Motion Control Systems Development

for Bearcat II

Thesis Advisor and Committee Chair: Dr. Ernest Hall

Created by: Ming Cao
Department of Mech. Ind. and Nuc. Eng.
University of Cincinnati
Cincinnati, OH 45219

Contents

Abstract

Motion control is one of the technological foundations of mobile robot. As part of a mobile robot project, this paper presents the development of motion control systems for a mobile robot. The paper began with the outline of the design purposes and goals to be achieved, followed by the examples of motion control designs. The mobile robot motion control systems was modeled, components were selected and constructed. System PID compensation theory and its achievement was presented. The motion control strategy was selected, mobile robot behaviors was identified and programmed. In the last two chapters, the mobile robot and its performance in the AUVS competition was described and observation of the motion system performance during the real life contest was stated. A mechanical design problem was concluded and possible solution was described. Also, velocity feedback as a possible improvements was point out for the future research.

Acknowledgements

After two years of researches at the Robotics Research Center at the University of Cincinnati, I have a chance to show my friends that my learning has real life meaning instead of brain storming only. Thanks for his true mentor, Dr. Ernest L. Hall, who guided me through my graduate education at the University of Cincinnati, who also led me into this fantastic robotics world and gave me endless encouragement.

I acknowledge my friends at the UC robotics team for their valuable suggestions and critiques. My sincere thanks to Karthik, Jin and Satish and Sameer for their cooperation, without which no motion system could be constructed.

My achievement also belongs to my parents, my brother and my sister in China, who gave me deepest love in the world.

Chapter 1 INTRODUCTION

Motion control is one of the technological foundations of industrial automation. Whether the motion of a product, the path of a cutting tool, the motion of an industrial robot arm conducting seam welding, the motion of a parcel being moved from a storage bin to a loading dock by a shipping cart, or another application, the control of motion is a fundamental concern. Putting an object in the correct place with the right amount of force and torque at the right time is essential for efficient manufacturing operations. To be able to control a motion process, the precise position of objects needs to be measurable. Feedback comparison of the target and actual positions is then a natural step in implementing a motion control system. This comparison generates an error signal that may be used to correct the system, thus yielding repeatable and accurate results.

However, the use of feedback can lead to an unstable system whose output may oscillate or even go to infinity with a small input signal. Stability determination is therefore an important design consideration. One specification for absolute stability requires that the poles of the transfer function must be in the left half of the s-plane. Absolute stability, often specified in the frequency domain, is essential and necessary but not sufficient. Frequency domain specifications relating to relative system stability may also be given. For relative stability, a certain phase margin and gain margin may be specified to ensure that the system will remain stable although some parameters change due to temperature changes, aging or other environmental changes. If a system is stable, then other performance criteria, specified in either the time or frequency domain, may be considered to meet the performance requirements. Short-term, or transient, response specifications such as rise-time or percent overshoot to a unit step function input, may be given. Long term, or steady state, response such as zero steady state error may also be specified.

1.1 Objectives

The Objective of this research was to apply motion control principles into a mobile robot systems design, which is going on in the University of Cincinnati Robotics Center. The mobile robot was constructed during the 1998-1999 academic year, called BEARCAT II. Its design has inherited many features of its predecessor, BEARCAT I, such as vision guidance, sonar detection and digital control. In addition, BEARCAT II achieved many innovative motion activities, featured as rotating sonar, ZTR (Zero Turning Radius), current control loop, multi-level controller design (conventional control and fuzzy logic control). This paper focuses on the vehicle steering control and sonar rotating systems development and programming. In the first part, an introduction to the motion control theory was introduced, followed by examples from simple to complex. Then, a description of the mobile robot design facilities was presented, analyzed and the design process was detailed and realized. The final results were tested and proofed. The last part draws a conclusion with future research.

1.2 Motion control strategies:

Automated Guided Vehicle (AGV) is an intelligent machine that has ‘intelligence’ to determine its motion status according to the environment conditions. For an AGV to operate it must sense its environment, be able to plan its operations and then react based on this plan. This type of system must be placed in a known space from which it can determine its orientation via the use of markers (e.g. white lines, walls, obstacles, etc.). The model of the AGV itself may be easy to establish, however, the environmental model is difficult to obtain.

The running environment could be varied, such as the path orientation, road flatness, obstacle position, road surface friction, etc. There are a great many uncertainties of what conditions will emerge during its operation. Thus a new control method other than the conventional control method is demanded to manage the response of the whole system.

In the last years, fuzzy logic has been applied mobile robot and autonomous vehicle control significantly. The best arguments supporting fuzzy control are the ability to cope with imprecise information in heuristic rule based knowledge and sensor measurements.

Fuzzy logic can help design robust individual behavior units. Fuzzy logic controllers incorporate heuristic control knowledge. It is a convenient choice when a precise linear model of the system to be controlled cannot be easily found. Another advantage of fuzzy logic control is to use fuzzy logic for representing uncertainties, such as vagueness or imprecision which can not be solved by probability theory. Also, fuzzy logic offers greater flexibility to users, among which, we can chose the one that best fits the type of combination to be performed.

The University of Cincinnati Robotics Research Center has been working on improving the ability of the automated guided vehicles for several years. Based on previous experiences, a new AGV ---- Bearcat II is being built. This new robot features more agility, smaller size, digital control, higher reliability, ZTR motion, higher level intelligence, etc. Bearcat-II is a computer controlled mobile robot. It is designed to navigate in a changing environment as shown in Figure 1.1 Generally, the road for the robot to travel is marked with white lines. The road direction will be changing while the width of the road remains stable. There will be obstacles blocking the road. It is expected that the mobile robot can pass through these obstacles without knocking them down. In short, the robot should be able to follow the changing path and avoid obstacles. In order to do this, the robot should have the ability to sense and deal with the issue of uncertainty and incomplete information about the environment in a reasonable time.

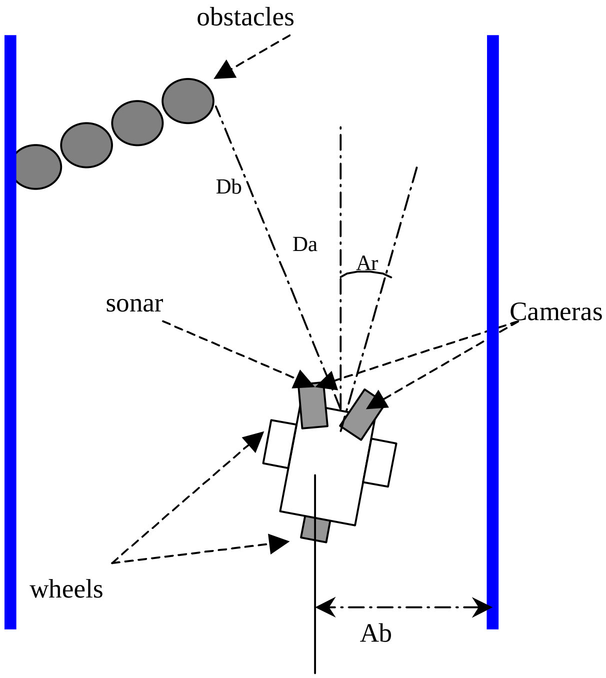


Figure 1.1 Bearcat II and its working environment

This research offers a design experience that is at the very cutting edge of engineering education. It encompasses the very latest technologies impacting industrial development and taps subjects of high interest to researchers. The technologies involved come from a wide range of disciplines and are those of great current interest in both industry and engineering education. Team organization and leadership are practiced. Students solicit and interact with industrial sponsors who provide component hardware and advice, and in that way get an inside view of industrial design and opportunities for employment. The technologies involved are those of emerging and burgeoning industries today, such as Detectors, Voltage Regulation, Sensors, PWM Amplifiers, Ultrasonics (SONARS), fuzzy logic, Computer Programming, Artificial Intelligence, Data Fusion, etc.

Among those applications are many with great opportunities for breakthroughs and innovation, and employment opportunities for knowledgeable young engineers.

1.3 **Design Objectives:**

The desire for a super intelligent robot vehicle that can beat all other teams during the contest forces us to build a successful control mechanism that can drive the mobile robot along a changing contest course, such as steep ramp, sand, wet ground. This requires the robot has the ability to "see" its course. Another consideration of this design is to let the robot sense the obstacles in front of it. This indicates the design of separate systems that may generate different control signals, which may be integrated with certain method to drive the conventional motion system. For the motion control system, some specific requirements can be listed as follows:

1. It can automatically navigate around an outdoor course within a pre-defined time, and at the same time, avoid obstacles on the track.
2. Vehicles must be unmanned and autonomous. They must compete based on their ability to perceive the course environment and avoid obstacles. All computational power, sensing and control equipment must be carried on board the vehicle.
3. There are at least one manual E-Stop and at least one remote E-stop device built into the entire system.

1.4 Specific goals:

Our approaches to achieve the above goals are based upon our previous design experiences and the devices available. The specific aim of the proposed research can be stated as follows:

1. Define the steering mechanism of the robot, which can meet the specific challenges of the AUVS contest.
2. Select and construct equipment that the desired functional and operational features and also the proper ability to drive the vehicle to the desired direction at the desired speed.
3. Program to achieve the motion behaviors under the control of central computer.

1.4 Specific goals:

Our approaches to achieve the above goals are based upon our previous design experiences and the devices available. The specific aim of the proposed research can be stated as follows:

1. Define the steering mechanism of the robot, which can meet the specific challenges of the AUVS contest.
2. Select and construct equipment that the desired functional and operational features and also the proper ability to drive the vehicle to the desired direction at the desired speed.
3. Program to achieve the motion behaviors under the control of central computer.

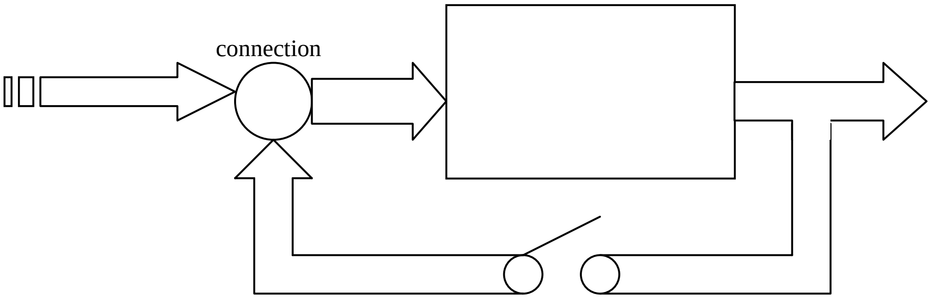
Chapter 2. Motion Control Review

Motion control systems may operate in an open loop, closed loop non-servo, closed loop servo, or a hybrid combination design, as shown in Figure 2.1. The open loop approach, shown in Figure 2.1 (a), has input and output but no measurement of the output for comparison with the desired response. A non-servo, on-off, or bang-bang control approach is shown in Figure 2.1 (b). In this system, the input signal turns the system on, and when the output reaches a certain level, it closes a switch that turns the system off. A proportion, or servo, control approach is shown in Figure 2.1 (c). In this case, a measurement is made of the actual output signal, which is fed back and compared to the desired response. The closed loop servo control system will be studied in this paper.

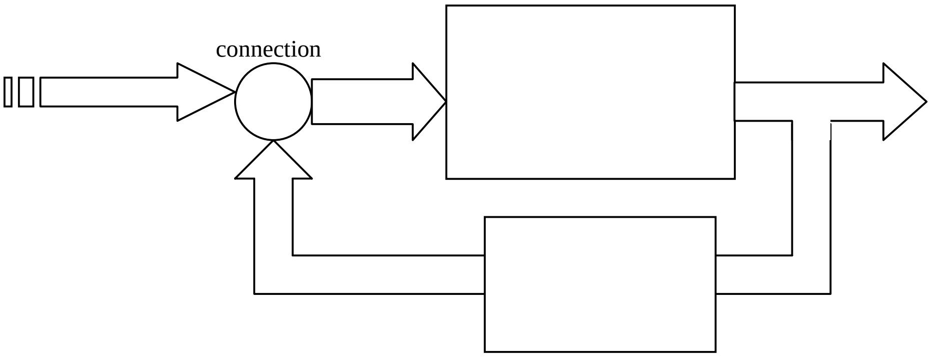
The components of a typical servo controlled motion control system may include an operator interface, motion control computer, control compensator, electronic drive amplifiers, actuator, sensors and transducers, and the necessary interconnections. The actuators may be powered by electro-mechanical, hydraulic or pneumatic or a combination of these power sources.

