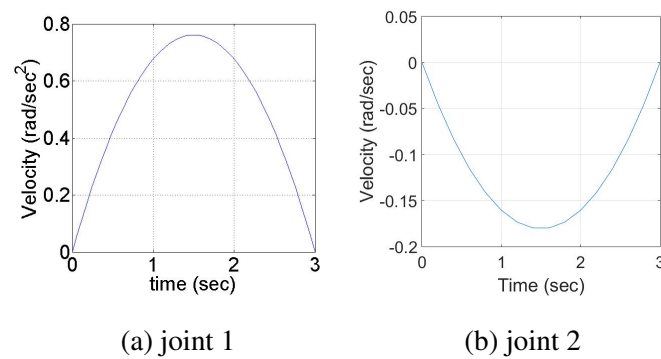


Figure 4.8: Velocity profile for cubic polynomial trajectory

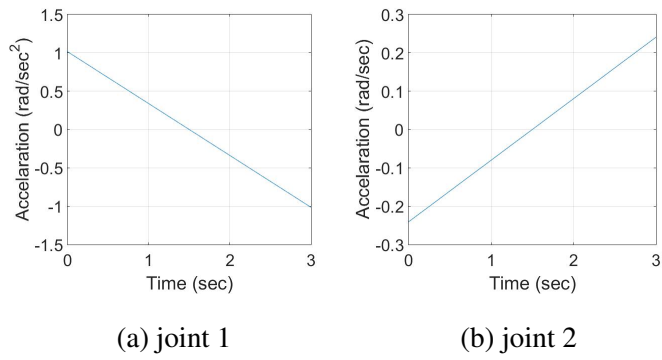


Figure 4.9: Acceleration profile for cubic polynomial trajectory

and by using the block diagram in matlab simulation (see figure 3.2)

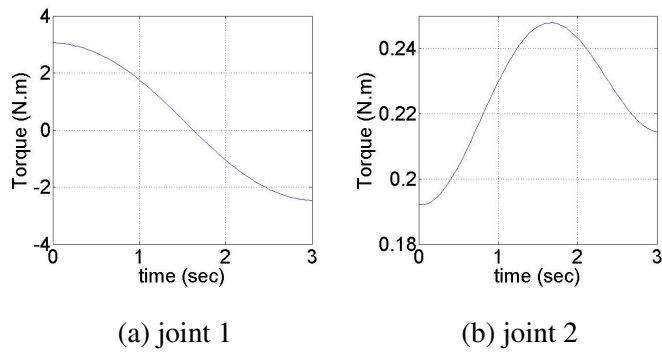


Figure 4.10: Torque curves for each joint for cubic polynomial trajectory

# Chapter 5

## Mechanical design

### 5.1 Stress analysis

So far, the system we need to calculate the stress on each link on it after we first assume the thickness is 10 mm and width of each link is 30 mm with shaft diameter 10 mm too. after these calculation we judge our assumptions if they meet our specification considering reasonable precise, robustness and performance.

About the material we choose Aluminium Alloy 6061-T6 which consider a light metal  $\rho = 2700\text{kg}/\text{m}^3$  with a 70 MPa yield strength and shear stress about 165 MPa which is suitable to a lot of applications

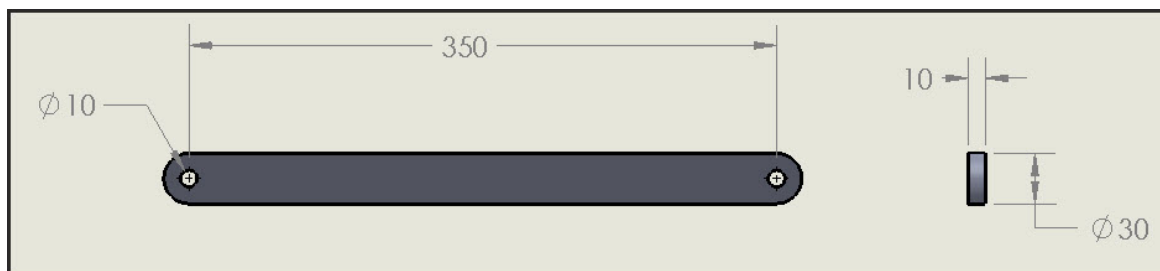


Figure 5.2: Upper link of manipulator projections

Now for the upper link this link is under torque form the second actuator with torque not to exceed 5 N.m . To guarantee our design is safe enough we find the stress on the most dangerous point which will be the hollow for the shaft. From strength of material we know the stress on bending beam is follow the 5.1.

$$\sigma = \frac{M.c}{I} \quad (5.1)$$



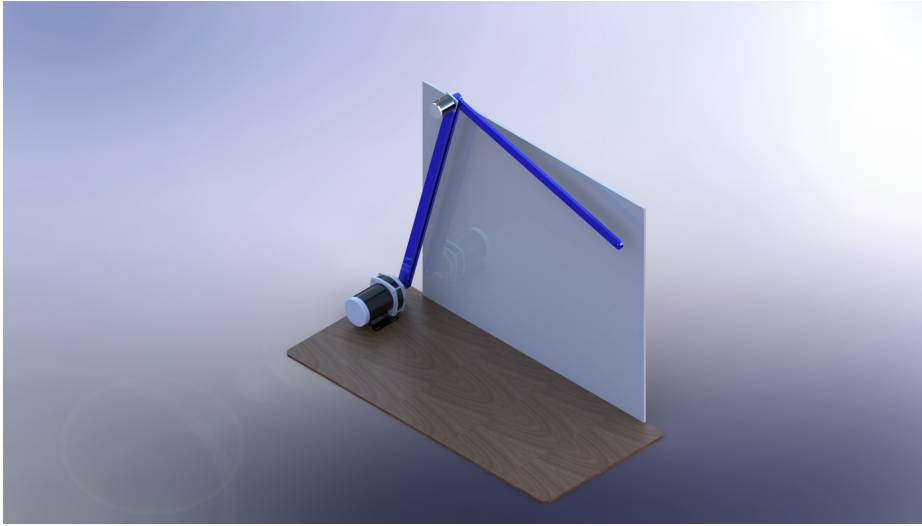


Figure 5.1: Isometric view for the CAD design of our manipulator

where  $M$  is the applied moment (Torque).

$c$  is the neutral axis of the cross sectional area of the area.

$I$  is Moment of Inertia.

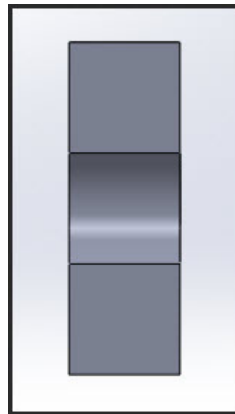


Figure 5.3: The cross sectional area for moment of Inertia

For cross sectional area that we take moment of inertia is expressed in 5.2 equation.

$$I = \frac{1}{12}bh_1^3 - \frac{1}{12}bh_2^3 \quad (5.2)$$

where  $b$  is the depth of the beam  $b = 10mm$ .

$h_1$  is the total thickness for the beam  $h_1 = 30mm$ .

$h_2$  is the diameter from the actuator shaft  $h_2 = 10mm$ . Now substitute these values in 5.2 equation we got

$$I = \frac{1}{12}(10mm)(30mm)^3 - \frac{1}{12}(10mm)(10mm)^3 = 21666.7mm^4 = 2.167 \times 10^{-8} \quad (5.3)$$

substitute I in equation with c equal to the half of length because the shape is symmetrical c = 15 mm equation 5.1 then equals

$$\sigma = \frac{.25 \times 15 \times 10^{-3}}{2.16 \times 10^{-8}} = .18Mpa \quad (5.4)$$

the stress is too little.

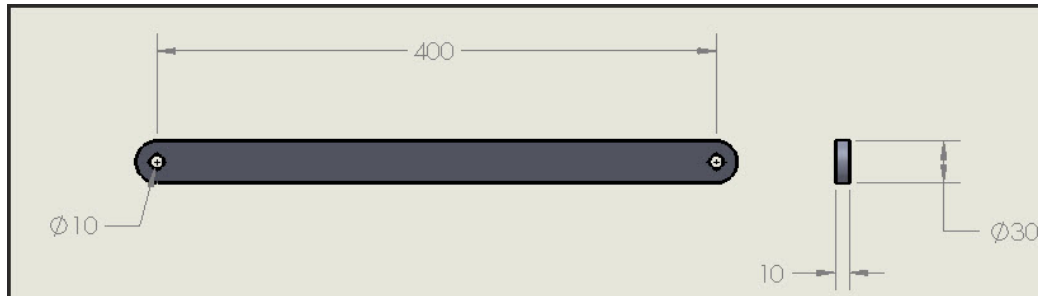


Figure 5.4: Lower link of manipulator projections

Now let us repeat the previous steps to find the stress for lower links from the section on the trajectory generation we notice that the maximum torque is less than 15 N.m this torque will be divided between the lower of two links so each one has 7.5 N.m and with the same geometry and sure same stress for both links. and the same moment of inertia as we find it for upper link. Now substitute 5.1. the result is

$$\sigma = \frac{4 \times 15 \times 10^{-3}}{2.16 \times 10^{-8}} = 2.8Mpa \quad (5.5)$$

and factor of safety is

$$FS = \frac{\sigma_{yeild}}{\sigma} = \frac{70}{5.21} = 24 \quad (5.6)$$

From the value we can conclude that our design is safe enough due to stress analysis, we can not reduce the thickness of the links because of availability of slabs that we form our links.

## 5.2 Deflection analysis

After we ensure our design is safe in terms of stress analysis, the deflection study is the second step and we need to ensure that the deflection of the system will not be large and will not effect other things of system like end-effector reached point. The worst possible case for that if we assume while link moving someone try to fix the beam, this case represented in figure 5.5. We notice that represent the cantilever beam. So we will write equation expresses that case. We perform free body diagram with a shear.

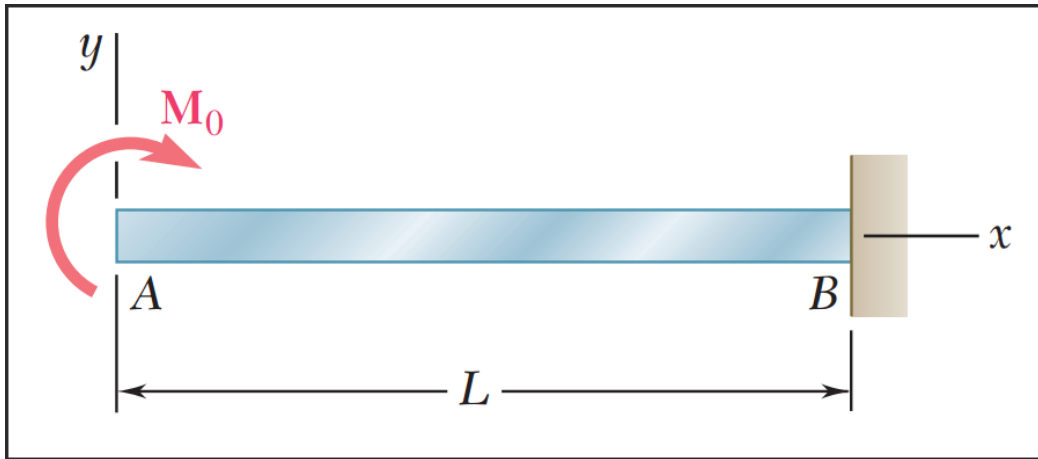


Figure 5.5: Diagram of the beam for deflection analysis

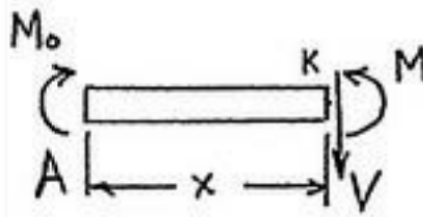


Figure 5.6: free body diagram with applied shear force

$$\Sigma M_k = 0 \Rightarrow M_o + M = 0 \quad (5.7)$$

$$M = M_o$$

$$EI \frac{d^2y}{dx^2} = M = M_o \quad (5.8)$$

$$EI \frac{dy}{dx} = M_o x + C_1$$

from the Conditions and after integration

$$\left[ x = L, \frac{dy}{dx} \right] : 0 = M_o L + C_1 \quad (5.9)$$

$$C_1 = -M_o L$$

Integration with respect to  $x$  again

$$Ely = \frac{1}{2} M_o x^2 + C_1 x + C_2$$

from the boundary condition

$$[x = L, y = 0] 0 = \frac{1}{2} M_o L^2 - M_o L^2 + C_2$$

then

$$C_2 = \frac{1}{2}M_0L^2$$

Then

$$y = \frac{M_0}{2EI}(x^2 - 2Lx + L^2)$$

by substituting the values of  $M_0 = 4Nm$ ,  $E = 70GPa$  and  $I = 2.167 \times 10^{-8}$  Now by Matlab solving this equation for  $0 \leq x \leq 0.4m$ .

We substitute these parameters on Matlab then maximum deflection will not exceed 3mm.

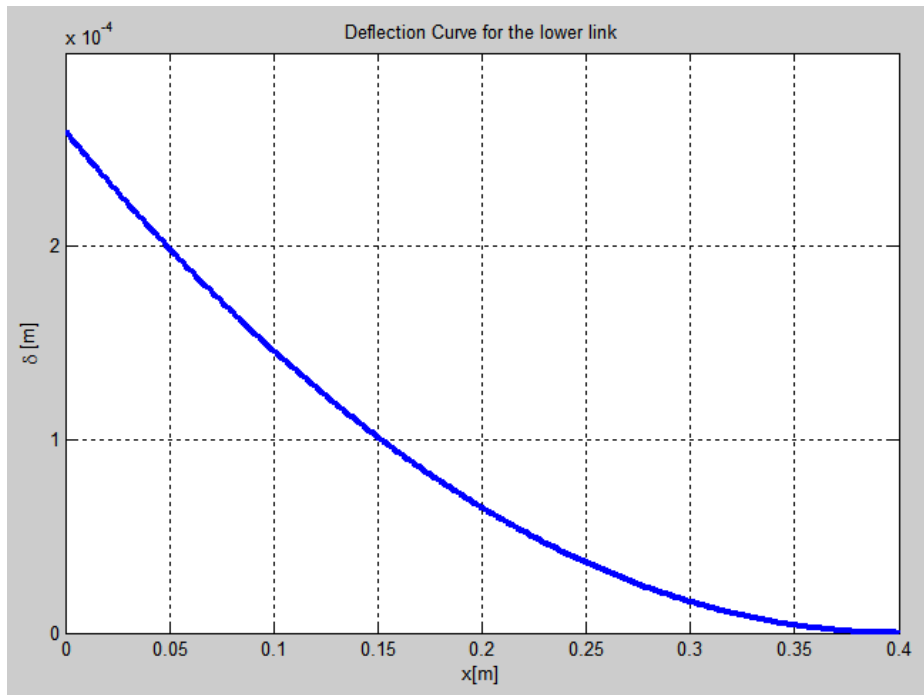


Figure 5.7: lower link deflection

For the upper link the same assumptions and aspects but  $M_0 = .25Nm$ ,  $L = .35m$  again on matlab the deflection will not exceed 2 mm.

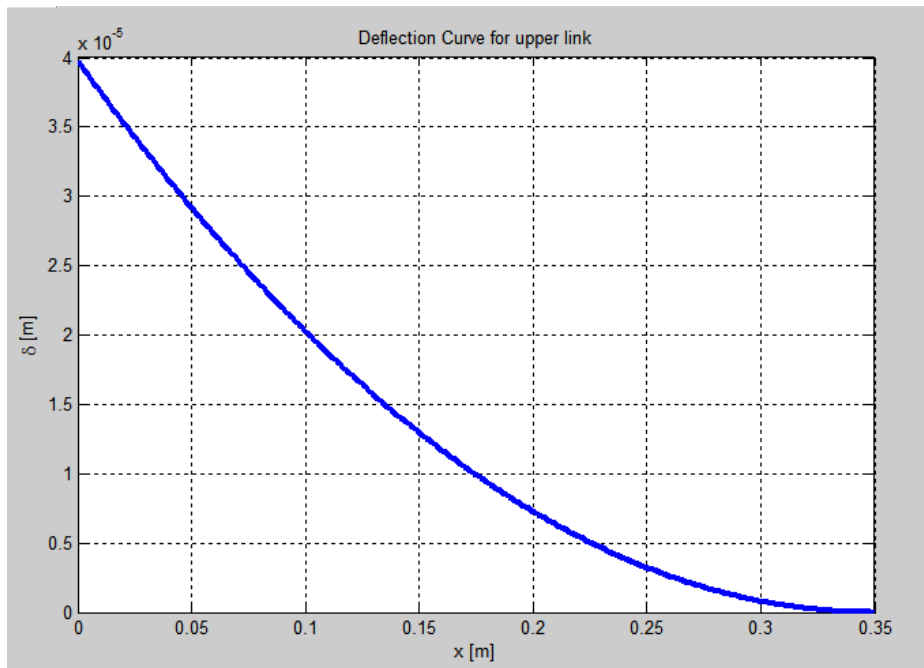


Figure 5.8: upper link deflection

Now after we grantee our design is safe we summarize the values of all parameters on table

Table 5.1: All parameters of the manipulator

	Symbol	Value
Lower link length	$L_1$	0.4 m
Upper link length	$L_2$	0.35 m
Gravitational acceleration	g	9.81 m <sup>2</sup> /s
Inertia of lower link	$j_1$	0.03456 kg.m <sup>2</sup>
Inertia of upper link	$j_2$	0.0058 kg.m <sup>2</sup>
Mass of upper link	$m_1$	0.1417 kg
Mass of lower link	$m_2$	0.648 kg
Upper motor and stand masses	$m_M$	0.5 kg

# Chapter 6

## Control Design and Hardware Implementation

### 6.1 Introduction

In this chapter we will demonstrate different types of controllers that we use, and all hardware and software that we implements on the projects, driver circuit for actuators ,rotary encoders , filters ,microcontroller and two types of controller PID and tracker.In this chapter also we will find Hardware gain of each actuator and its own driving circuit.never forget that our system is MIMO we have two input and two output and measurements.

### 6.2 Hardware components

#### 6.2.1 Mircontroller

There were many choices to use, like PIC micro controller from microchip or Atmel microcontrollers, but in this project Arduino board based on Atmel is used; because its open source and one can deal with it easily, thus the Arduino Mega 2560 has been chosen to control our manipulator.

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560. It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable to get started.

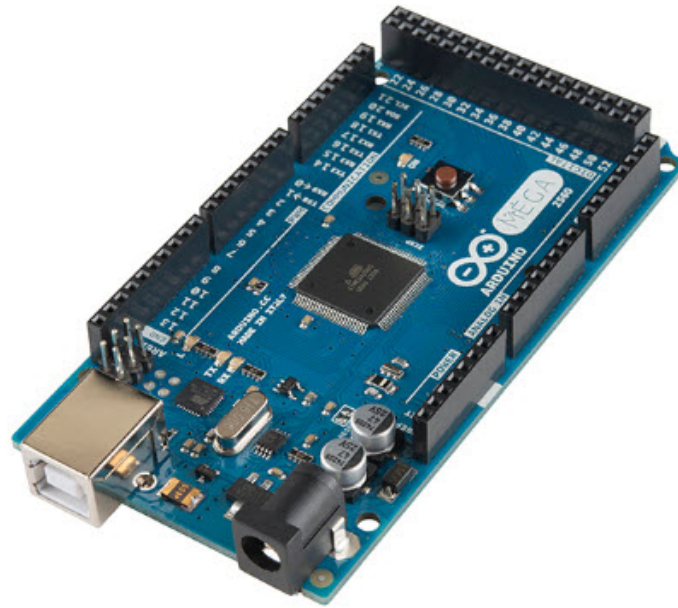


Figure 6.1: arduino mega 2560

Arduino provide a simple and comfortable software development to program and upload the code on the microcontroller Arduino 1.6.9

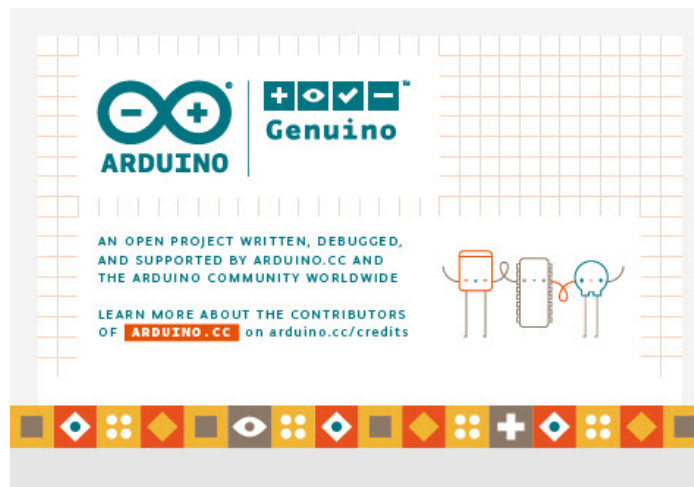


Figure 6.2: arduino software development tool

## 6.2.2 Rotary potentiometer

To obtain the current opened angel of each joint we prefer using rotary frictionless potentiometer it provides a reasonable accuracy and it is cheap and it rotate till 300 degree which is good enough for our project , we will not exceed 200 degree at any joint.

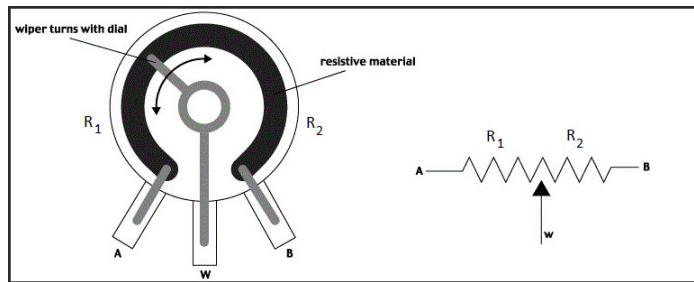


Figure 6.3: internal connection of potentiometer

### 6.2.3 Actuators

We have two actuators lower one is stepper motor with gear box and the upper one is DC motor (pittman) with gear box, stepper motor has 10 N.m peak torque after the gear box, the upper motor has approximately 1.5 N.m after gear box.



Figure 6.4: wiper motor





Figure 6.5: pitman motor with gear box

### 6.2.4 Motor drivers

After our experimentation (open loop) we measure the current flow in each one we get about 3 A for stepper motor and less than 2 A for Pittman motor and the operating voltage for the both is 12 V so that we choose our drivers based on these measurements. the best drive method for DC motor is PWM with switching circuit (H-Bridge), the H-bridge control the path of current so it can control direction of rotation and PWM modulation technique (after smoothing the signal)for speed.

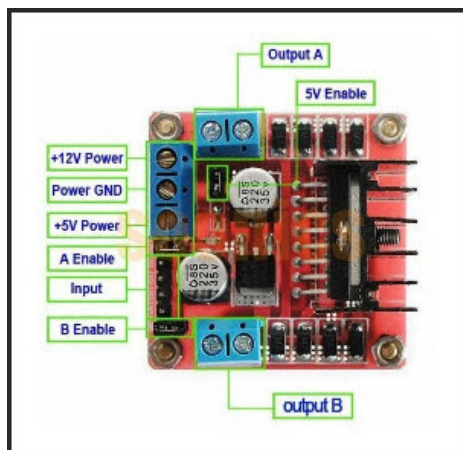


Figure 6.6: Upper motor driving circuit

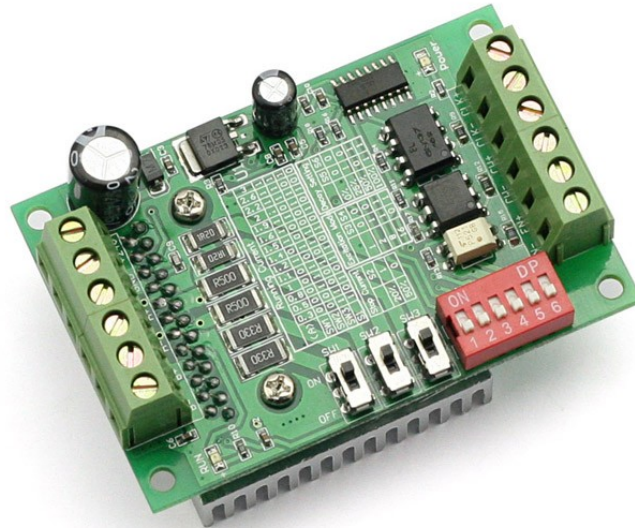


Figure 6.7: Lower motor driving circuit

figure 6.7 shows the driver circuit for wiper motor this H-bridge can afford upto 40 A and our motor consumes just less than 15 A and that is good.

## 6.3 Control design

### 6.3.1 PID controller

As we mentioned we will first implement PID controller for our manipulator ,In this technique we apply a PID for each actuator independently, we know that our system is nonlinear system so we apply PID and tuning the gains all these things after finding the dynamic of the Actuators with its own driver circuit and because the dynamic is very fast we can consider that equivalent to some gain.

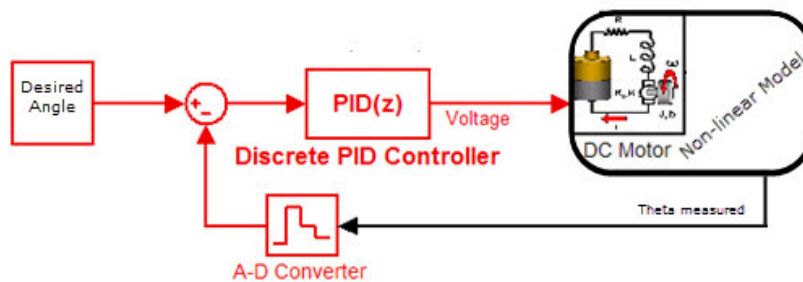


Figure 6.8: equivalent block diagram for motor and attached link

**finding hardware gain for the DC motor** we operate the system manually and rotate the links to 0 degree and we start increase the voltage flow and the motor start overcome the inertia was 1.8 v

$$\begin{aligned} \Sigma T &= 0 & (6.1) \\ +T_m - m_2 g \frac{l_2}{2} &= 0 \end{aligned}$$

by substitution values see (table 5.2)

$$T_m = .1908Nm$$

and

$$K_{fv} = 9.4$$

## 6.4 Tests and results

At first we used just a P controller for each motor, the inputs was (10, 10), (20, 20), (30, 30), (40, 30), (40, 20), (50, 20) the steady state error was very large 10cm because of the gear backlash in the lower motor.