The operator interface may include a combination of switches, indicators and displays, including a computer keyboard and a monitor or display. The motion control computer generates command signals from a stored program for a real time operation. The control compensator is a special program in the motion control computer. Selecting the compensator parameters is often a critical element in the success of the overall system. The drive amplifiers and electronics must convert the low power signals from the computer to the higher power signals required to drive the actuators. The sensors and transducers record the measurement of position or velocity that are used for feedback to the controller. The actuators are the main drive devices that supply the force or torque required to move the load. All of these subsystems must be properly interconnected in order to function properly.

(a) Open loop system



(b) Non-servo system



(c) Servo system

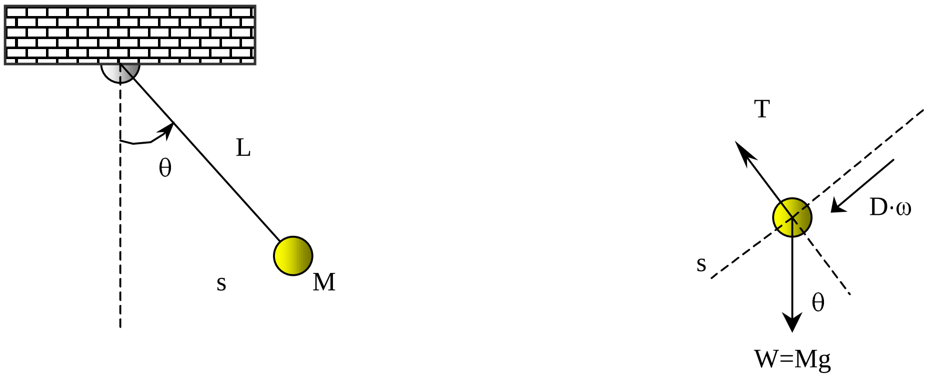
1. Motion Control Example

Consider the simple pendulum shown in Figure 2.2 that has been studied for more than 2000 years. Aristotle first observed that a bob swinging on a string would come to rest, seeking a lower state of energy. Later, Galileo Galilee made a number of incredible, intuitive inferences from observing the pendulum. Galileo's conclusions are even more impressive considering that he made his discoveries before the invention of calculus.

2.1 Flexible link pendulum:

The pendulum may be described as a bob with mass, M, and weight given by W=Mg, where g is the acceleration of gravity, attached to the end of a flexible cord of length, L as shown in Figure 2.2. When the bob is displaced by an angle θ, the vertical weight component causes a restoring force to act on it. Assuming that viscous damping, from resistance in the medium, with a damping factor, D, causes a retarding force proportional to its angular velocity, ω, equal to D·ω. Since this is a homogeneous, unforced system, the starting motion is set by the initial conditions. Let the angle at time θ(t=0) be 45 °. For definiteness let the weight, W = 40 lbs., the length, L = 3 ft, D = 0.1 (lb.s) and g=32.2 (ft/s^2).

The analysis is begun by drawing a free body diagram of the forces acting on the mass. We will use the tangent and normal components to describe the forces acting on the mass. The free body diagram shown in Figure 2.2 (b) and Newton's second law are then used to derive a differential equation describing the dynamic response of the system. Forces may be balanced



(a) Physical diagram *(b) Free body diagram*

Figure 2.2 Pendulum as studied by Galileo Galilee

in any direction, however, a particularly simple form of the equation for pendulum motion can be developed by balancing the forces in the tangential direction:

$$\sum F_t = Ma_t$$ ~~(1)~~

This gives the following equation:

$$-Mg\sin\theta - D\frac{d\theta}{dt} = Ma_t$$ ~~(2)~~2.

The tangential acceleration is given in terms of the rate of change of velocity or arc length by the equation:

$$a_t = \frac{dv}{dt} = \frac{d^2 s}{dt^2}$$ ~~(3)~~2.

Since the arc length, *s*, is given by:

$$s = L\theta$$ ~~(4)~~2.

Substituting *s* into the differential in Equation 2.3 yields:

$$a_t = L\frac{d^2\theta}{dt^2}$$ ~~(5)~~2.

Thus, combining Equation 2.2 and Equation 2.5 yields:

$$-Mg\sin\theta - D\frac{d\theta}{dt} = Ma_t = ML\frac{d^2\theta}{dt^2}$$ ~~(6)~~2.

Note that the unit of each term is force. In English units, W is in lb_f, g is in $ft./sec^2$, D is in lb.sec, L is in feet, θ is in radians, dθ/dt is in rad/sec and $d^2\theta/dt^2$ is in rad/sec^2. In international SI units, M is in kg, g is in m/sec^2, D is in kg-m/sec, L is in meters, θ is in radians, dθ/dt is in rad/sec and $d^2\theta/dt^2$ is in rad/sec^2.

This may be rewritten as:

$$\frac{d^2\theta}{dt^2} + \frac{D}{ML}\frac{d\theta}{dt} + \frac{g}{L}\sin\theta = 0$$ ~~(7)~~2.

This equation may be said to describe a *system*. While there are many types of systems, systems with no output are difficult to observe, and systems with no input are difficult to control. To emphasize the importance of position, we can describe a *kinematic* system, such as y = T(x). To emphasize time, we can describe a *dynamic* system, such as g = h(f(t)). Equation 7 describes a dynamic response. The differential equation is non-linear because of the sin θ term.

For a linear system, y = T (x), two conditions must be satisfied:

1. If a constant, a, is multiplied by the input, x, such that ax is applied as the input, then the output must be multiplied by the same constant, as in Equation 8:

$$T(ax) = ay$$

~~(8)~~2.

2. If the sum of two inputs is applied, the output must be the sum of the individual outputs and the principal of superposition must hold as demonstrated by the following equations:

$$T(x_1 + x_2) = y_1 + y_2$$

~~(9)~~2.

where

$$T(x_1) = y_1$$

~~(10)~~

and

$$T(x_2) = y_2$$

~~(11)~~

Equation 7 is non-linear because the sine of the sum of two angles is not equal to the sum of the sines of the two angles. For example, sin(45°) = 0.707 while sin(90°) = 1.

Invariance is an important concept for systems. In an optical system, such as reading glasses, position invariance is desired, whereas, for a dynamic system time invariance is very important.

Since an arbitrary input function, f(t) may be expressed as a weighted sum of impulse functions using the Dirac delta function, δ(t-τ). This sum can be expressed as:

$$f(t) = \int_{-\infty}^{\infty} f(\tau)\delta(t-\tau)d\tau$$

(Note that t is the time the output is observed and τ is the time the input is applied.)

The response of the linear system to this arbitrary input may be computed by:

$$g(t) = h\left[\int_{-\infty}^{\infty} f(\tau)\delta(t-\tau)d\tau\right]$$

Thus by the property of linearity we obtain:

$$g(t) = \int_{-\infty}^{\infty} f(\tau)h[\delta(t-\tau)]d\tau$$

Therefore, the response of the linear system is characterized by the response to an impulse function. This leads to the definition of the impulse response, h(t,τ), as:

$$h(t,\tau) = h\left[\delta(t-\tau)\right]$$

Since the system response may vary with the time the input is applied, the general computational form for the output of a linear system is the superposition integral called the Fredholm integral equation [2-3]:

$$g(t) = \int_{\alpha}^{\beta} f(\tau)h(t,\tau)d\tau$$

$$g(t) = \int_{-\infty}^{+\infty} f(\tau)h(t,\tau)d\tau$$

~~(17)~~2.

The limits of integration are important in determining the form of the computation. Without any assumptions about the input or system, the computation must extend over an infinite interval.

An important condition of realizability for a continuous system is that the response be non-anticipatory, or casual, such that no output is produced before an input is applied:

$$h(t,\tau) = 0 \qquad for \quad t - \tau < 0$$

~~(18)~~

The causality condition leads to the computation:

$$g(t) = \int_{-\infty}^{t} f(\tau)h(t,\tau)d\tau$$

~~(19)~~

With the condition that f(t)=0 for t<0, the computation reduces to:

$$g(t) = \int_0^t f(\tau)h(t,\tau)d\tau$$

If the system is time invariant, then:

$$h(t,\tau) = h(t-\tau)$$

This leads to the familiar convolution equation:

$$g(t) = \int_0^t f(\tau)h(t-\tau)d\tau$$

The reason that linear systems are so important is that they are widely applicable and that a systematic method of solution has been developed for them. The relationship between the input and output of a linear, time invariant system is known to be a convolution relation. Furthermore, transformational techniques, such as the Laplace transforms, can be used to convert the convolution into an equivalent product in the transform domain. The Laplace transform F(s) of f(t) is:

$$F(s) = \int_0^\infty f(t)e^{-st}dt$$

The convolution theorem states that:

$$G(s) = H(s)F(s)$$

Where

$$G(s)=\int_{0}^{\infty} g(t)e^{-st}dt$$

And

$$H(s)=\int_{0}^{\infty} h(t)e^{-st}dt$$

(Note that this theorem shows how to compute the convolution with only multiplication and transform operations.) The transform, H(s), of the system function, h(t), is called the system transfer function. For any input, f(t), its transform, F(s), can be computed. Then multiplying by H(s) yields the transform G(s). The inverse Laplace transform of G(s) gives the output time response, g(t).

This transform relationship may also be used to develop block diagram representations and algebra for linear systems, which is very useful to simplify the study of complicated systems.

2.1.1 Linear approach modeling

Returning to the pendulum example, the solution to this non-linear equation with D=0 involves the elliptic function. (The solutions of this non-linear system will be investigated later using Simulink.) Using the approximation sin θ = θ in Equation 7 gives the linear approximation:

$$\frac{d^2\theta}{dt^2}+\frac{D}{ML}\frac{d\theta}{dt}+\frac{g}{L}\theta=0$$

When D=0, Equation 27 simplifies to the linear differential equation for simple harmonic motion:

$$\frac{d^2\theta}{dt^2}+\frac{g}{L}\theta=0$$

A Matlab[1] m-file may be used to determine the time response to the linear differential equation. To use Laplace transforms in Matlab, we must use the linear form of the system and provide initial conditions since no forcing function is applied.

Remembering that the Laplace transform of the derivative is:

$$L\left\{\frac{d\theta}{dt}\right\}=s\Theta(s)-\theta(0^-)$$

2.~~(~~29

and

$$L\left\{\frac{d^2\theta}{dt^2}\right\}=s^2\Theta(s)-s\theta(0^-)-\frac{d\theta(0^-)}{dt}$$

2.~~(~~30

Taking the Laplace transform of the linear differential Equation 27 gives:

$$s^2\Theta(s)-s\theta(0^-)-\frac{d\theta(0^-)}{dt}+\frac{D}{ML}\left[s\Theta(s)-\theta(0^-)\right]+\frac{g}{L}\Theta(s)=0$$

2.~~(~~31)

This may be simplified to:

$$\Theta(s)=\frac{s\theta(0^-)-\dfrac{D}{ML}\theta(0^-)+\dfrac{d\theta(0^-)}{dt}}{s^2+\dfrac{D}{ML}s+\dfrac{g}{L}}$$

2.~~(~~32

(Note that the initial conditions act as a forcing function for the system to start it moving.) It is more common to apply a step function to start a system. The unit step function is defined as:

$$u(t) = \begin{cases} 1 & for \quad t >= 0 \\ 0 & for \quad t < 0 \end{cases}$$ (2.33

(Note that the unit step function is the integral of the delta function.) It may also be shown that the Laplace transform of the delta function is 1, and that the Laplace transform of the unit step function is 1/s.

To use Matlab to solve the transfer function for θ(t), we must tell Matlab that this is the output of some system. Since G(s) = H(s)F(s), we can let H(s) = 1 and F(s) = Θ(s). Then the output will be G(s) = Θ(s), and the impulse function can be used directly. If Matlab does not have an impulse response but it does have a step response, then a slight manipulation is required. (Note that the impulse response of system G(s) is the same as the step response of system s (G(s)).)

The transform function with numerical values substituted is:

$$\Theta(s) = \frac{45(s - 0.0268)}{s^2 + 0.0268s + 10.73}$$ (2.34

Note that θ (0) = 45°, and dθ(0)/dt = 0. We can define T0 = θ (0) for ease of typing, and express the numerator and denominator polynomials by their coefficients as shown by the num and den vectors below.

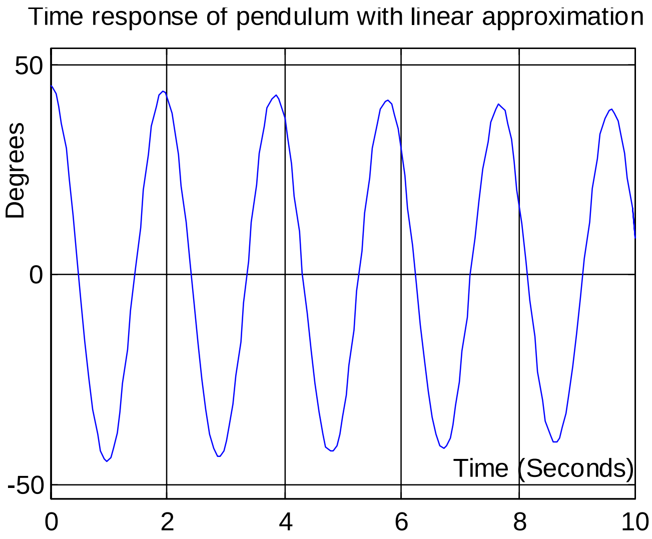
To develop a Matlab m-file script using the step function, define the parameters from the problem statement:

```
T0 =45

D = 0.1

M = 40/32.2

L = 3

G=32.3

num = [T0, D*T0/(M*L), 0];

den = [1, D/(M*L), G/L ];

t= 0:0.1:10;

step(num,den,t);

grid on

title('Time response of the pendulum linear approximation')
```

This m-file or script may be run using Matlab and should produce an oscillatory output. The angle starts at 45 degrees at time 0 and goes in the negative direction first, then oscillates to some positive angle and dampens out. The period, T in seconds (or frequency, f = 1/T in

cycles/second or Hertz) of the response can be compared to the theoretical solution for an undamped pendulum given in Equation 2.35 [4]. The response is shown in Figure 2.3

$$T = 2\pi\sqrt{\frac{L}{g}}$$



2.1.2 Non-linear approach modeling

To solve the non-linear system, we can use Simulink to develop a graphical model of the system and plot the time response. This requires developing a block diagram solution for the differential equation and then constructing the system using the graphical building blocks of Simulink. From this block diagram, a simulation can be run to determine a solution.

To develop a block diagram, write the differential equation in the following form:

$$\frac{d^2\theta(t)}{dt^2} = \frac{-D}{ML}\frac{d\theta}{dt} - \frac{g}{L}\sin\theta(t)$$

Note that this can be drawn as a summing junction with two inputs and one output. Then note that θ can be derived from $d^2\theta/dt^2$ by integrating twice. The output of the first integrator gives dθ/dt. An initial velocity condition could be put at this integration. A pick-off point could also be put here to be used for velocity feedback. The output of the second integrator gives θ. The initial position condition can be applied here. This output position may also be fed back for the position feedback term. The constants can be implemented using gain terms on amplifiers since an amplifier multiplies its input by a gain term. The sine function can be represented using a non-linear function.

The motion is started by the initial condition, θ(0+) = 45°, which was entered as the integration constant on the integrator which changes dθ/dt to θ. Note that the sin function expects an angle in radians, not degrees. Therefore, the angle must be converted before computing the sine. In addition, the output of the sine function must be converted back to degrees. A block diagram of this non-linear model is shown in Figure 2.4. The mathematical model to analyze such a non-linear system is complicated. However, a solution is easily obtained with the sophisticated software of Simulink. The response of this non-linear system is shown in Figure 2.5. Note that it is very similar to the response of the linear system with an amplitude swinging between + 45° and – 45°, and a period slightly less than 2 seconds, indicating that the linear system approximation is not bad. Upon close inspection, one would see that the frequency of the non-linear solution is not, in fact, constant.

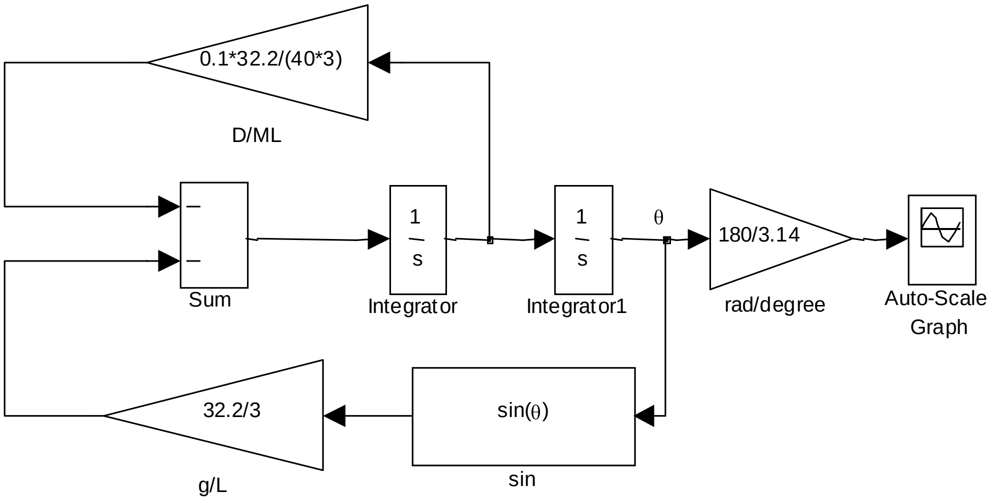
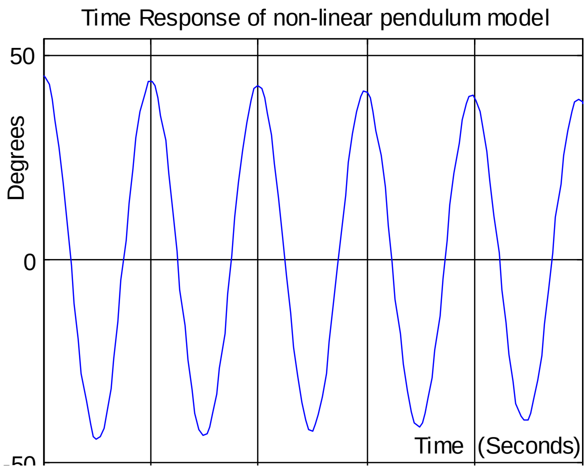


Figure 2.4 Block diagram of the non-linear system



2.2 Rigid link pendulum.

Consider a related problem, the dynamic response for the mechanical system model of the human leg. The transfer function relates the output angular position about the hip joint to the input torque supplied by the leg muscle. The model assumes an input torque, T(t), viscous damping, D at the hip joint, and inertia, J, around the hip joint. Also, a component of the weight of the leg, W = Mg, where M is the mass of the leg and g is the acceleration of gravity, creates a non-linear torque. Assume that the leg is of uniform density so that the weight can be applied at the centroid at L/2 where L is the length of the leg. For definiteness let D = 0.01 lb. s, J= 4.27 ft lb. s^2, W = Mg = 40 pounds, L=3 feet, We will use a torque amplitude of T (t)= 75 ft lbs. of torque.

The pendulum gives us a good model for a robot arm with a single degree of freedom. With a rigid link, it is natural to drive the rotation by a torque applied to the pinned end and to represent the mass at the center of mass of the link. Other physical variations lead to different robot designs. For example, if we mount the rigid link horizontally and then articulate it, we reduce the effect of gravity on the motion. This option is used in the Cartesian, cylindrical and SCARA (Selective Compliant Articulated Robot for Assembly) robot designs. The spherical and vertically articulated industrial robots have the rigid link pointed upward. A gantry or an overhead-mounted, vertically articulated industrial robot has the link pointing down as in the simple pendulum.

We can draw a free body diagram, as shown in Figure 2.6, for the application of the rotational form of Newton's second law about the pinned end to balance the torque. The angle of motion is shown with positive direction counter clockwise. The motion is resisted by three torque: the component of weight is (MgL/2)·sin θ; the damping torque is D·(dθ/dt); and the inertial torque is $J \cdot d^2\theta/dt^2$. For a bob mass at the end of a link, the inertia is $J = ML^2$. However, for a distributed link the inertia is only $ML^2/12$.

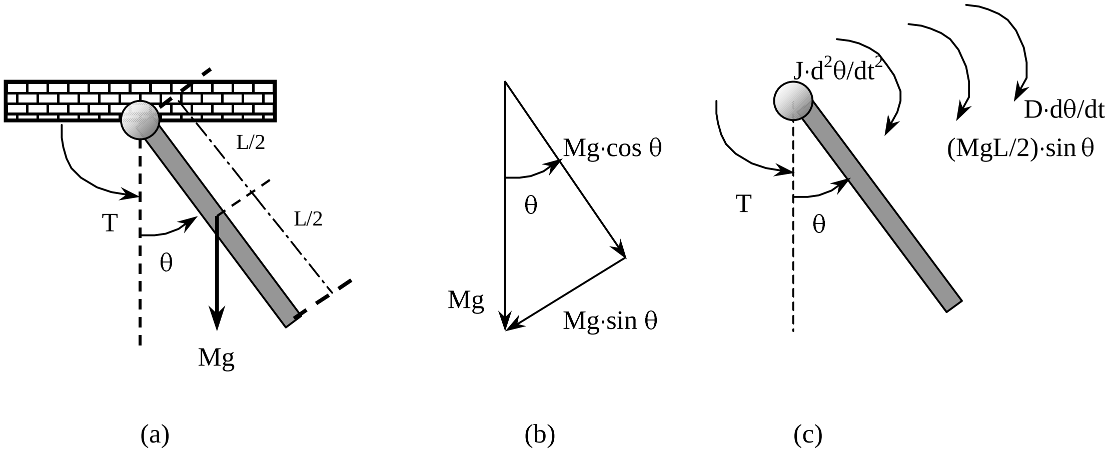


Figure 2.6 Rigid link pendulum structure diagram. (a) Physical diagram; (b) Components of weight vector; (c)Free body torque

We can write the differential equation that describes the system dynamic response and obtain both non-linear and linear equations. A solution can be developed by using the rotary system torque balance.

$$J\frac{d^2\theta}{dt^2}+D\frac{d\theta}{dt}+\frac{MgL}{2}\sin\theta=T(t)$$

~~(~~2.37

Using the small angle approximation, sin θ = θ gives

$$J\frac{d^2\theta}{dt^2}+D\frac{d\theta}{dt}+\frac{MgL}{2}\theta=T(t)$$

~~(~~2.38

Equation 38 is a linear form. Since it is linear, we can take the Laplace transform to obtain the transfer function between the output and input.

$$\frac{\Theta(s)}{T(s)}=\frac{1}{Js^2+Ds+\frac{MgL}{2}}=\frac{\frac{1}{J}}{s^2+\frac{D}{J}s+\frac{MgL}{2J}}$$

~~(~~2.39

It is also interesting to show how the equations simplify for a pendulum mounted in a horizontal plane rather than a vertical plane. For a horizontally articulated pendulum or robot,

the weight is perpendicular to the motion and does no work so the equation simplifies as:

$$T(t)=J\frac{d^2\theta}{dt^2}+D\frac{d\theta}{dt}$$

~~(~~2.4

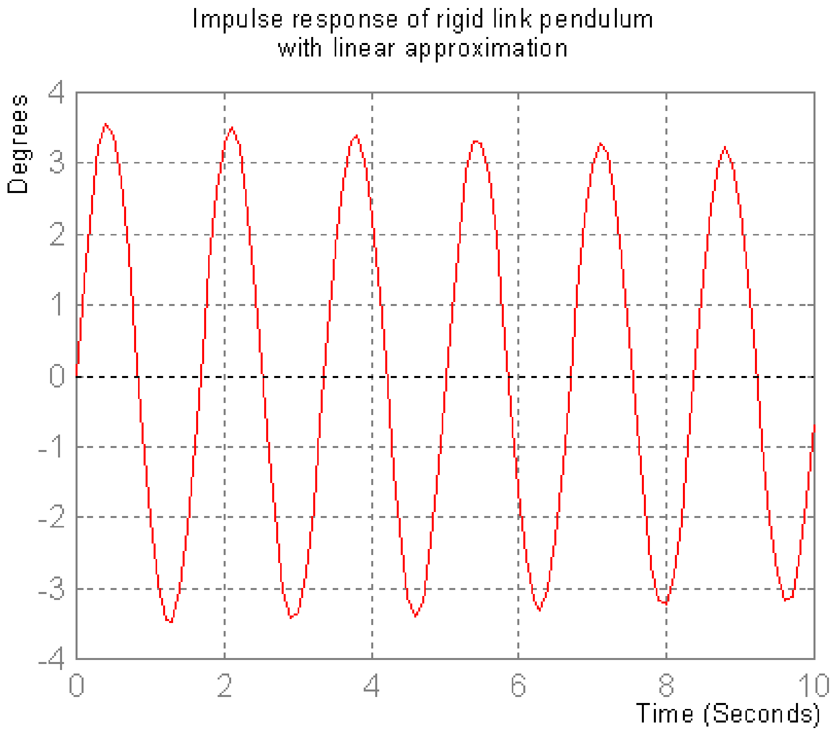
This linear system is easier to control and shows why the SCARA robot design has become popular. The SCARA robot assembles items accurately and efficiently as well.