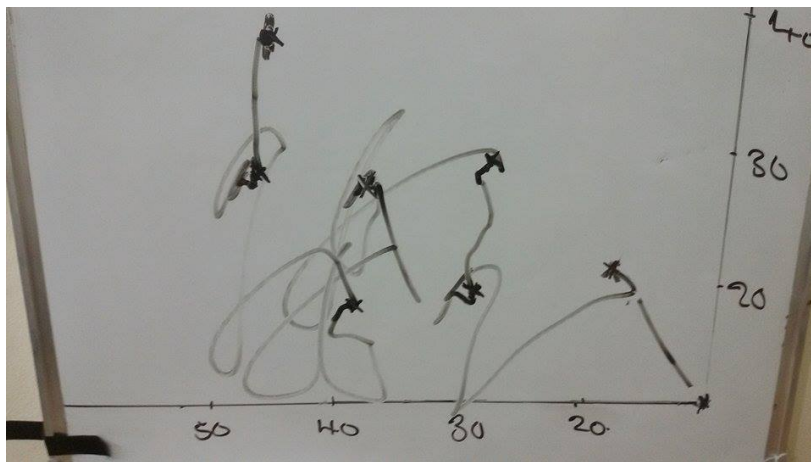


Figure 6.9: first trial

for repeatably test the result was as shown in figure 6.10 steady state error within 10cm.

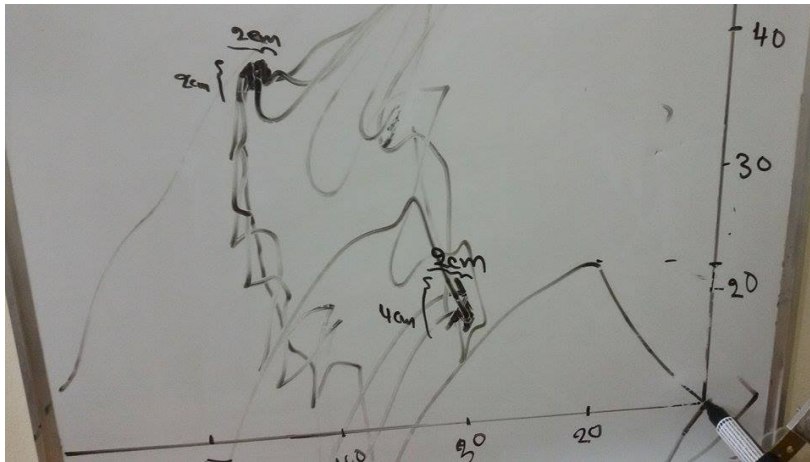


Figure 6.10: first trial

Then we change the lower motor and by adjusting the upper motor controller gains to be  $k_p = 2, k_d = 0.1, k_i = 0.05$  the response become better as shown :

### 1. For the Lower Motor

when theta one moved from 90 to 150 degree the response was as show in figure 6.11 , the settling time was 6.89 sec and the steady-state error is .3 degree.

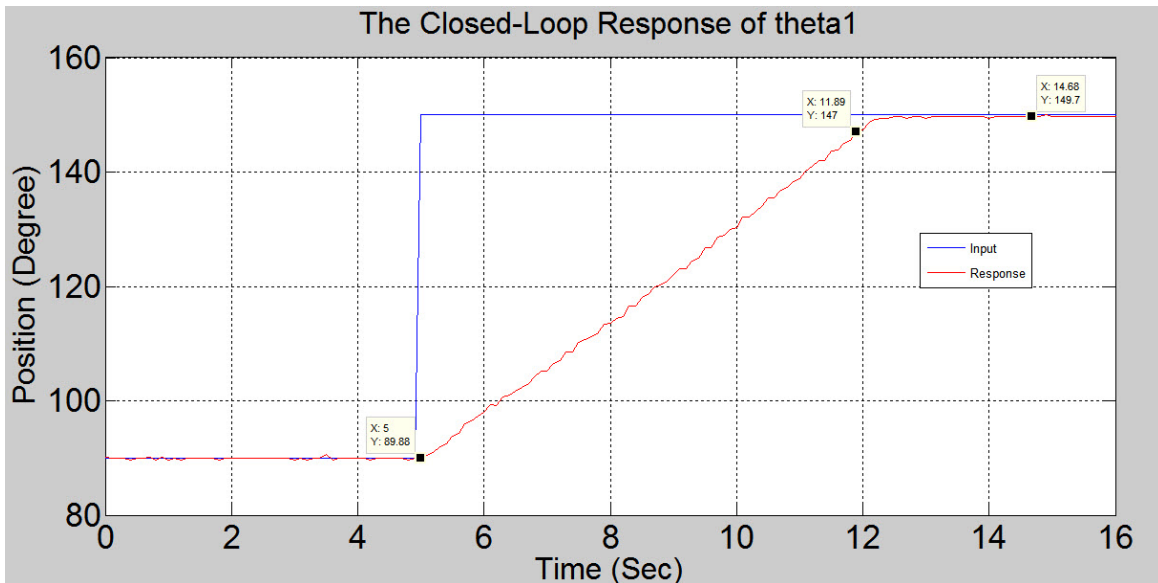


Figure 6.11: theta one move form 90 to 150 degree

and when it moves from 110 to 70 degree the settling time was 4.8 sec and the steady state error 0.15 degree.

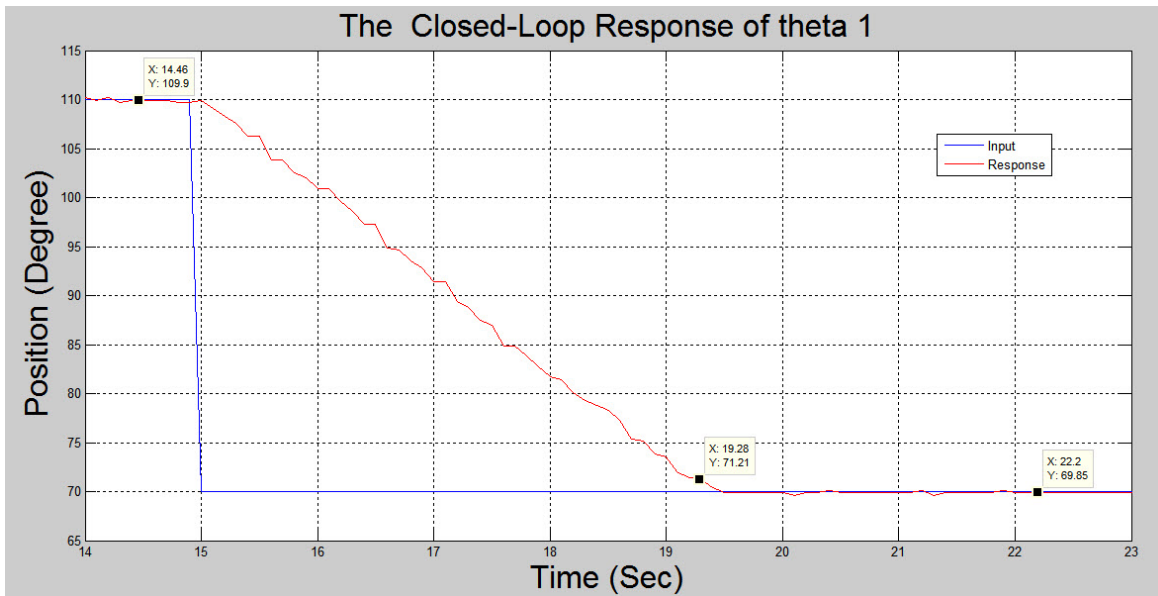


Figure 6.12: theta one move form 110 to 70 degree

## 2. For the upper Motor

when theta two moves form 150 to 10 degree the response was as shown in figure 6.13 , the settling time was 4.32 and the steady state error .3 degree.

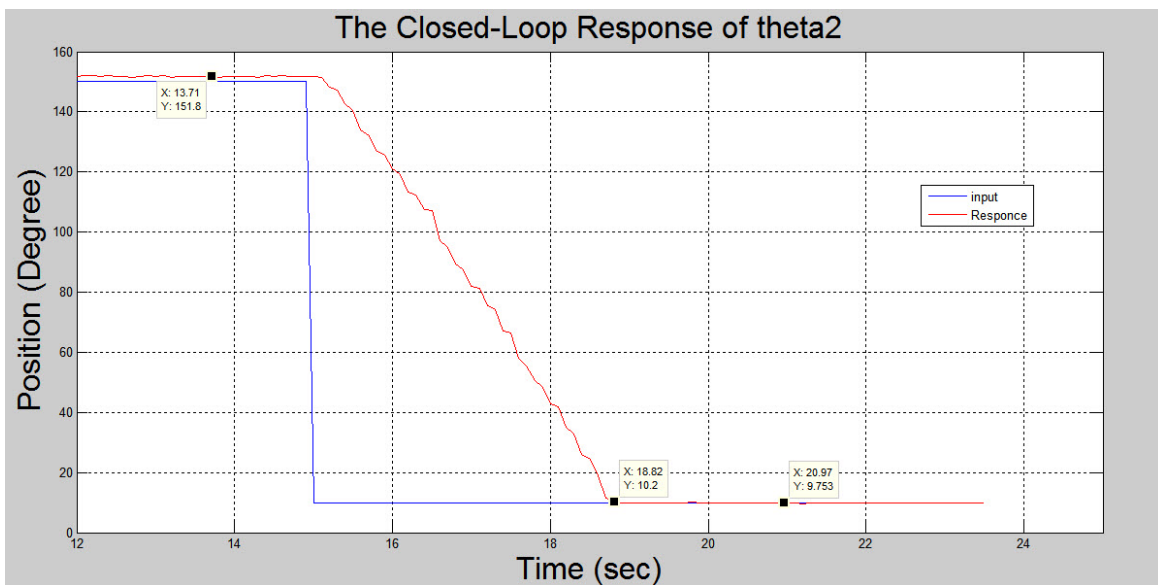


Figure 6.13: theta one move form 110 to 70 degree

then when it moves form 90 to 150 degree the settling time was 1.13 sec and the steady state error was 1.3 degree.

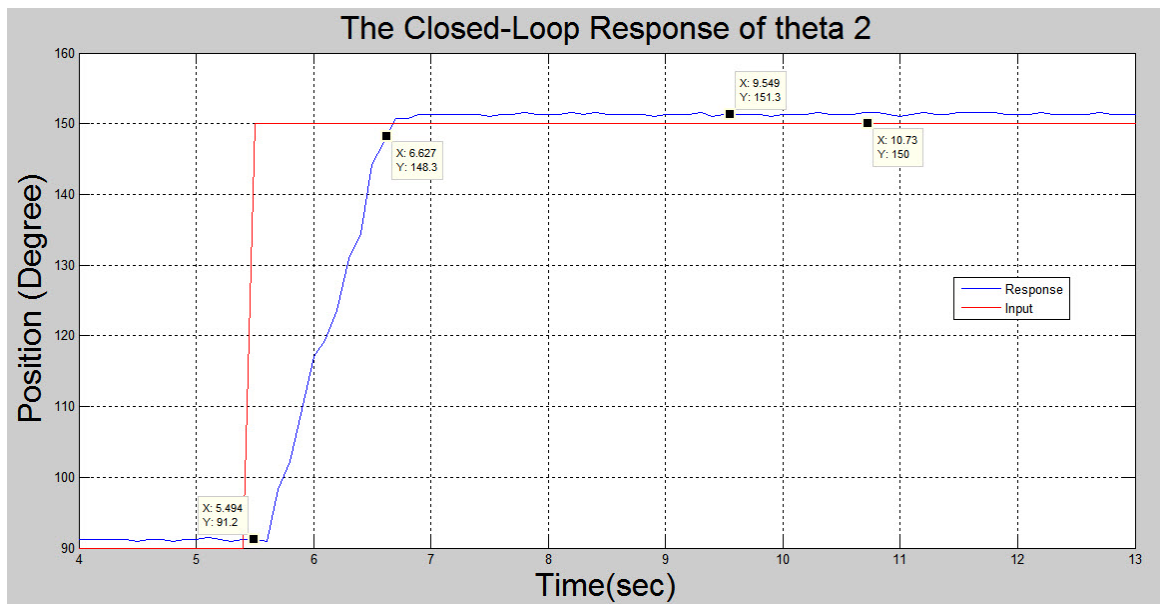


Figure 6.14: theta one move form 150 to 10 degree

# Bibliography

- [1] *Introduction To Robotics, Mechanics And Control*. 1989.
- [2] *Robot modeling and control*. 2006.
- [3] Vladimir Dukovski. Comparison of the characteristics between serial and parallel robots. *ACTA TEHNICA CORVINIENSIS*, 2014.