We can also develop a Matlab m-file solution to this linear differential equation.

```
J = 4.27;
D = 0.1;
M = 40/32.2;
g = 32.2;
L = 3;
num = [0, 180/3.14159];  % 180/3.14159 is to translate radians into degrees
den = [J, D, M*g*L/2 ];
t= 0:0.1:10;
impulse(num,den,t); %find impulse response
grid on;
xlabel=(‘Degrees’);
ylabel=(‘Time(seconds)’);
title(‘Unit impulse response of the rigid link pendulum');
```

When one runs this program using Matlab, it produces the result shown in Figure 2.7.

One can also use Simulink to develop a graphical model and solve the non-linear system. To develop the block diagram recall that T(t) is the input and θ is the output. We can manipulate the differential equation and develop the block diagram. Various forms of the block diagram may be developed depending on how one solves the equation. One form is shown in Figure 2.8. When the torque step input is T(0+)=75, the time response is as shown in Figure 2.9. Rather than oscillating, the angle output appears to be going to infinity. This corresponds to the rigid link rotating continuously about its axis.



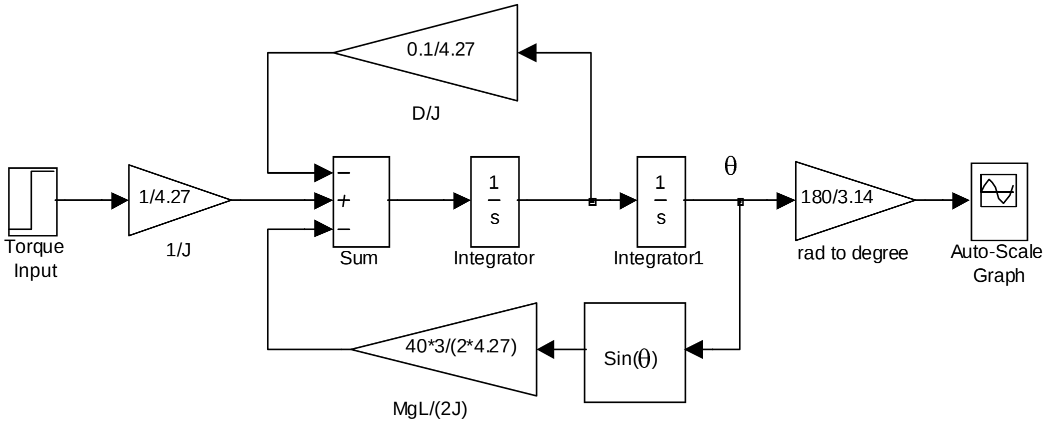


Figure 2.8 Block diagram of the non-linear rigid link system with torque

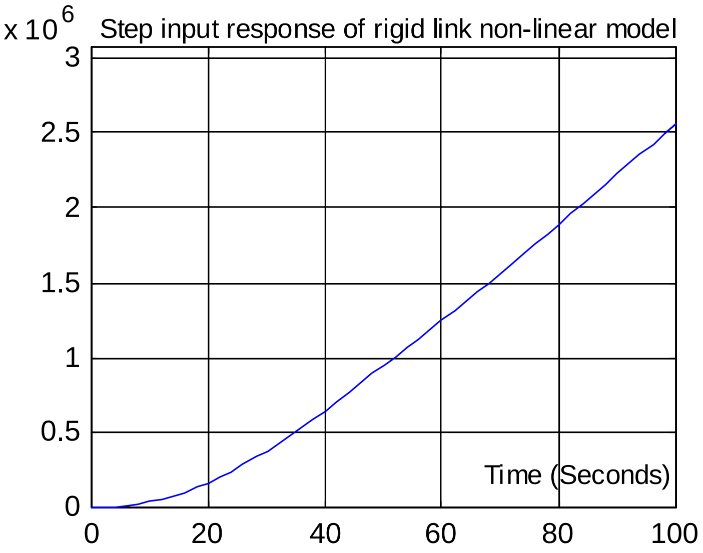


Figure 2.9 Response of the non-linear rigid link model
when step input T(0)=75 ft lbs

2.3 Motorized Robot Arm

As previously mentioned, a rigid link model is in fact the basic structure of a robot arm with a single degree of freedom. Now let us add a motor to such a robot arm.

A DC motor with armature control and a fixed field is assumed. The electrical model of such a DC motor is shown in Figure 2.10. The armature voltage, $e_a(t)$ is the voltage supplied by an amplifier to control the motor. The motor has a resistance R_a, inductance L_a and back electromotive force constant, K_b. The back emf voltage, $v_b(t)$ is induced by the rotation of the armature windings in the fixed magnetic field. The counter emf is proportional to the speed of the motor with the field strength fixed. That is,

$$v_b(t) = K_b \frac{d\theta}{dt}$$

~~(57)~~2.

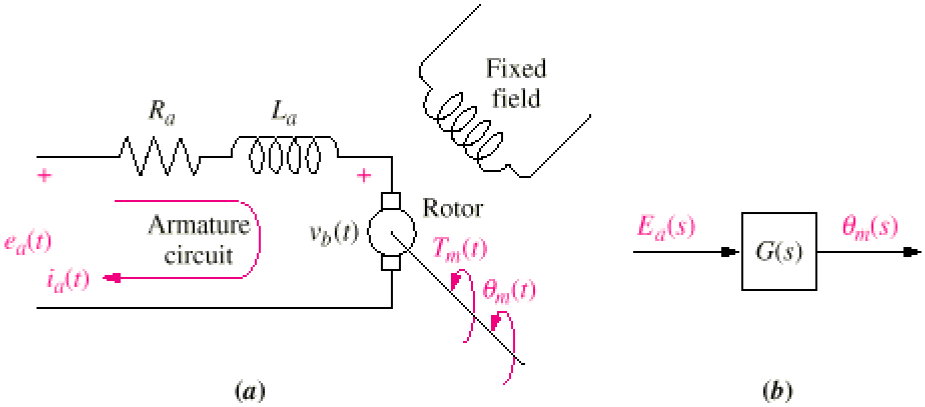


Figure 2.10 Fixed Field DC motor (a) circuit diagram; (b) block diagram (From Nice)

Taking the Laplace transform gives:

$$V_b(s) = sK_b\,\Theta(s)$$

~~(58)~~2.

The circuit equation for the electrical portion of the motor is:

~~(59)~~2.

$$E_a\ (s)= R_a\ I_a\ (s)+ L_a\ sI_a\ (s)+V_b\ (s)$$

This may also be written as:

$$I_a(s) = \frac{E_a(s) - K_b s\Theta(s)}{L_a s + R_a}$$ ~~(59)~~2.

The torque developed by the motor is proportional to the armature current.

$$T_m(s) = K_t I_a(s)$$ ~~(59)~~2.

This torque moves the armature and load.

Balancing the torques at the motor shaft gives the torque relation to the angle that may be expressed as follows

$$T(t) = J\frac{d^2\theta_m}{dt^2} + D\frac{d\theta_m}{dt}$$ ~~(59)~~2.

where

θ_m = *the motor shaft angle position,*

J = all inertia connected to the motor shaft

D= all friction (air friction, bearing friction, etc.) connected to the motor shaft

Taking the Laplace transform gives:

$$T_m(s)= Js^2\Theta_m(s)+ D\ s\Theta_m(s)$$

2.47

Solving Equation (63) for the shaft angle, we get:

(~~59~~64)

$$\theta_m(s) = \frac{T_m(s)}{Js^2 + Ds}$$

If there is a gear train between the motor and load, then the angle moved by the load is different from the angle moved by the motor. The angles are related by the gear ratio relationship, which may be derived by noting that an equal arc length, S, is traveled by two meshing gears. This can also be described by the following equation: 2.48

$$S = R_m \theta_m = R_L \theta_L . \qquad 2.49$$

The gear circumference of the motor's gear is $2\pi R_m$ that has N_m teeth and the gear circumference of the load's gear is $2\pi R_L$ that has N_L teeth so the ratio of circumferences is equal to the ratio of radii and the ratio of number of teeth so that

$$N_L \theta_L = N_m \theta_m \qquad 2.50$$

or

$$\frac{\theta_L}{\theta_m} = \frac{N_m}{N_L} = n \qquad 2.51$$

The gear ratio may also be used to reflect quantities on the load side of a gear train back to the motor side so that a torque balance can be done at the motor side. Assuming a lossless gear train, it can be shown by equating mechanical , $T\omega_1$, and electrical, EI, power that the quantities such as inertia, J, viscous damping D, and torsional springs with constants K may be reflected back to the motor side of a gear by dividing by the gear ratio squared. This can also be described with the equations below:

$$J_{mL} = \frac{J_L}{n^2}$$

$$D_{mL} = \frac{D_L}{n^2}$$

$$K_{mL} = \frac{K_L}{n^2}$$

Using these relationships, the equivalent load quantities for J and D may be used in the previous block diagram.

From equation 2.59, 2.60, 2.61, 2.64, 2.67 we can get the block diagram of the armature-controlled DC motor as shown in Figure 2.11.

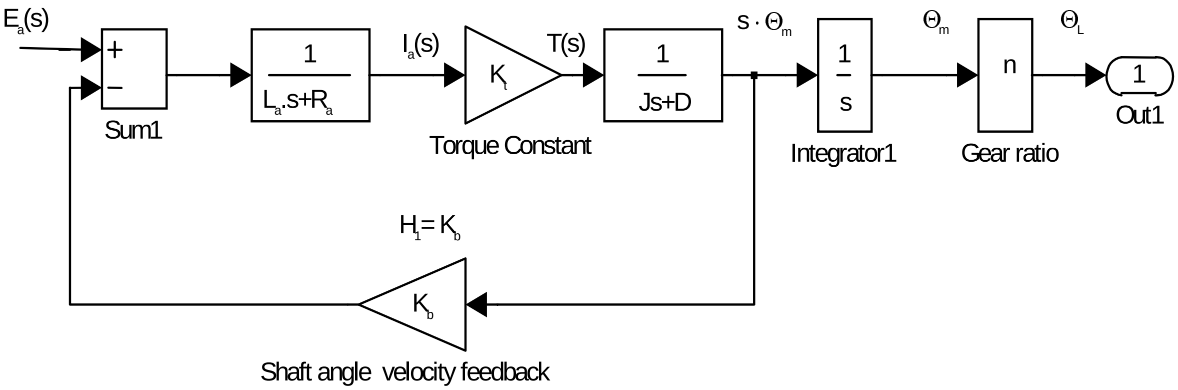


Figure 2.11 Armature-Controlled DC Motor Block Diagram

By simplifying the block diagram in Figure 2.11, we can get the Armature-Controlled motor transfer function as:

$$G(s) = \frac{\Theta_L(s)}{E(s)} = \frac{K_t n}{s\left[(Js + D)(L_a s + R_a) + K_b K_t\right]}$$

$$G(s) = \frac{K_t n}{s\left[(JL_a s^2 + (JR_a + DL_a)s + DR_a + K_b K_t\right]} \qquad 2.55$$

As we can see, this model is of the third order. However, in the servomotor case, the inductance of the armature L_a could usually be ignored. Thus this model could be reduced to a second order system.