# Appendices

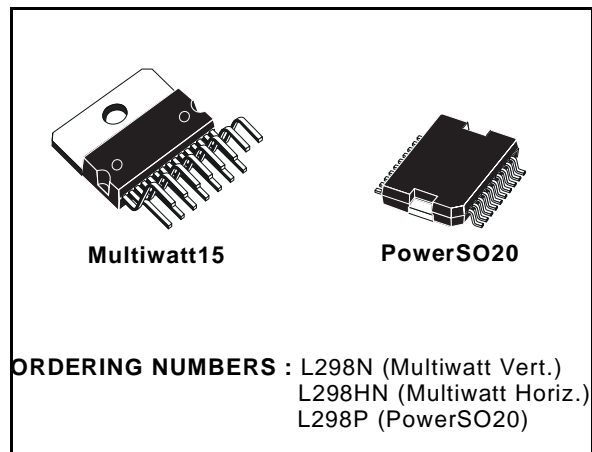


## DUAL FULL-BRIDGE DRIVER

- OPERATING SUPPLY VOLTAGE UP TO 46 V
- TOTAL DC CURRENT UP TO 4 A
- LOW SATURATION VOLTAGE
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)

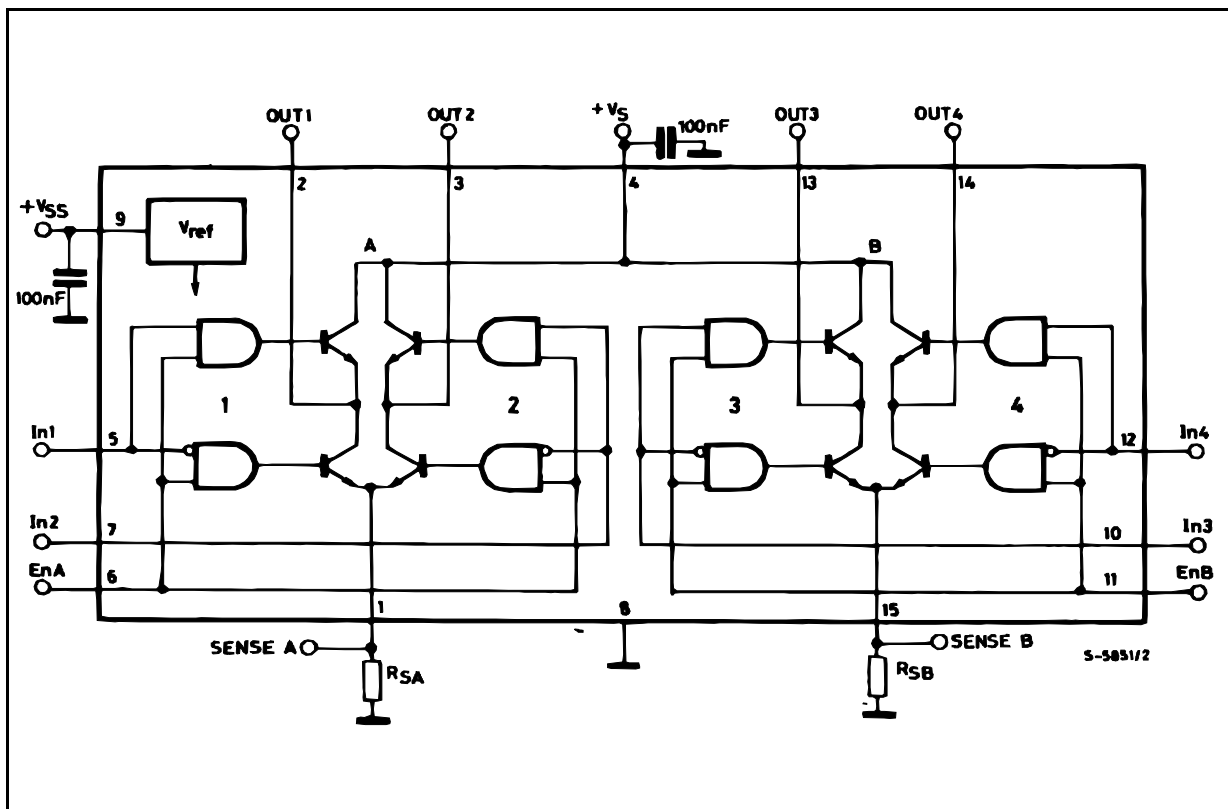
### DESCRIPTION

The L298 is an integrated monolithic circuit in a 15-lead Multiwatt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the con-



nection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.

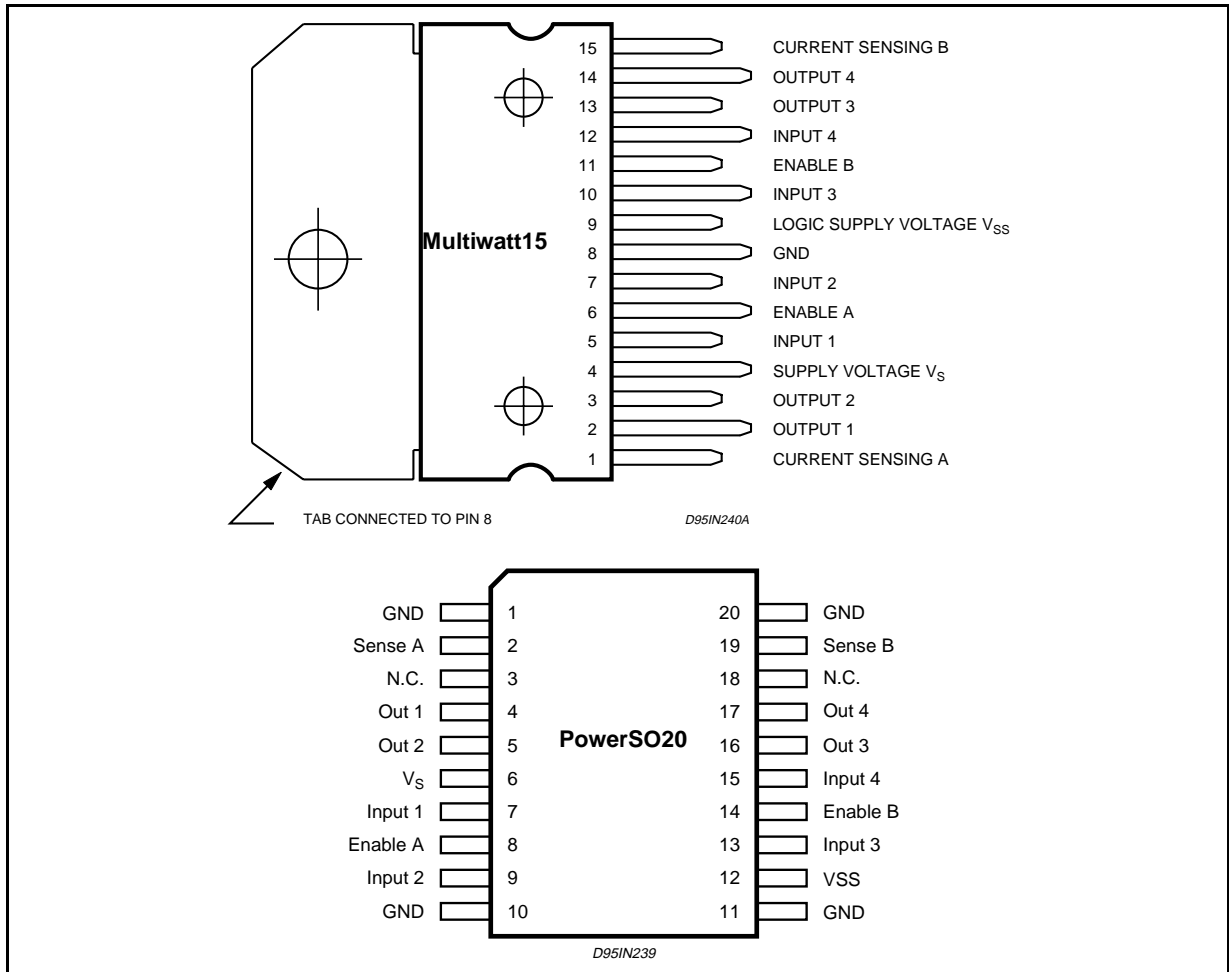
### BLOCK DIAGRAM



**ABSOLUTE MAXIMUM RATINGS**

Symbol	Parameter	Value	Unit
$V_S$	Power Supply	50	V
$V_{SS}$	Logic Supply Voltage	7	V
$V_I, V_{En}$	Input and Enable Voltage	-0.3 to 7	V
$I_O$	Peak Output Current (each Channel)		
	- Non Repetitive ( $t = 100\mu s$ )	3	A
	- Repetitive (80% on -20% off; $t_{on} = 10ms$ )	2.5	A
	-DC Operation	2	A
$V_{sens}$	Sensing Voltage	-1 to 2.3	V
$P_{tot}$	Total Power Dissipation ( $T_{case} = 75^\circ C$ )	25	W
$T_{op}$	Junction Operating Temperature	-25 to 130	$^\circ C$
$T_{stg}, T_j$	Storage and Junction Temperature	-40 to 150	$^\circ C$

**PIN CONNECTIONS (top view)**



**THERMAL DATA**

Symbol	Parameter		PowerSO20	Multiwatt15	Unit
$R_{th\ j-case}$	Thermal Resistance Junction-case	Max.	-	3	$^\circ C/W$
$R_{th\ j-amb}$	Thermal Resistance Junction-ambient	Max.	13 (*)	35	$^\circ C/W$

(\*) Mounted on aluminum substrate



## PIN FUNCTIONS (refer to the block diagram)

MW.15	PowerSO	Name	Function
1;15	2;19	Sense A; Sense B	Between this pin and ground is connected the sense resistor to control the current of the load.
2;3	4;5	Out 1; Out 2	Outputs of the Bridge A; the current that flows through the load connected between these two pins is monitored at pin 1.
4	6	V <sub>s</sub>	Supply Voltage for the Power Output Stages. A non-inductive 100nF capacitor must be connected between this pin and ground.
5;7	7;9	Input 1; Input 2	TTL Compatible Inputs of the Bridge A.
6;11	8;14	Enable A; Enable B	TTL Compatible Enable Input: the L state disables the bridge A (enable A) and/or the bridge B (enable B).
8	1,10,11,20	GND	Ground.
9	12	V <sub>SS</sub>	Supply Voltage for the Logic Blocks. A100nF capacitor must be connected between this pin and ground.
10; 12	13;15	Input 3; Input 4	TTL Compatible Inputs of the Bridge B.
13; 14	16;17	Out 3; Out 4	Outputs of the Bridge B. The current that flows through the load connected between these two pins is monitored at pin 15.
–	3;18	N.C.	Not Connected

ELECTRICAL CHARACTERISTICS (V<sub>S</sub> = 42V; V<sub>SS</sub> = 5V, T<sub>j</sub> = 25°C; unless otherwise specified)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
V <sub>S</sub>	Supply Voltage (pin 4)	Operative Condition	V <sub>IH</sub> +2.5		46	V
V <sub>SS</sub>	Logic Supply Voltage (pin 9)		4.5	5	7	V
I <sub>S</sub>	Quiescent Supply Current (pin 4)	V <sub>en</sub> = H; I <sub>L</sub> = 0 V <sub>i</sub> = L V <sub>i</sub> = H		13 50	22 70	mA mA
		V <sub>en</sub> = L V <sub>i</sub> = X			4	mA
I <sub>SS</sub>	Quiescent Current from V <sub>SS</sub> (pin 9)	V <sub>en</sub> = H; I <sub>L</sub> = 0 V <sub>i</sub> = L V <sub>i</sub> = H		24 7	36 12	mA mA
		V <sub>en</sub> = L V <sub>i</sub> = X			6	mA
V <sub>iL</sub>	Input Low Voltage (pins 5, 7, 10, 12)		–0.3		1.5	V
V <sub>iH</sub>	Input High Voltage (pins 5, 7, 10, 12)		2.3		V <sub>SS</sub>	V
I <sub>iL</sub>	Low Voltage Input Current (pins 5, 7, 10, 12)	V <sub>i</sub> = L			–10	μA
I <sub>iH</sub>	High Voltage Input Current (pins 5, 7, 10, 12)	V <sub>i</sub> = H ≤ V <sub>SS</sub> –0.6V		30	100	μA
V <sub>en</sub> = L	Enable Low Voltage (pins 6, 11)		–0.3		1.5	V
V <sub>en</sub> = H	Enable High Voltage (pins 6, 11)		2.3		V <sub>SS</sub>	V
I <sub>en</sub> = L	Low Voltage Enable Current (pins 6, 11)	V <sub>en</sub> = L			–10	μA
I <sub>en</sub> = H	High Voltage Enable Current (pins 6, 11)	V <sub>en</sub> = H ≤ V <sub>SS</sub> –0.6V		30	100	μA
V <sub>CEsat</sub> (H)	Source Saturation Voltage	I <sub>L</sub> = 1A I <sub>L</sub> = 2A	0.95	1.35 2	1.7 2.7	V V
V <sub>CEsat</sub> (L)	Sink Saturation Voltage	I <sub>L</sub> = 1A (5) I <sub>L</sub> = 2A (5)	0.85	1.2 1.7	1.6 2.3	V V
V <sub>CEsat</sub>	Total Drop	I <sub>L</sub> = 1A (5) I <sub>L</sub> = 2A (5)	1.80		3.2 4.9	V V
V <sub>sens</sub>	Sensing Voltage (pins 1, 15)		–1 (1)		2	V

ELECTRICAL CHARACTERISTICS (continued)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
T <sub>1</sub> (V <sub>i</sub> )	Source Current Turn-off Delay	0.5 V <sub>i</sub> to 0.9 I <sub>L</sub> (2); (4)		1.5		μs
T <sub>2</sub> (V <sub>i</sub> )	Source Current Fall Time	0.9 I <sub>L</sub> to 0.1 I <sub>L</sub> (2); (4)		0.2		μs
T <sub>3</sub> (V <sub>i</sub> )	Source Current Turn-on Delay	0.5 V <sub>i</sub> to 0.1 I <sub>L</sub> (2); (4)		2		μs
T <sub>4</sub> (V <sub>i</sub> )	Source Current Rise Time	0.1 I <sub>L</sub> to 0.9 I <sub>L</sub> (2); (4)		0.7		μs
T <sub>5</sub> (V <sub>i</sub> )	Sink Current Turn-off Delay	0.5 V <sub>i</sub> to 0.9 I <sub>L</sub> (3); (4)		0.7		μs
T <sub>6</sub> (V <sub>i</sub> )	Sink Current Fall Time	0.9 I <sub>L</sub> to 0.1 I <sub>L</sub> (3); (4)		0.25		μs
T <sub>7</sub> (V <sub>i</sub> )	Sink Current Turn-on Delay	0.5 V <sub>i</sub> to 0.9 I <sub>L</sub> (3); (4)		1.6		μs
T <sub>8</sub> (V <sub>i</sub> )	Sink Current Rise Time	0.1 I <sub>L</sub> to 0.9 I <sub>L</sub> (3); (4)		0.2		μs
f <sub>c</sub> (V <sub>i</sub> )	Commutation Frequency	I <sub>L</sub> = 2A		25	40	KHz
T <sub>1</sub> (V <sub>en</sub> )	Source Current Turn-off Delay	0.5 V <sub>en</sub> to 0.9 I <sub>L</sub> (2); (4)		3		μs
T <sub>2</sub> (V <sub>en</sub> )	Source Current Fall Time	0.9 I <sub>L</sub> to 0.1 I <sub>L</sub> (2); (4)		1		μs
T <sub>3</sub> (V <sub>en</sub> )	Source Current Turn-on Delay	0.5 V <sub>en</sub> to 0.1 I <sub>L</sub> (2); (4)		0.3		μs
T <sub>4</sub> (V <sub>en</sub> )	Source Current Rise Time	0.1 I <sub>L</sub> to 0.9 I <sub>L</sub> (2); (4)		0.4		μs
T <sub>5</sub> (V <sub>en</sub> )	Sink Current Turn-off Delay	0.5 V <sub>en</sub> to 0.9 I <sub>L</sub> (3); (4)		2.2		μs
T <sub>6</sub> (V <sub>en</sub> )	Sink Current Fall Time	0.9 I <sub>L</sub> to 0.1 I <sub>L</sub> (3); (4)		0.35		μs
T <sub>7</sub> (V <sub>en</sub> )	Sink Current Turn-on Delay	0.5 V <sub>en</sub> to 0.9 I <sub>L</sub> (3); (4)		0.25		μs
T <sub>8</sub> (V <sub>en</sub> )	Sink Current Rise Time	0.1 I <sub>L</sub> to 0.9 I <sub>L</sub> (3); (4)		0.1		μs

- 1) Sensing voltage can be -1 V for t ≤ 50 μsec; in steady state V<sub>sens</sub> min ≥ -0.5 V.
- 2) See fig. 2.
- 3) See fig. 4.
- 4) The load must be a pure resistor.

Figure 1 : Typical Saturation Voltage vs. Output Current.

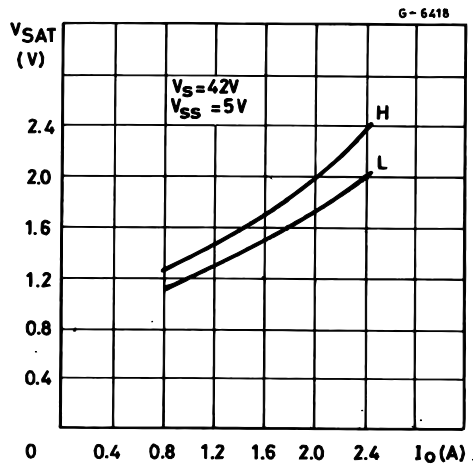
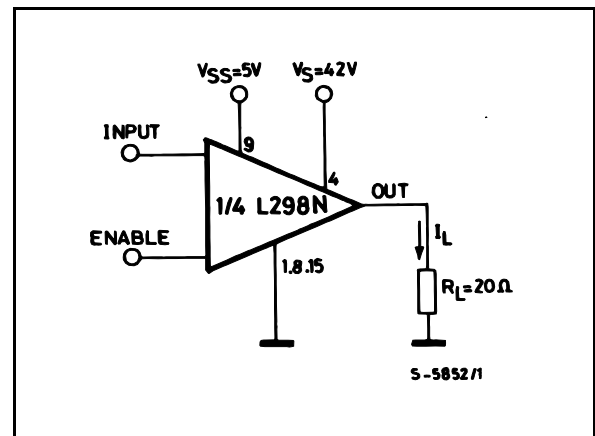
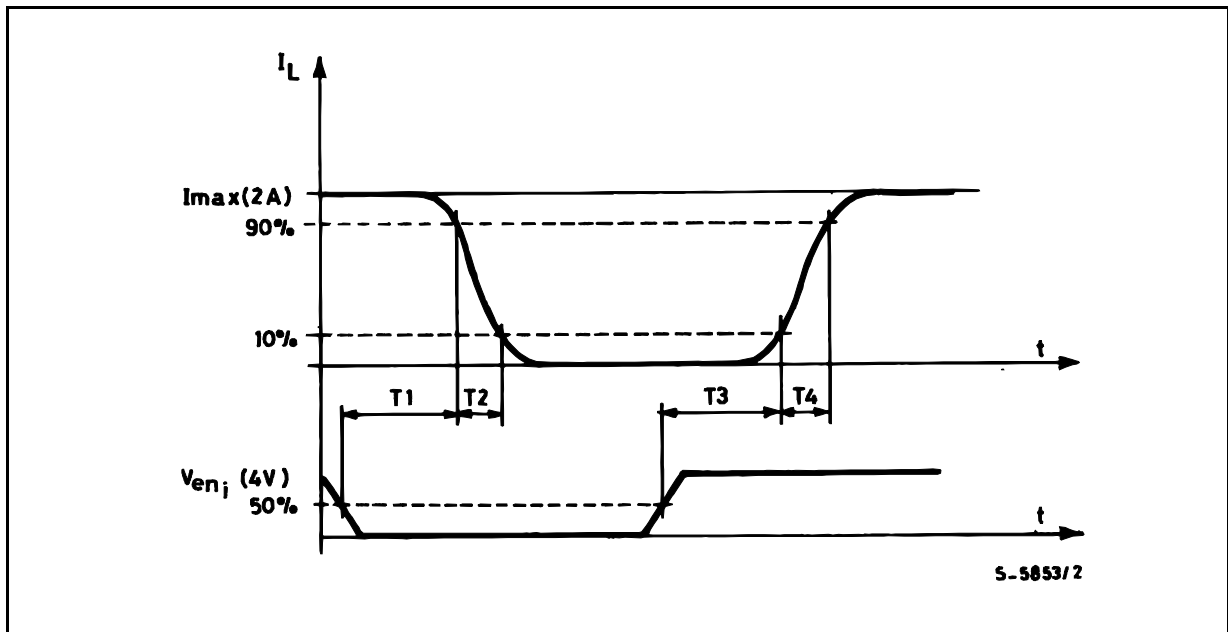


Figure 2 : Switching Times Test Circuits.

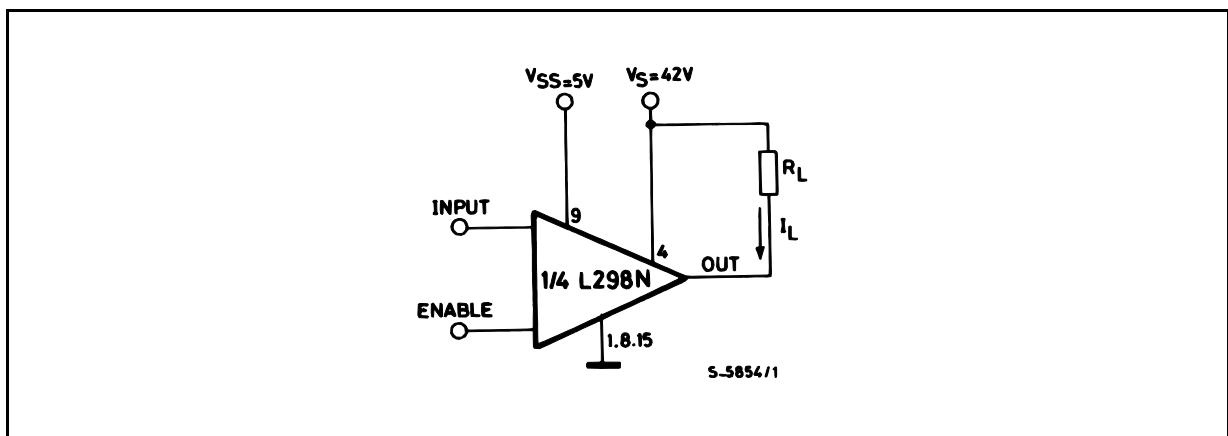


Note : For INPUT Switching, set EN = H  
 For ENABLE Switching, set IN = H

**Figure 3 :** Source Current Delay Times vs. Input or Enable Switching.



**Figure 4 :** Switching Times Test Circuits.



**Note :** For INPUT Switching, set EN = H  
For ENABLE Switching, set IN = L

Figure 5 : Sink Current Delay Times vs. Input 0 V Enable Switching.