Now, apply this model to a simple example. Suppose a DC motor is used to drive a robot arm horizontally as shown in Figure 2.12. The link has a mass, M=5Kg, length L=1 m, and viscous damping factor D = 0.1. Assume the system input is a voltage signal with a range of 0-10 volts. This signal is used to provide the control voltage and current to the motor. The motor parameters are given below. The goal is to design a compensation strategy so that a voltage of 0 to 10 volts corresponds linearly of an angle of 0° to an angle of 90°. The required response should have an overshoot below 10%, a settling time below 0.2 second and a steady state error of zero. The motor parameters are given below.

J_a = 0.001 kg-m^2/ s^2

D_a = 0.01 N-m s/rad

R_a = 1 Ohm

L_a = 0 H

K_b = 1 V-s/rad

K_t = 1 N-m/A

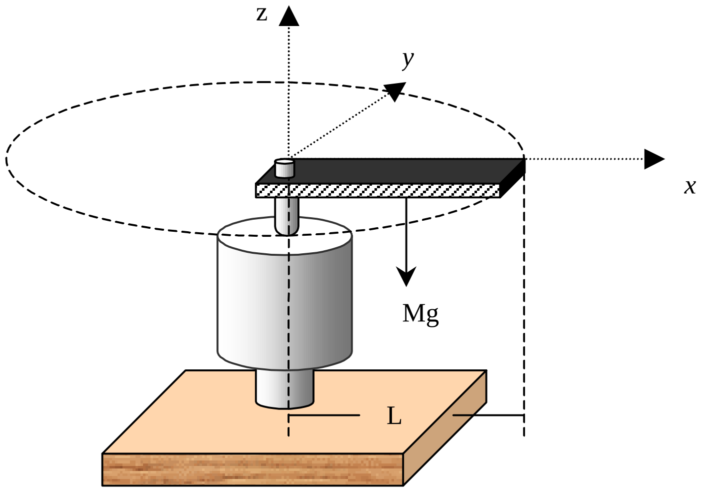


Figure 2.12 A single joint robot arm driven by an armature-controlled DC

First, consider a system without gears or a gear ratio of 1. The inertia of the rigid link as defined before is:

$$J_L = \frac{ML^2}{12} = \frac{5\times 1^2}{12} = 0.4167 kg.m^2 \qquad 2.56$$

According to the impedance reflection model established before, the total inertia J and total damping factor D are:

$$J = J_a + J_L = 0.001+0.4167 = 0.4177 \text{ Kg.m}^2$$

$$D= D_a + D_L= 0.01+0.1=0.11$$

Substituting the known values into equation 2.56 we can get:

$$G(s) = \frac{\Theta_L(s)}{E(s)} = \frac{1}{s\left[(0.4177s + 0.11)(1 + 0 \times s) + 1 \times 1\right]} \tag{2.57}$$

$$G(s) = \frac{1}{s(0.4177s + 1.11)} \tag{2.58}$$

The above process could be calculated with Matlab scripts as follows:

```
J=0.001+0.4167;
D=0.1+0.01;
La=0;
Ra=1;
Kt=1;
Kb=1;
n=1;
Num=Kt*n;
Den=[J*La J*Ra+La*D D*Ra+Kt*Kb 0];
step(Num, Den);
title(‘Step Response of the Motorized Robot Arm’);
End;
```

The step response of this system is shown in Figure 2.13. As we can see the system does not go to a steady state value, but to an increasing angle at constant value This means the armature rotates at a constant speed, which is achieved by its built-in velocity feedback factor K_b

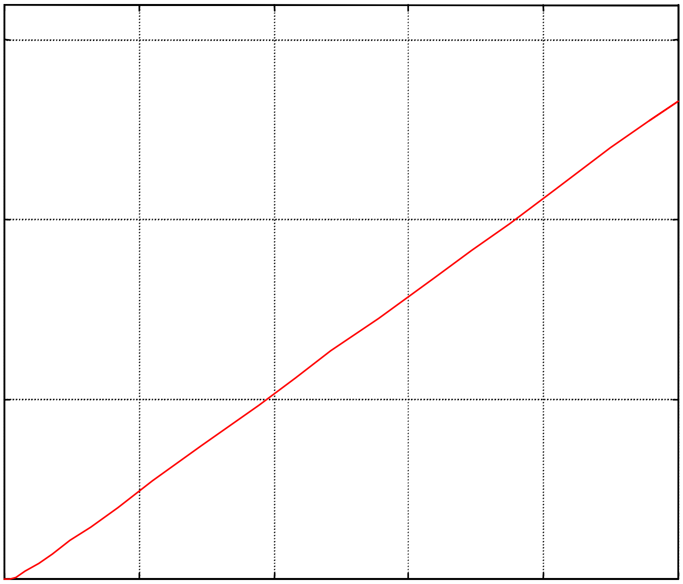


Figure 2.13 The motorized robot arm with velocity feedback step response

However, we want the motor to move the robot arm to a proper angular position corresponding to the input. This can be achieved by a positional servomechanism, i.e., we feedback the actual robot arm position, θ_L , convert the position information into voltage via a constant K_p and then negatively feed back this signal back into the system. Accordingly, the feedback signal in voltage is $E_f = \theta_L \cdot K_p$ where K_p (V/degree) is the proportional constant depending on the input and desired position output.

A simple position sensor is a potentiometer, a variable resistor. The resistor can measure rotational position. A DC voltage V is applied to both ends of the resistor. R_1 is the resistance from the pointer to one fixed end, while R is the total resistance. R_1 is proportional to the

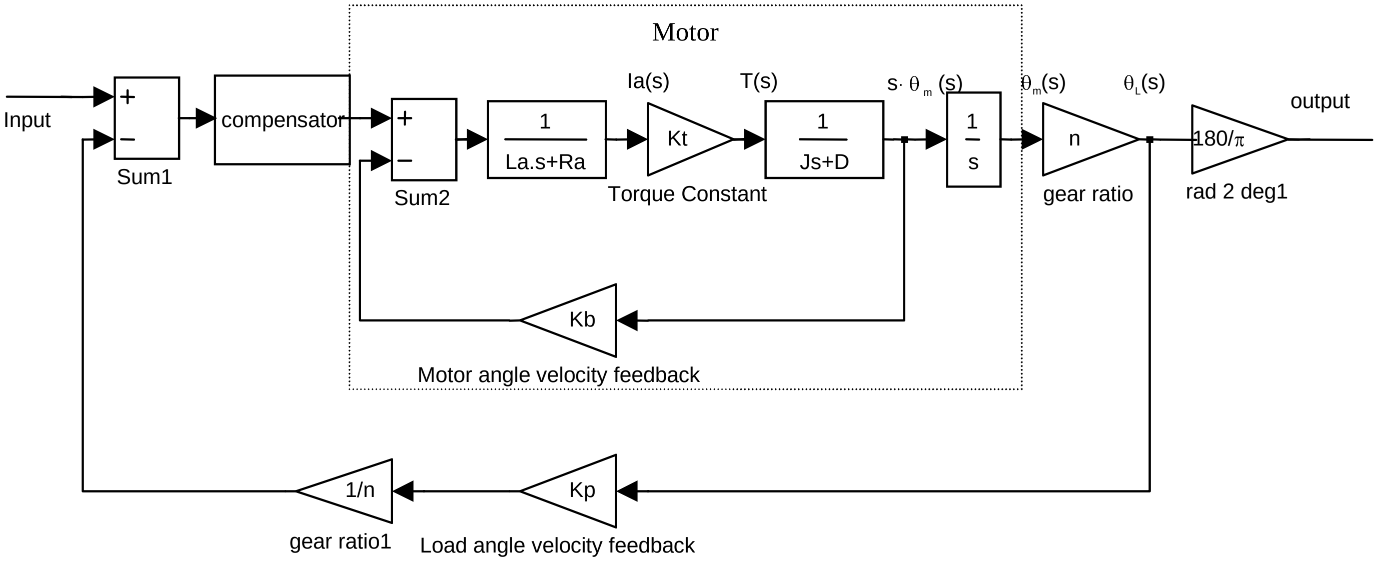


Figure 2.14 Position and velocity feedback model of the motorized rigid link

The revised system block diagram is shown in Figure 2.14. Suppose the voltage between both ends of the potentiometer is 10V, and that K_p is the ratio of the voltage change to the corresponding angle change. In this case, K_p=(10-0)/(90-0)=0.1111 V/degree; The gear ratio is 1.

The new transfer function is:

$$G'(s) = \frac{G(s)}{1 + G(s) \cdot K_p}$$

$$G'(s) = \frac{1}{0.4177s^2 + 1.11s + 0.0011}$$

The step response can be determined with the following program

```
V=10;

Angle=90;

Kp=V/Angle; %feedback voltage/angle constant

G=tf([1],[0.4177 1.11 0]);

% the transfer function of the velocity loop

sysclose=feedback(G,Kp);

%the closed loop function of position feedback

step(sysclose);

end
```

After position feedback, the steady response tends to be stable as shown in Figure 2.15.