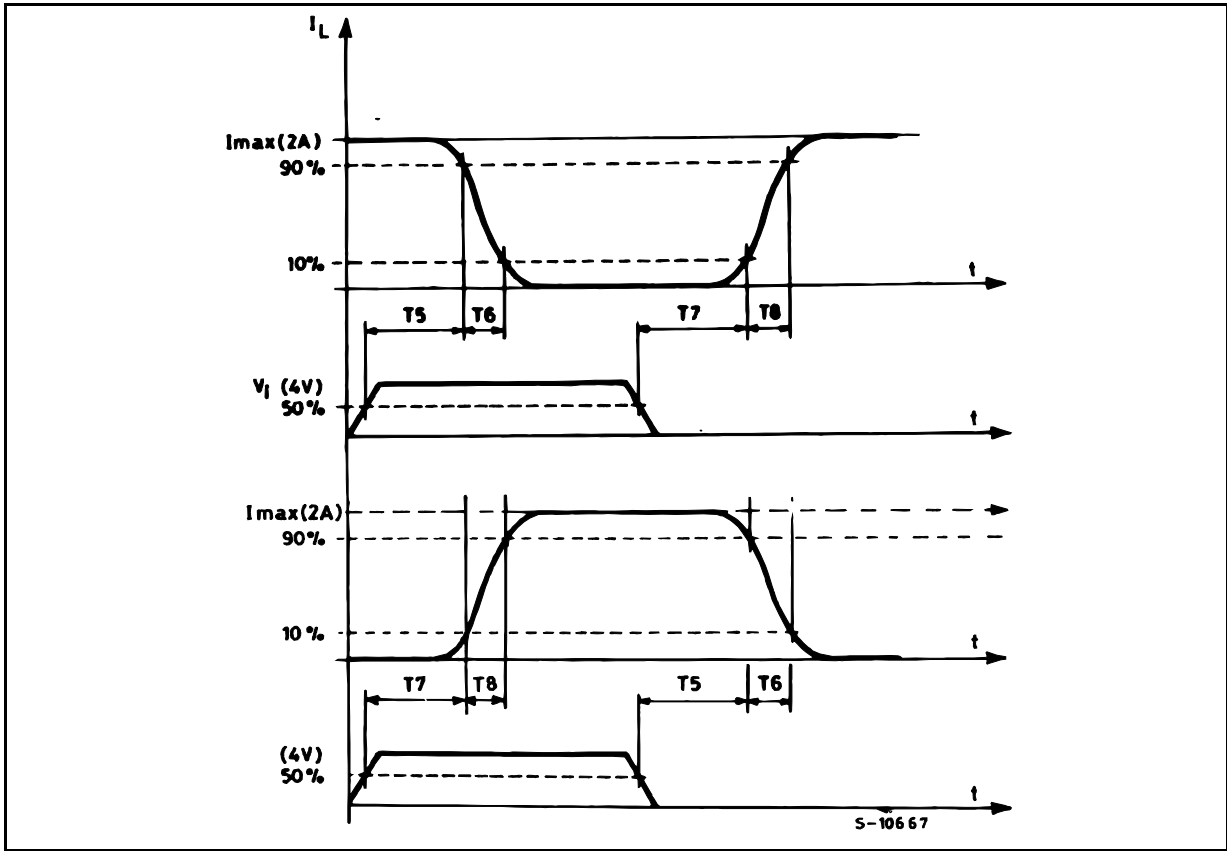
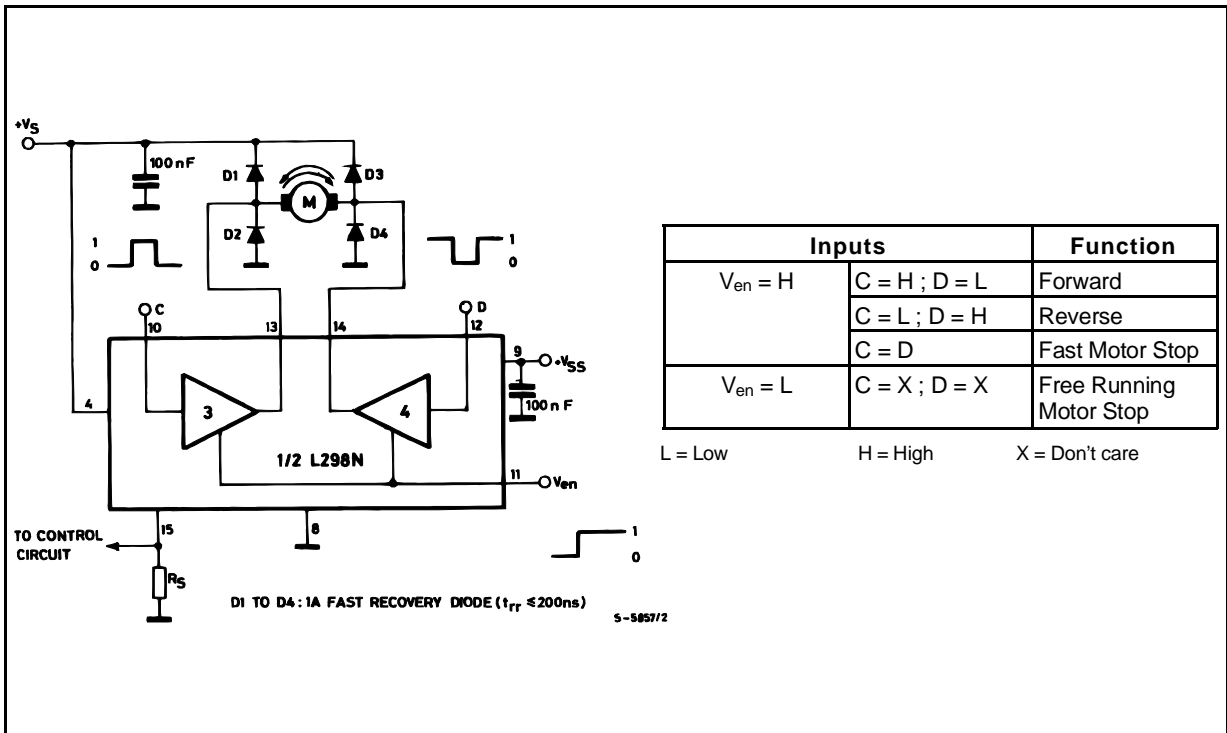
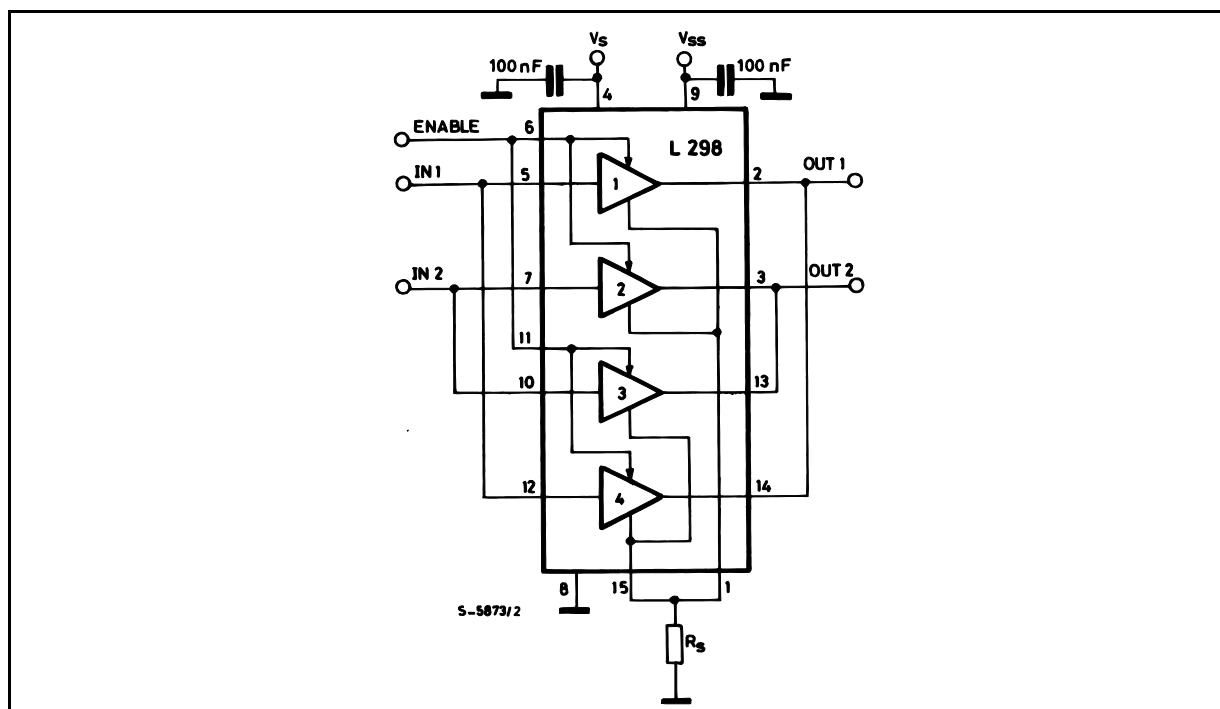


Figure 6 : Bidirectional DC Motor Control.



**Figure 7 :** For higher currents, outputs can be paralleled. Take care to parallel channel 1 with channel 4 and channel 2 with channel 3.



## APPLICATION INFORMATION (Refer to the block diagram)

### 1.1. POWER OUTPUT STAGE

The L298 integrates two power output stages (A ; B). The power output stage is a bridge configuration and its outputs can drive an inductive load in common or differenzial mode, depending on the state of the inputs. The current that flows through the load comes out from the bridge at the sense output : an external resistor ( $R_{SA}$  ;  $R_{SB}$ .) allows to detect the intensity of this current.

### 1.2. INPUT STAGE

Each bridge is driven by means of four gates the input of which are  $In_1$  ;  $In_2$  ;  $EnA$  and  $In_3$  ;  $In_4$  ;  $EnB$ . The  $In$  inputs set the bridge state when The  $En$  input is high ; a low state of the  $En$  input inhibits the bridge. All the inputs are TTL compatible.

## 2. SUGGESTIONS

A non inductive capacitor, usually of 100 nF, must be foreseen between both  $V_s$  and  $V_{ss}$ , to ground, as near as possible to GND pin. When the large capacitor of the power supply is too far from the IC, a second smaller one must be foreseen near the L298.

The sense resistor, not of a wire wound type, must be grounded near the negative pole of  $V_s$  that must be near the GND pin of the I.C.

Each input must be connected to the source of the driving signals by means of a very short path.

Turn-On and Turn-Off : Before to Turn-ON the Supply Voltage and before to Turn it OFF, the Enable input must be driven to the Low state.

## 3. APPLICATIONS

Fig 6 shows a bidirectional DC motor control Schematic Diagram for which only one bridge is needed. The external bridge of diodes  $D1$  to  $D4$  is made by four fast recovery elements ( $trr \leq 200$  nsec) that must be chosen of a  $V_F$  as low as possible at the worst case of the load current.

The sense output voltage can be used to control the current amplitude by chopping the inputs, or to provide overcurrent protection by switching low the enable input.

The brake function (Fast motor stop) requires that the Absolute Maximum Rating of 2 Amps must never be overcome.

When the repetitive peak current needed from the load is higher than 2 Amps, a paralleled configuration can be chosen (See Fig.7).

An external bridge of diodes are required when inductive loads are driven and when the inputs of the IC are chopped ; Shottky diodes would be preferred.

This solution can drive until 3 Amps In DC operation and until 3.5 Amps of a repetitive peak current.

On Fig 8 it is shown the driving of a two phase bipolar stepper motor ; the needed signals to drive the inputs of the L298 are generated, in this example, from the IC L297.

Fig 9 shows an example of P.C.B. designed for the application of Fig 8.

**Figure 8 :** Two Phase Bipolar Stepper Motor Circuit.

This circuit drives bipolar stepper motors with winding currents up to 2 A. The diodes are fast 2 A types.

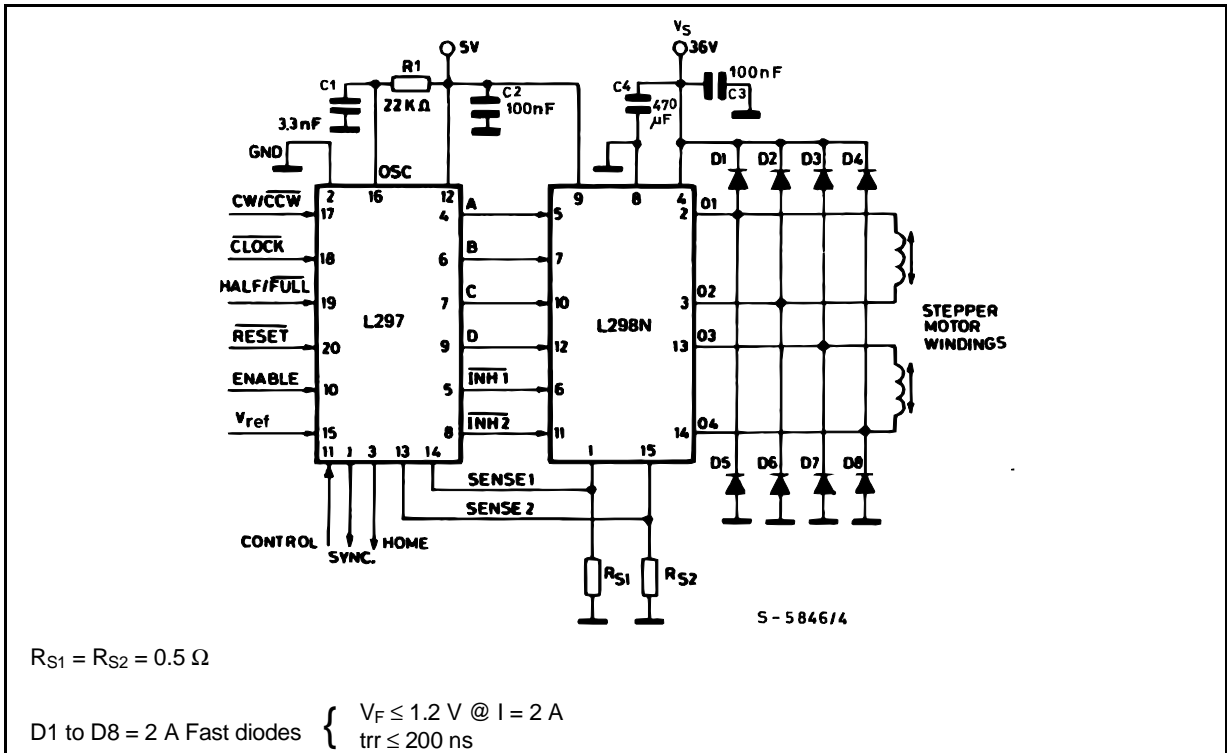


Fig 10 shows a second two phase bipolar stepper motor control circuit where the current is controlled by the I.C. L6506.



Figure 9 : Suggested Printed Circuit Board Layout for the Circuit of fig. 8 (1:1 scale).

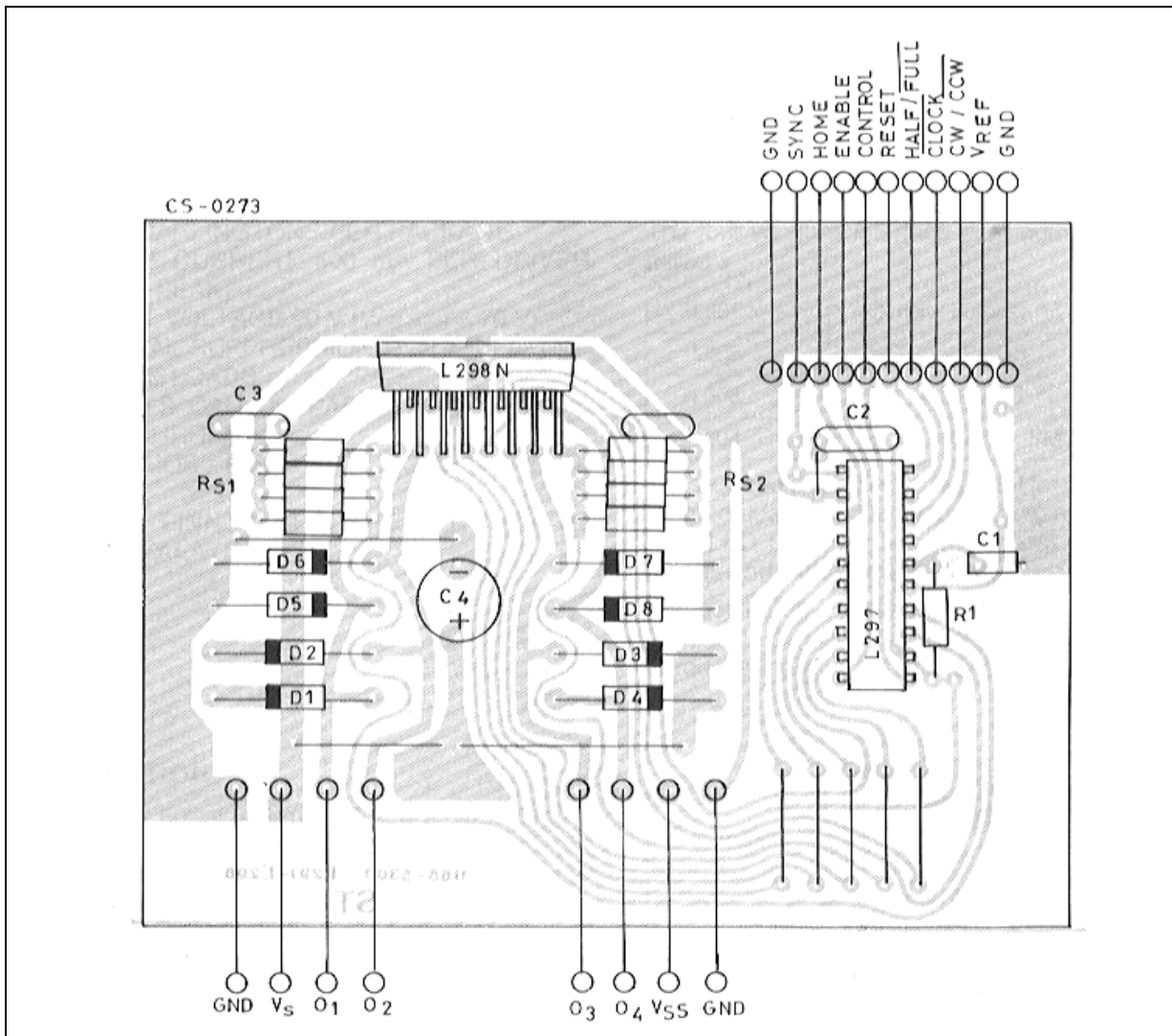
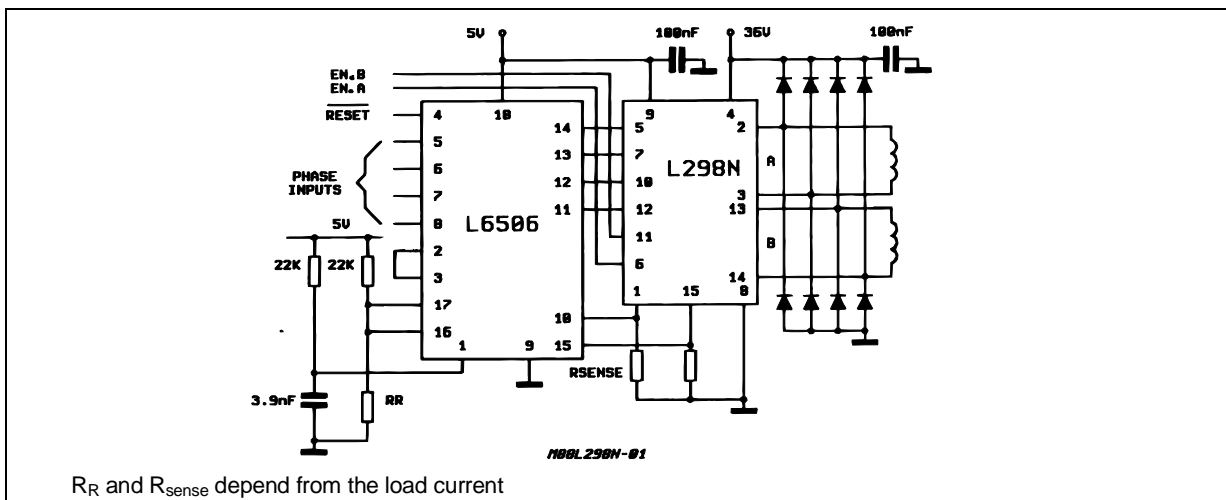
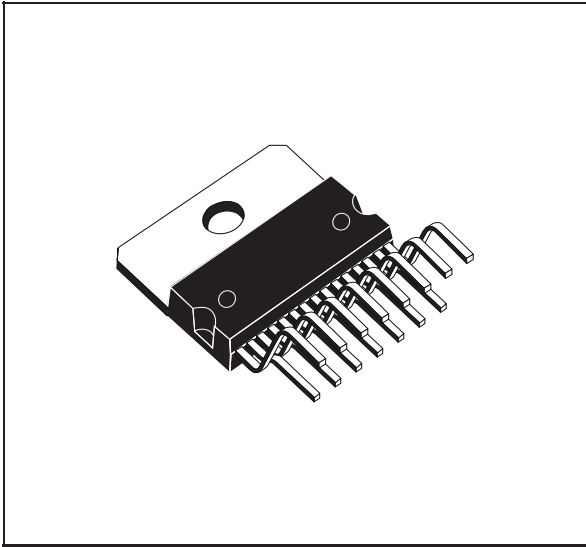


Figure 10 : Two Phase Bipolar Stepper Motor Control Circuit by Using the Current Controller L6506.

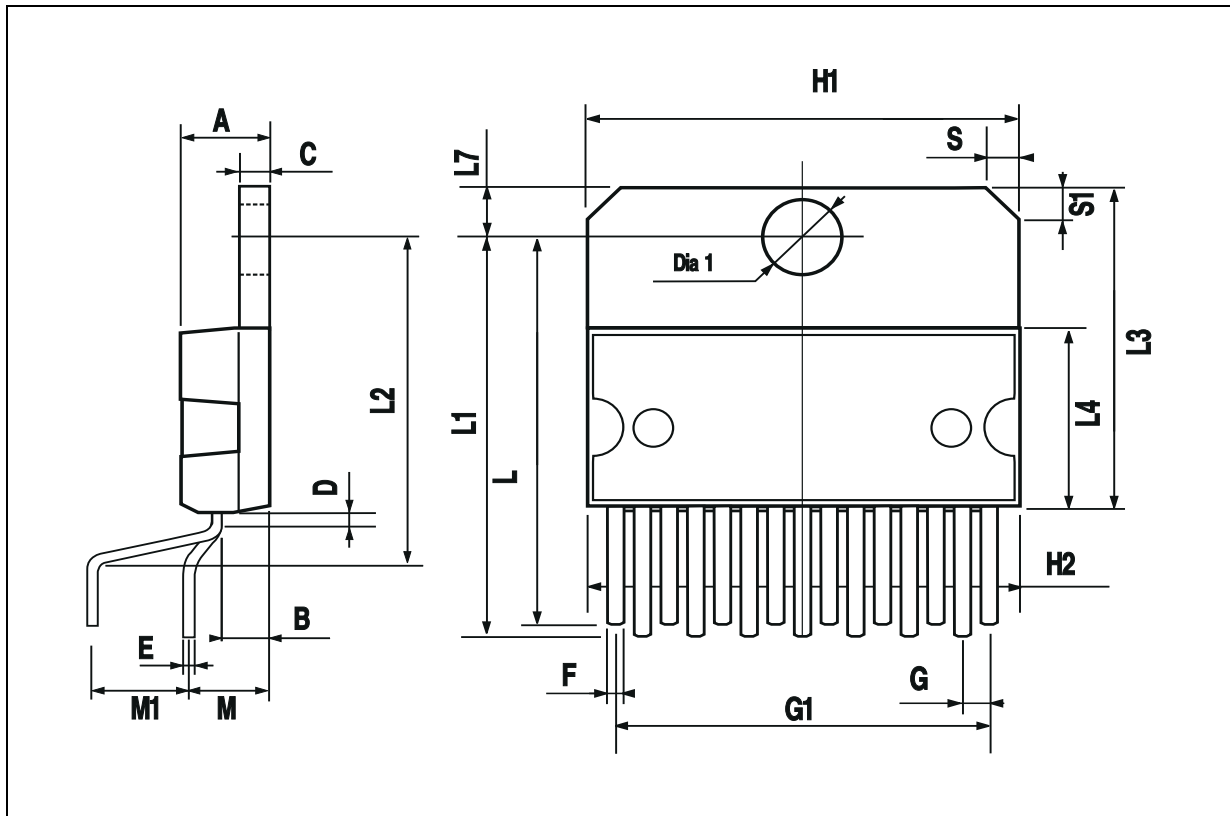


DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			5			0.197
B			2.65			0.104
C			1.6			0.063
D		1			0.039	
E	0.49		0.55	0.019		0.022
F	0.66		0.75	0.026		0.030
G	1.02	1.27	1.52	0.040	0.050	0.060
G1	17.53	17.78	18.03	0.690	0.700	0.710
H1	19.6			0.772		
H2			20.2			0.795
L	21.9	22.2	22.5	0.862	0.874	0.886
L1	21.7	22.1	22.5	0.854	0.870	0.886
L2	17.65		18.1	0.695		0.713
L3	17.25	17.5	17.75	0.679	0.689	0.699
L4	10.3	10.7	10.9	0.406	0.421	0.429
L7	2.65		2.9	0.104		0.114
M	4.25	4.55	4.85	0.167	0.179	0.191
M1	4.63	5.08	5.53	0.182	0.200	0.218
S	1.9		2.6	0.075		0.102
S1	1.9		2.6	0.075		0.102
Dia1	3.65		3.85	0.144		0.152

**OUTLINE AND MECHANICAL DATA**

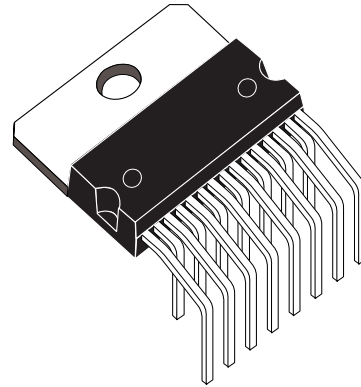


**Multiwatt15 V**

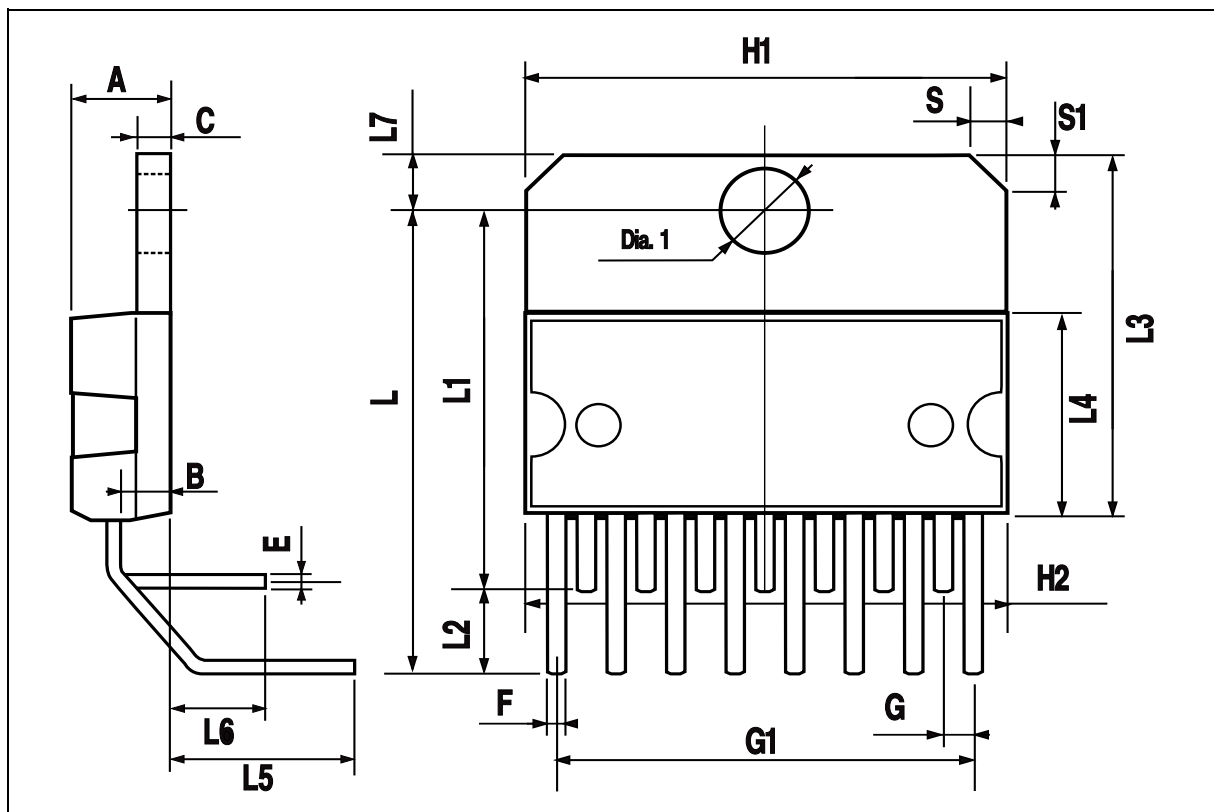


DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			5			0.197
B			2.65			0.104
C			1.6			0.063
E	0.49		0.55	0.019		0.022
F	0.66		0.75	0.026		0.030
G	1.14	1.27	1.4	0.045	0.050	0.055
G1	17.57	17.78	17.91	0.692	0.700	0.705
H1	19.6			0.772		
H2			20.2			0.795
L		20.57			0.810	
L1		18.03			0.710	
L2		2.54			0.100	
L3	17.25	17.5	17.75	0.679	0.689	0.699
L4	10.3	10.7	10.9	0.406	0.421	0.429
L5		5.28			0.208	
L6		2.38			0.094	
L7	2.65		2.9	0.104		0.114
S	1.9		2.6	0.075		0.102
S1	1.9		2.6	0.075		0.102
Dia1	3.65		3.85	0.144		0.152