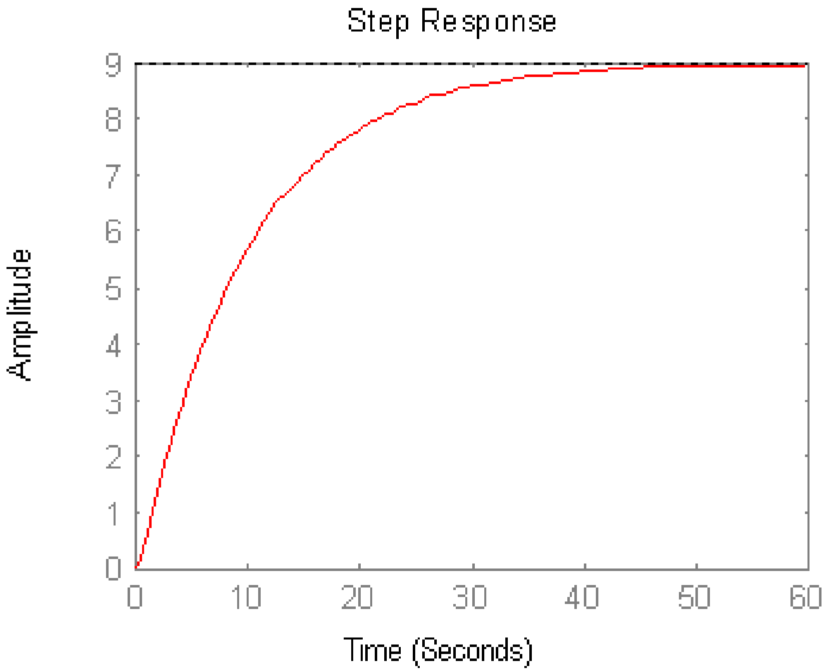


Figure 2.15 Step response of the position feedback system

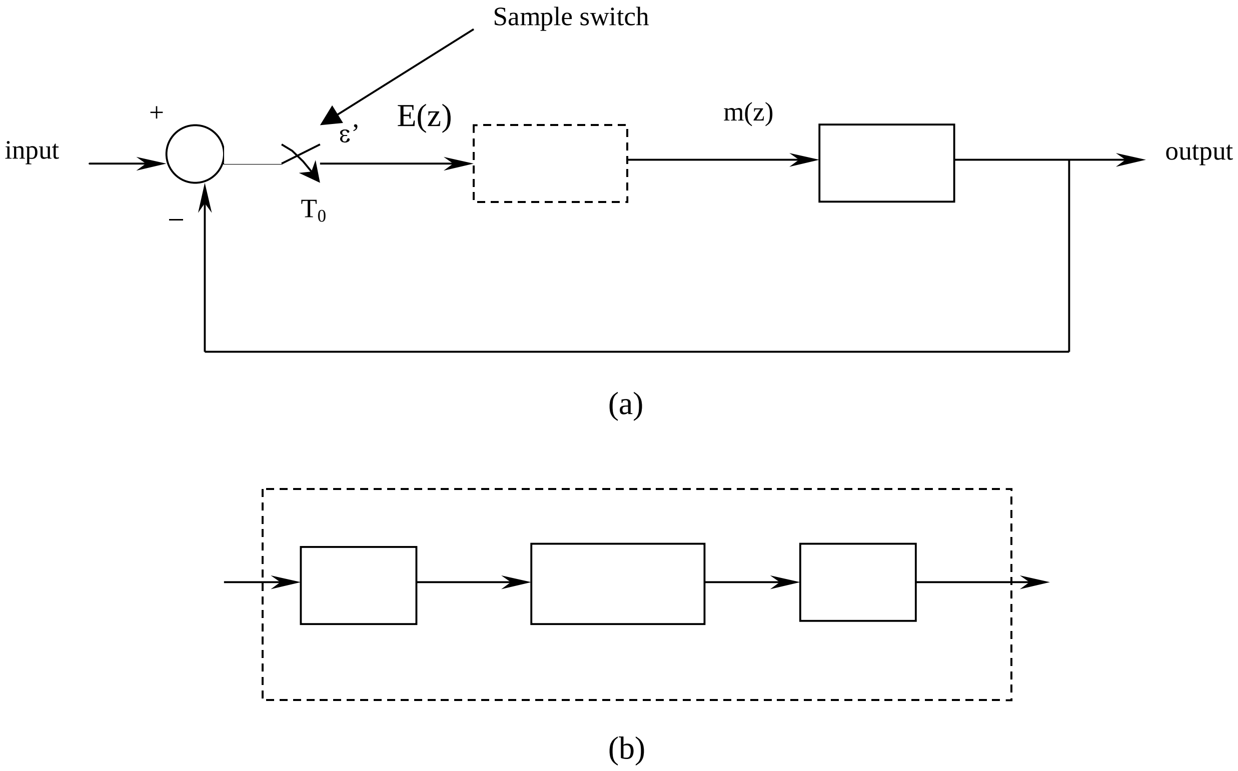
2.4 Digital Motion Control

2.4.1 Digital Controller:

With the many computer applications in control systems, digital control systems have become more important. A digital system usually employs a computerized controller to control continuous components of a closed loop system. The block diagram of the digital system is shown in Figure 2.16. The digital system first samples the continuous difference data ε, and then, with an A/D converter, changes the sample impulses into digital signals and transfers them into the computer controller. The computer will processes these digital signals with predefined control rules. At last, through the digital to analog (D/A) converter, the computing results are converted into an analog signal, m(t), to control those continuous components. The sampling switch closes every T_0 seconds. Each time it closes for a time span of h with $h<T_0$. The sampling frequency, f_s, is the reciprocal of T_0, $f_s= 1/T_{0.}$ And $\omega_s =2\pi/T_0$ is called the sampling angular frequency. Digital controller provides the system with great flexibility. It can achieve compensation values that are hard for analog controllers to obtain.

2.4.2 Digital-controlled Servo System:

A servo system is a motor control system that combines various components to automatically control a machine’s operation. It is generally composed of a servo drive, a motor, and feedback device. The controller gets a feedback signal and outside control signal and controls the servo drive and motor together to precisely control torque, velocity, or position of the motor shaft. The feedback continuously reports the real time status, which is compared to the command value. Differences between the command position and feedback signals are automatically adjusted by the closed-loop servo system. This closed-loop provides the servo system with accurate, high performance control of a machine.



(a)

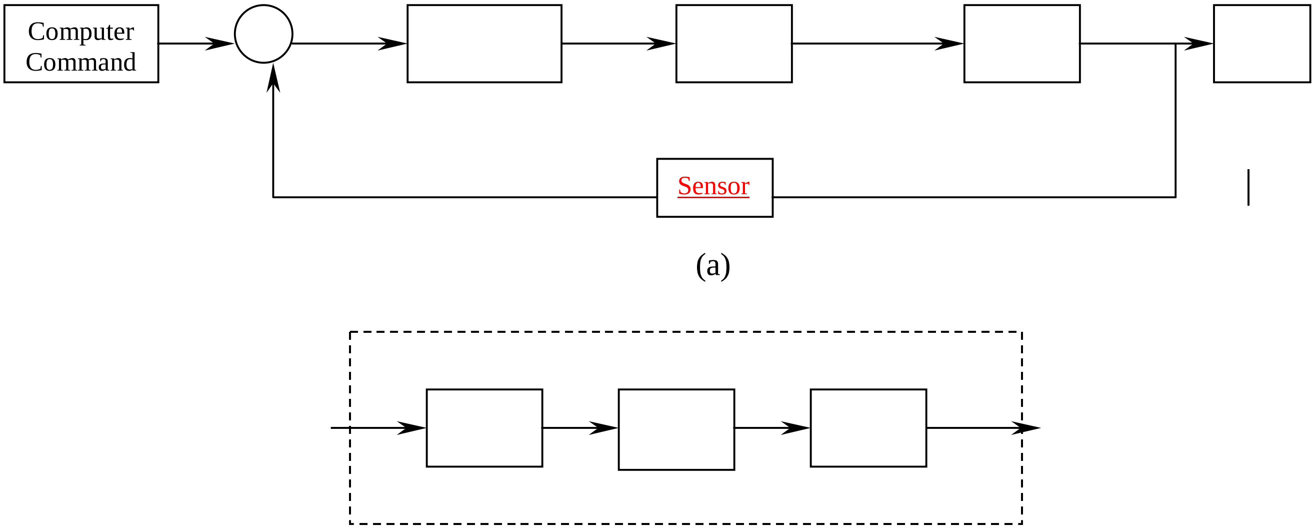
(b)

Figure 2.16 Two representations of digital control systems
(a) digital control system (b) digital controller

A servo motor is characterized by a long, thin configuration. This design permits the motor to provide torque while minimizing the growth in rotor inertia. It results in an improved mechanical time constant and faster time response.

The controller of the servo system is, in most cases, programmable. It allows one piece of equipment to do many related jobs or functions by selecting a different program.

A typical computer-controlled servo system is shown in Figure 2.17. It has three main elements, a digital controller, an amplifier, a motor and an encoder. The controller is composed of a digital filter, a zero-order-hold (ZOH) and a Digital to Analog Converter (DAC). The purpose of the controller is to compensate the system to make an unstable system become stable. This is usually achieved by adjusting the parameters of the filter. The controller accepts both encoder feedback and commands from the computer and finds the error between them. The error signal passes through the digital filter, ZOH and DAC to generate control signals to control the amplifier (AMP). The amplifier amplifies the control signals from the controller to drive the motor. The encoder is usually mounted on the motor shaft. When the shaft moves, it generates electrical impulses, which are processed into digital position information. This position information is then feedback directly into the controller. The mathematical model of the above components could be varied among different products. However, they tend to be the same in most aspects.



(a)

(b)

Chapter 3 Motion control systems modeling

3.1 Systems Introduction

In general the robot motion including moving forward and backward, turning left and right as well as stop. It often happens during the Bearcat I operation that the robot will miss the control line when there is a small radius turn on the track. This requires a built in mechanism that can move the robot in a sharp turn.

One neat solution of Bearcat II motion control is to drive two wheels with independent motors. These motors drive the left and right wheel respectively through two independent gearboxes, which increase the motor torque by about forty times. An amplifier will deliver the power to each individual motor. If one wheel rotates forward while the other wheel rotates backward, the robot will turn about its axis. Thus the ZTR is accomplished. With ZTR, the robot will gain better ability to run a sharp turn.

The motor is driven by the motor amplifier. The motor amplifier gets its control signal from a control board, which is connected to a computer. The control board is a information center. It gets the motor rotation condition via encoder, which is built into the motor. Then, it transfers the encoder signal into proper data format and then sends it to the main computer. The computer is the final decision-maker on the machine. It gets information from both vision systems and the sonar system, integrate the information, make decisions based on those signals and then send the command to the control board.

The commands are then transferred into voltage (analog) control signal that is applied to the amplifier.

The sonar system could be used to detect obstacles in front of it. This requires another important motion behavior, that is, the rotation of sonar. This requires a fast speed, accurate motor to rotate sonar to the desired angle.

3.2 Bearcat II subsystems description:

The physical structure of Bearcat II is characterized by digital control, vision guidance, ultrasonic distance sensors and emergency stop. The main components of the BEARCAT II includes: the central computer, I-scan and two CCD camera as vision sensor, Plaroid ultrasonic sonar, two 12V batteries as motor power source, GALIL digital controller and two DC motors.

- **Vision Systems**

  The vision guidance system helps the vehicle to know its current position along the road. It needs to be calibrated before its real functioning. The purpose of calibration is to transform the camera coordinates into robot coordinates. Two CCD cameras were used to judge the current position of the robot.

- **Obstacle Avoidance System**:

  The sonar system reliably detected obstacles between 0.5 meter and 6 meters within an accuracy of 0.02 meter. The system interfaces between the Polaroid unit, Galil controller and the supervisory controller were found to be successful. The fuzzy logic controller computed correction angles. To drive the robot around the obstacles going downslope on a ramp, it was observed that the sonar detected the ground as an obstacle. This problem was taken care of by modifying the sonar algorithm such that if sonars on either sides gave identical readings, the obstacle shall be ignored.

- **Steering control system**:

  The steering mechanism worked satisfactorily. Implementing the PID controller to control two identical motors on both sides of the robot. The action of turning is achieved with the speed variations of the two motors. The working status (velocity and position) is continuously reported by encoders, built in the motor.

- **Safety and Emergency Stop Braking System**:

A safety switch and a remote safety switch are in serial connection into the motor power circuit. When bad situation happens, one could press the emergency stop button or remote stop button to stop the robot immediately.

3.3 General descriptions of the motion systems

One neat solution of vehicle motion control is to drive two wheels with independent motors. These motors drive the left and right wheel respectively through two independent gearboxes, which increase the motor torque by about twenty times. An amplifier will deliver the power to each individual motor. The amplifier gets its control signal from a control board, which is connected to a computer. The control board is a information center. It gets the motor rotation condition via encoder, which is built into the motor. Then, it transfers the encoder signal into proper data format and then sends it to the main computer. The computer is the final decision-maker on the machine. It gets information from both vision systems and the sonar system, integrate the information, make decisions based on those signals and then send the command to the control board. The commands are then transferred into voltage (analog) control signal that is applied to the amplifier. The function block diagram is shown in Figure 3.1

According to the master design, we decided to use a rotation sonar to detect obstacles in front of it. This requires another important motion behavior, that is, the rotation of sonar. This requires a fast speed, accurate motor to rotate sonar to the desired angle.

3.4 Equipment Selection:

- Power supply

The robot has two kinds of power supply systems: motor power supply and equipment power supply. The first design will use four 12 V batteries as power resource. One battery was used to supply the equipment via a 12/110V transformer, the other three 12V batteries were serially connected and provide power to the motor amplifiers. This design have put too much load on the robot that its caster often trapped into the grassland. Then, the final design uses only two 12 V batteries in serial connection. One of them is used to supply both motor and the equipment, the other is used in serial connection with this one to provide power supply to the motor amplifiers. The real life operation shows this combination would provide the robot car move for 2 hours of continuous motion.