## OUTLINE AND MECHANICAL DATA



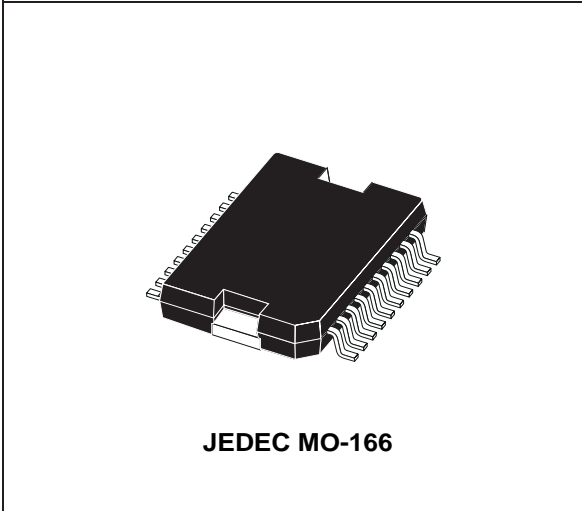
**Multiwatt15 H**



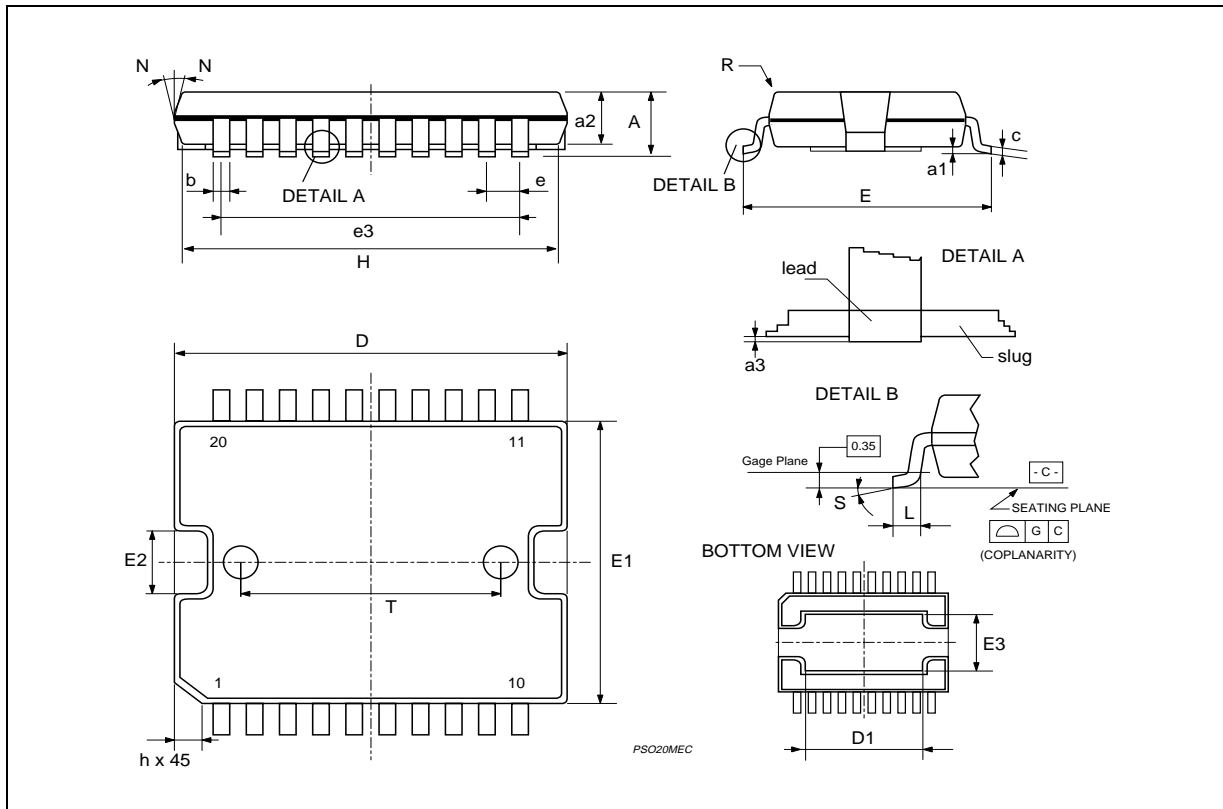
DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			3.6			0.142
a1	0.1		0.3	0.004		0.012
a2			3.3			0.130
a3	0		0.1	0.000		0.004
b	0.4		0.53	0.016		0.021
c	0.23		0.32	0.009		0.013
D (1)	15.8		16	0.622		0.630
D1	9.4		9.8	0.370		0.386
E	13.9		14.5	0.547		0.570
e		1.27			0.050	
e3		11.43			0.450	
E1 (1)	10.9		11.1	0.429		0.437
E2			2.9			0.114
E3	5.8		6.2	0.228		0.244
G	0		0.1	0.000		0.004
H	15.5		15.9	0.610		0.626
h			1.1			0.043
L	0.8		1.1	0.031		0.043
N	10° (max.)					
S	8° (max.)					
T		10			0.394	

(1) "D and F" do not include mold flash or protrusions.  
 - Mold flash or protrusions shall not exceed 0.15 mm (0.006").  
 - Critical dimensions: "E", "G" and "a3"

**OUTLINE AND MECHANICAL DATA**



**PowerSO20**



Information furnished is believed to be accurate and reliable. However, STMicroelectronics assumes no responsibility for the consequences of use of such information nor for any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of STMicroelectronics. Specification mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. STMicroelectronics products are not authorized for use as critical components in life support devices or systems without express written approval of STMicroelectronics.

The ST logo is a registered trademark of STMicroelectronics  
© 2000 STMicroelectronics – Printed in Italy – All Rights Reserved  
STMicroelectronics GROUP OF COMPANIES

Australia - Brazil - China - Finland - France - Germany - Hong Kong - India - Italy - Japan - Malaysia - Malta - Morocco -  
Singapore - Spain - Sweden - Switzerland - United Kingdom - U.S.A.

<http://www.st.com>

# HIGH CURRENT 30A DC MOTOR DRIVER



This driver uses two high current half bridge Infineon BTS 7960 chip for motor drive applications. Interfacing to a microcontroller or Arduino is made easy using this driver which features current sensing, slew rate adjustment and protection against over-temperature, overvoltage, under-voltage, Over-current and short circuit.

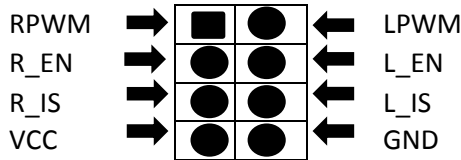
This small size driver provides a cost optimized solution for protected high current PWM motor drives.

## Feature

- Operating Voltage 5 to 27V (B+)
- Control motor speed by PWM up to 25 kHz.
- Motor forward and backward motion control
- Switched mode current limitation for reduced power dissipation.
- Current limitation level of 30 A Current sense capability
- Over-temperature shut down Over-voltage lock out.
- Large size heat sink is mounted to driver.

- Size: 4\*5\*1.2 cm.
- Weight: 66 gm.

## How to connect to Microcontroller (Arduino)



1	RPWM	Forward level or PWM signal input, active HIGH
2	LPWM	Reverse level or PWM signal input, active HIGH
3	R_EN	Forward drive enable input, HIGH enable, LOW Close
4	L_EN	Reverse drive enable input, HIGH enable, LOW Close
5	R_IS	Forward drive – side current alarm output
6	L_IS	Reverse drive – side current alarm output
7	VCC	5V power input from Arduino board.
8	GND	Ground.

- 5V power supply from Arduino , GND connected Arduino GND
- R\_EN and L\_EN shorted and connected to 5V level, the driver to work.
- L\_PWM, input PWM signal or high motor forward.
- R\_PWM, input PWM signal or high motor revers.

## Arduino Code

```
/*
IBT-2 Motor Control Board driven by Arduino.
```

Speed and direction are separated in two functions.

```
Connection to the IBT-2 board:
IBT-2 pin 1 (RPWM) to Arduino pin 5(PWM)
IBT-2 pin 2 (LPWM) to Arduino pin 6(PWM)
IBT-2 pins 3 (R_EN), 4 (L_EN), 7 (VCC) to Arduino 5V pin
IBT-2 pin 8 (GND) to Arduino GND
IBT-2 pins 5 (R_IS) and 6 (L_IS) not connected
*/
```

```
int RPWM_Output = 5; // Arduino PWM output pin 5; connect to IBT-2 pin 1 (RPWM)
int LPWM_Output = 6; // Arduino PWM output pin 6; connect to IBT-2 pin 2 (LPWM)
int reversePWM;
int forwardPWM;

void setup()
{
  pinMode(RPWM_Output, OUTPUT);
  pinMode(LPWM_Output, OUTPUT);
}

void loop()
{
```

```
// reverse rotation
for ( reversePWM=0;reversePWM<255;reversePWM++)
{
    analogWrite(LPWM_Output, 0);
    analogWrite(RPWM_Output, reversePWM);
}
delay(200);
// forward rotation
for ( forwardPWM=0;forwardPWM<255;forwardPWM++)
{
    analogWrite(LPWM_Output, forwardPWM);
    analogWrite(RPWM_Output, 0);
}
delay(200);
}
```

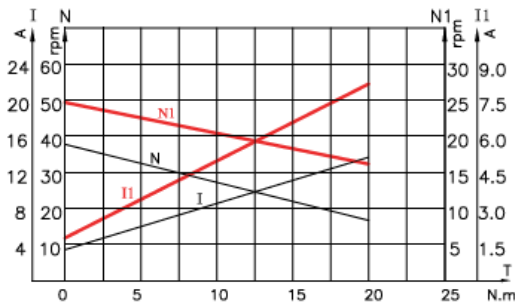
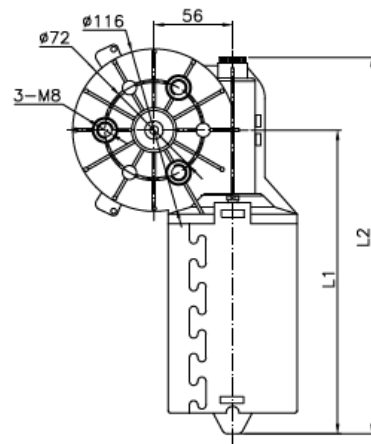
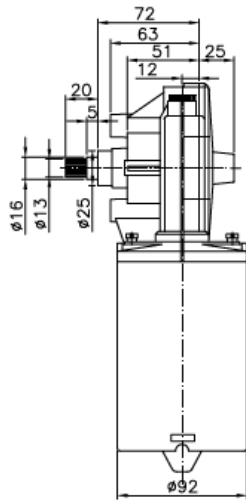


# Wiper Motor –WD21100/21100-B



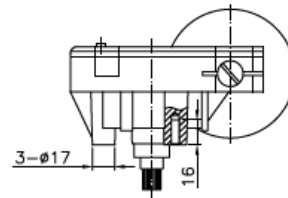
- 12VDC and 24VDC Available
- Aluminum Diecasting Gearbox
- Rolled Steel Housing Construction
- Dual Speed Design
- Dynamically Balanced Rotor

## Mechanical



WD11100

---BLK High Speed, ---RED Low Speed



## Performance Data at Rated 12V DC

Type	No-Load Speed	Rated			Peak Torque N.m	L1 mm	L2 mm
		Speed rpm	Current A	Torque N.m			
WD21100	38	25	10	12.5	20	223.5	276.5
	25	19	5.7	13	20		
WD21100-B	37	25	9.7	12	20	235	288
	23	15	7.8	16	20		