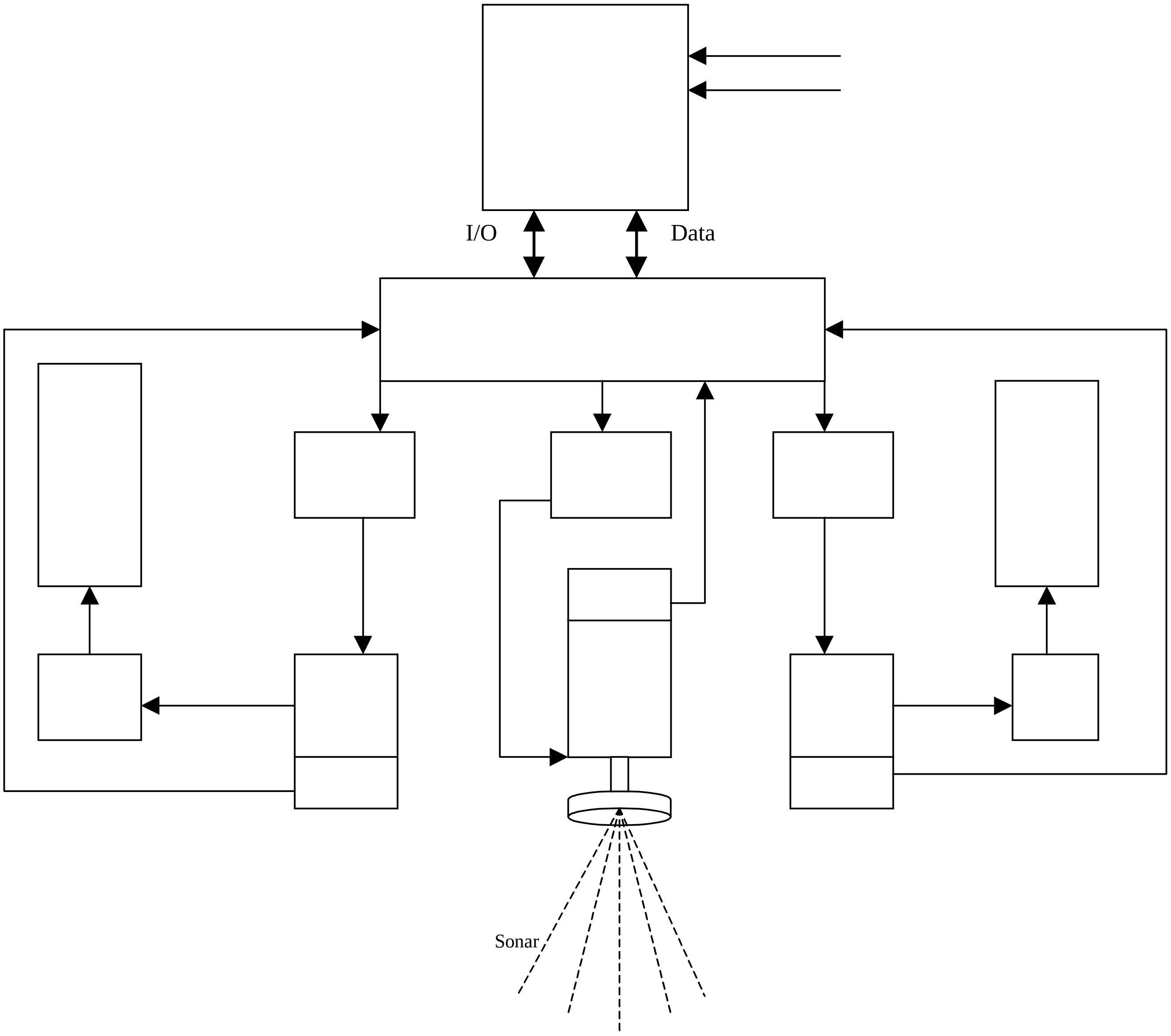
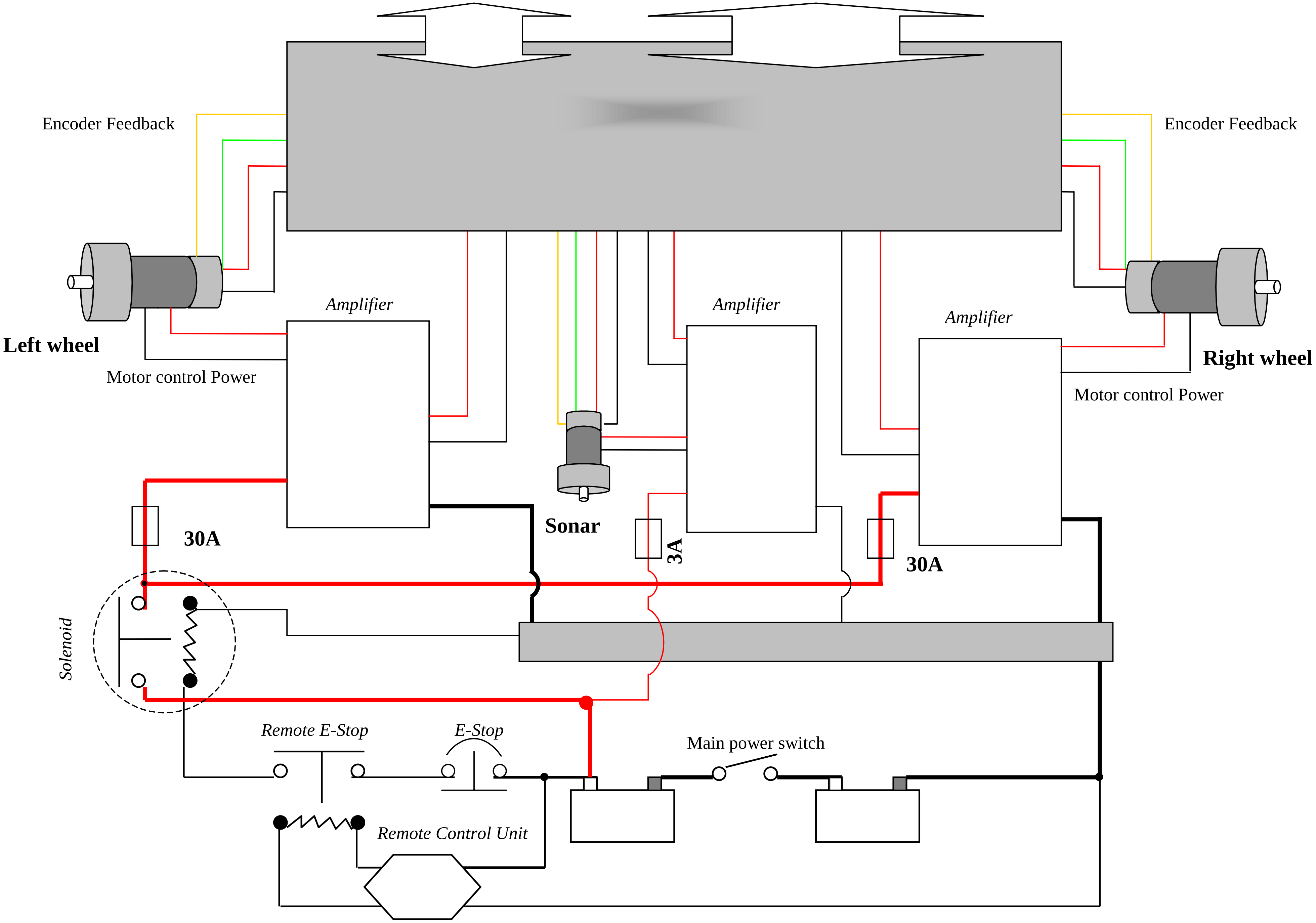


Figure 3.1 functional components of the motion control system

Figure 3.2 BEARCAT II Motion Control Systems Architecture



- Control board

A 3 axis Galil DMC-1000 motion control board was selected as the interface between the computer controller and the drive components. . It is a very flexible motion controller that can be plugged into the PC bus. Some eminent capabilities of this product include: 8 MHz encoder input frequency, 16 bit motor command output DAC, maximum of +/- 2 billion counts total travel per movement, faster sample rate, but interrupts and non volatile memory for parameter storage. It provides high performance and flexibility while maintaining simplicity and low cost.

DMC-1000 provides the maximum systems flexibility. One board can control 3 independent motors simultaneously. One control axis could be sacrificed to extend the control axis to five, six, seven or more. The DMC 1000 can be interfaced to a variety of motors, including servo motors, stepper motors, etc. In this design, two axes were used to control two wheel motors independently, the third axe was used to control sonar motor.

- Wheel motor

The motor that drive wheels will assume the majority of the load when the vehicle is moving. This includes load from both the friction from gearbox and some times, load by the vehicle weight. This requires a high current motor. A good selection is the Electro-Craft 0728-39-003 servo motor. Some specifications are listed as following:

Peak torque 180 lb-in

Continuous torque 36 lb-in

Maximum Current 60 A

Maximum Terminal Voltage 120 V

Maximum Speed 2500 rmp

- Wheel motor amplifier

Since the two robot wheel motors will assume the heaviest load, the amplifier to drive these motors would have high power capacity. Advanced Motion Controls 25A series product can be a good choice. This servo amplifier is to drive brush type DC motors at a high switching frequency. Single red/green LED indicates its operating status. All models are fully protected against over-voltage, over-current, over-heating and short-circuits across motor, ground and power leads. It requires a single unregulated DC power supply. Four 14-turn potentiometers can be adjusted to adapt loop gain, current limit, input gain and offset. Important specifications including: DC supply voltage 20-80 V, peak Current +/-25A. The encoder attached to the shaft of this motor has a parameter of 2000 counts/circle.

- Sonar motor

Three features characterized sonar motion:

a) Fast response. The sonar would be rotated to the desired position at the fastest speed.

b) Small load. The only load mounted on the sonar motor is the sonar, which is less than a quarter of pound.

   Based upon these features, we select Electro-Craft 0260-06-018 servomotor. The encoder of this motor is 2000 counts/circle.

- Sonar motor amplifier

Since the sonar rotation is a light load. A Miniature Brush type Servo Amplifier, MSA-12-80, was selected. It is low-cost, easy to use amplifier for driving brush type servo motors at high switching frequencies. The amplifier utilizes power MOSFETs and surface mount technology to produce high power in a small package. The MSA-12-80 accepts a +/- 10V range input signal directly from Galil control board, or it can be configured as a stand-alone drive. An unregulated DC power supply is requied to drive the MSA-12-80.

- Safety devices

In compliance with the contest rules, there must be at least one remote E-stop existed in the system. The mobile robot should be turned on/off from a distance of no less than 50 feet. The Futaba F0301U was selected to assume this responsibility. Its operation distance can reach over 65 feet under normal condition. The F0301U has 3 channel radio control systems that can control 3 relays independently. The user interface consists of three manually operated membrane switches. The transmitter employs 6VDC to transmit FM signals at 72.470 MHz. The receiver is powered by an external voltage between 9V to 31 VDC. The receiver receives signal from transmitter and transfers it into current , which will be amplified 120 times. This amplified current actives the contacts of the relay that activate the E-stops. The unique feature of this product is its direct sequence data encoding coupled with a 16 bit identification code (65536 distinct address codes) and CRC error checking. This provides us exceptional security and reliability.

Also, there are fuse devices to protect motors and amplifiers from over current.

3.5 Motion systems architecture and descriptions

- Amplifier mode settings

There are generally 3 modes of configurations for each amplifier, the current mode(torque mode), voltage mode and velocity mode. The current mode will provide large torque while the voltage mode will provide fast speed. During the field test, the robot will always have to concur ramps, high friction grassland, etc. So the two wheel motors were configured as current mode. The only load of the sonar motor is the quarter-pound sonar head. So, the load for the sonar amplifier would be small. Voltage mode would be selected to support a prompt response.

- Control board and amplifier configuration

The basic configuration between the motor, amplifier and the Galil controller was shown in Figure 3.2. The Galil motion controller plugs into the PC bus and accepts several high level, ASCII commands. Analog motor command of ±10 V range signal for driving servo amplifiers; 16-bit resolution or. The Galil board is the main interface to transfer data to the computer and also the commands to the amplifier. It gets motor status information via encoder and then reports the data to the computer. The computer command was first sent to the Galil board and then transferred into analog control signal, which is between +/- 10 V via a 16 bit D/A converter (resolution 0003 Volts). This signal was a reference voltage which is applied to the amplifier reference in +/- pins to control the motor direction and speed. Depending on the amplifier mode, the control signal controls current, voltage or speed. The two wheel motor amplifiers was powered by two 24V DC power (2 batteries in serial connection). These amplifiers get control signals from Reference in + and

Reference in -, then output voltage/current power to the motor accordingly. The variation of the reference in error will determine the strength of the power output to the motor.

The encoder attached to the motor will read and give negative feedback information of motor motion status. the negative feedback will cause a difference between the command signal and feedback signal. This difference is called the error signal. The amplifier comapnares the feedback signal to the command signal to produce the required output to the load by continually reducing the error signal to zero. The encoder gets +5V DC power directly from the Galil board. The encoder signal will be transferred to the Galil board via pin *A+ and pin *B+. The motor amplifiers get control signals from the Galil controller pin ACMD* and pin AEN* (* means X or Y or Z, depend on the axis). For the safety consideration, the motor amplifier was protected by 30A fuse and sonar amplifier by 3A fuse.

3.6 Amplifier tuning

There are 4 potentiometers to be adjusted built into the amplifier, loop gain, current limit, reference gain and offset. The loop gain adjustment in voltage & velocity modes. Voltage to current scaling factor adjustment in current mode; The current limit adjusts both continuous and peak current limit by maintaining their ratio(50%); The Reference gain adjusts the ratio between input signal and output variables (voltage, current, velocity); The offset/test is used to adjust any imbalance in the input signal or in the amplifier. Before operation, these potentiometers need to be tuned to gain an optimum performance. The tuning procedures are outlined as following:

1. Initialization

Apply zero speed command to the amplifier. Reference inputs should be grounded. Turn all the potentiometer to minimum settings (counterclockwise).

2. Offset adjustments

   Put the offset switch in the on position. Trim the offset potentiometer for minimum amplifier output current by observing motor drift. Turn test offset switch off.

3. Loop gain adjustments

   Turn clockwise the loop gain potentiometer until the motor shaft oscillates, then back one turn.

4. Reference gain adjustment

   Reference gain determines the ratio between input signal and output variables (voltage, current, velocity). Turn this potentiometer clockwise until the required output is obtained for a given input signal.

5. Current limit adjustment

   It is important to set the current limit so that the instantaneous motor current does not exceed the specified motor peak current rating. The maximum current output of the Advanced Motion Controls 25A8 is +/-25 A, far less than the motor current limit 60A, this potentiometer could be set as its maximum value (14 clockwise turn).

A quardrat encoder is mounted on each drive motor shaft. The encoder feedback position signals, which can be differentiated to provide the velocity feedback into the controller. The AGV moving direction is achieved by differentiating speed of the drive wheel. These are drive wheels whose speeds can be varied according to the change in the direction of the track being followed.

3.7 Motion control factor identification

The control of the vehicle movement in a two dimensional space can be reduced to the control of two variables: the vehicle speed and the orientation of the vehicle

The speed of the vehicle refers to the speed of the mid-point of the vehicle. It can be defined as

$$V = ds / dt \qquad 3.1$$

Where V is the motion speed

s is the change of the moving distance of the robot.

This velocity may also be expressed in terms of the velocities of the left and right wheels as:

$$V = (V_L + V_R) / 2 \qquad 3.2$$

Where V_L is the speed of the left wheel

V_R is the speed of the right wheel

The speeds of these two wheels would determine the speed and orientation of the vehicles.

When $V_L > V_R$ The vehicle would moving toward right

When $V_L = V_R$ The vehicle would moving straight

When $V_L < V_R$ The vehicle would moving toward left

When $V_L + V_R = 0$ and V_L !=0, the vehicle is making ZTR

The orientation angle of the vehicle is also determined by the left and right speeds as

$$d\theta / dt = (V_L - V_R) / W \qquad 3.3$$

Where W is the width of the axle or distance between the wheels.

The most common methods of turning a wheeled vehicle are the steering wheel or with differential left and right wheel drives.

With equation 3.1 and 3.3, the wheel speeds can be controlled by the sum and the difference of the two wheel speeds.

$$V_L + V_R = 2V_m \qquad 3.4$$

$$V_L - V_R = W (d\theta/dt) \qquad 3.5$$

Or

$$\omega_L + \omega_R = (V_m / \pi R) \qquad 3.6$$

$$\omega_L - \omega_R = (W / 2\pi R) . (d\theta / dt) \qquad 3.7$$

Control factor computation

V_m and ΔV are now identified as the inputs to the computer, where V_m is the speed for the middle point of the wheel shaft andΔV is the difference between the two wheel speeds. The outputs are V_L and V_R, the speeds of the two wheels respectively.

From equation 3.6 and equation 3.7, we can get

$$\omega_L = (2V_m + \Delta V) / 2\pi R) \qquad 3.8$$

$$\omega_R = (2V_m - \Delta V) / 2\pi R) \qquad 3.9$$

and

$$V_L = V_m + (\Delta V / 2) \qquad 3.10$$

$$V_R = V_m - (\Delta V / 2) \qquad 3.11$$

Two error variables have been identified to keep the straight-line-tracking, namely the distance error Y and the orientation error θ. The distance error is the perpendicular distance from the vehicle cenroid point to the track center. The orientation error is the angle enclosed by a tangent drawn to the track line at the position of the vehicle and the direction to which the vehicle is heading or the angle enclosed between two consecutive co-ordinates picked by the vision tracking system of the vehicle.

Two different cases were presented to define ΔV, the difference between the velocities of the left and right wheels of the vehicle, from which the individual velocity of the left and right drive wheel is calculated.

Case I

$$\Delta V = K_1 . Y + K_2 . \theta \qquad 3.12$$

Where Y is the distance error

θ is the attitude or orientation error

K_1 and K_2 are the control coefficients.

For example, in a simple case

$$K_1 = 1 \qquad 3.13$$

$$K_2 = 2 . (WV_{max})^{1/2} \qquad 3.14$$

Where V_{max} is the maximum vehicle middle point velocity, which was found to be 1.347 m/h at maximum voltage delivered by the amplifier to the drive motors.

Y and θ are obtained from the vision guidance algorithm.

ΔV is calculated by substituting the values of the K_1, K_2, Y and θ in the equation. From the value of ΔV, the individual velocities of the left and right drive wheel can be calculated.

Case II

In case II, a clock function is used to define dt, the time interval, between two consecutive co-ordinates from the vehicle vision guidance system. $d\theta$ is found from the two consecutive x coordinates, which is got from the vision guidance system.

Substituting these values of dt and $d\theta$, ΔV or ($V_L - V_R$) is calculated from:

$$d\theta /dt = (V_L - V_R) / W \qquad 3.15$$

Both cases were tested on the Bearcat II Autonomous Guided Vehicle. Bearcat II Autonomous Guided Vehicle in the Case II followed the track lines closely than in Case I as it was for a simple case but not an optimal case.

3.8 Motion Control System Modeling

One important feature of Galil controller is the digital PID compensation. The right PID value would help the motor acted accurate and in time. The motion system of the robot helps itself negotiate curves and also avoid obstacles. Thus it needs the motion system closely respond to the computer command. Thus it is critical task to setup correct PID values.

3.8.1 the Motor and the amplifier

Modeling the Amplifier can be configured in three modes namely, voltage loop, current loop and the velocity loop. The transfer function relating the input voltage V to the motor position P depends upon the configuration mode of the system as illustrated in Figure 3.3

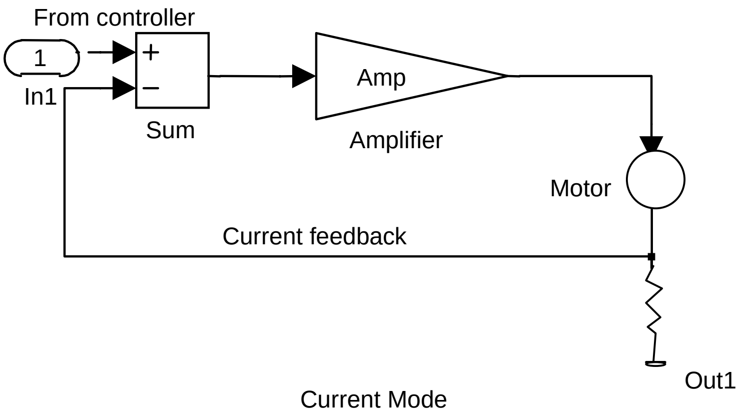
(a) Voltage drive

(b) Current drive

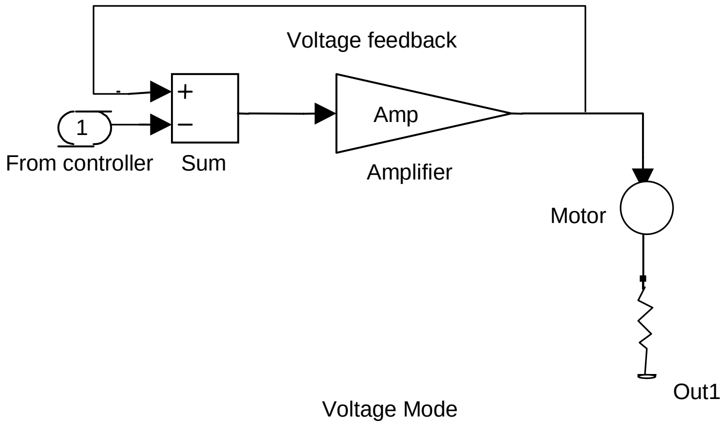
(c) Velocity loop

- Voltage drive

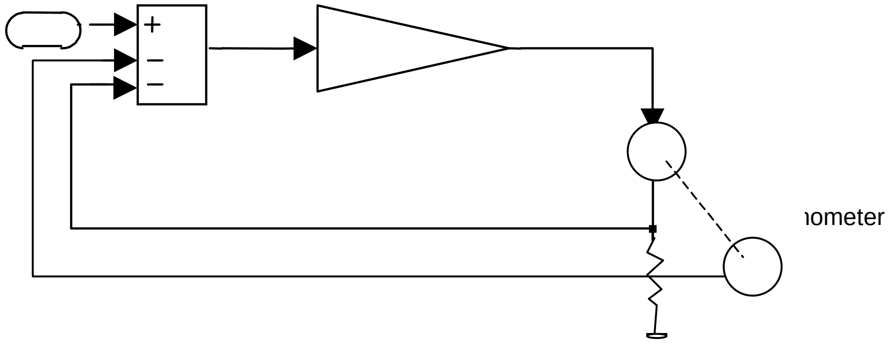
In the voltage loop, the amplifier works as a power source. It receives the control voltage from the controller and amplifies it into the motor drive voltage. The system gain due to this amplifier is K_v[volts/volt]. The transfer function relating the control voltage V and the motor position Θ are similar to those of the armature-controlled position model. It may be written as:



Current Mode



Voltage Mode



$$\frac{\Theta}{V} = \frac{K_v}{K_t s(s\tau_m + 1)(s\tau_e + 1)} \qquad [rad\ /\ volt\ ] \tag{3.17}$$

where

$$\tau_e = \frac{L}{R} \qquad [\sec\ ]$$

$$\tau_m = \frac{J}{D} \qquad [\sec] \tag{3.18}$$

In the voltage loop, the amplifier works as a power source. It receives the control voltage from the controller and amplifies it into the motor drive voltage. The system gain due to this amplifier is K_v[volts/volt]. The transfer function relating the control voltage V and the motor position Θ are similar to those of the armature-controlled position model. It may be written as:

- Current Drive

In current drive mode, the amplifier is configured to generate current, which is proportional to the input voltage V. The current drive will contribute to the system a gain of K_a [Volt/Amp]. The resulting transfer function in this case is

$$\frac{P}{V} = \frac{K_a K_t}{J \cdot s^2} \qquad 3.19$$

Where K_t and J are the same as described before.

- Velocity Loop

The motor velocity can be sensed in two ways:

1. Attach a tachometer on the motor shaft to directly feedback a voltage signal into the amplifier.
2. Differentiate the feedback signals from a position sensor, such as a encoder digital feedback.

The velocity loop block diagram is illustrated in Figure 3.3 (c)

The transfer function between the input voltage and the velocity is

$$\frac{\omega}{V} = \frac{K_a K_t}{J \cdot s[1 + K_a K_t K_g / Js]} = \frac{1}{K_g (s\tau + 1)} \qquad 3.20$$

where $\tau = J/K_a K_t K_g$

The position transfer function could be written as:

$$\frac{P}{V} = \frac{1}{K_g s(s\tau + 1)} \quad 3.21$$

- **The Encoder**

The motor shaft position feedback from increamental encoder with two channels in quadrature, CHA and CHB on the Galil Board. Any resolution encoder may be used as long as the maximum frequency does not exceed 500,000 quadrature states/sec.

In this design, the encoder has two signals A and B, which are in quadrature. The resolution of the encoder is increased to 4 times the number of the pulses generated by the encoder per revolution.

The encoder gain can thus be represented by:

$$K_f = \frac{4N}{2\pi} \; [counts/rad] \quad 3.22$$

- **The Controller**

The controller has a built-in digital comparator. The digital comparator calculates the binary arithmatic difference (error) between the desired position and the actual motor shaft position. The error voltage, passing through the digital filter, DAC and ZOH, generates the control voltage, which provides the input to the amplifier.

- **DAC**

The DAC (Digital to Analog Converter) converts the digital signal into an analog voltage. The input range of the numbers is +/- 10V. The output is a 16 bit digital signal. Therefore, the effective gain of the DAC is K_d= 20V/2^16= 0.0003 V/count.

- **Digital Filter**

The PID compensator is the most commonly used compensator in the feedback control system. Here, P stands for speed of response, D stands for the stability and damping and I stands for accuracy at the cost of response time and stability. It is generally defined as:

$$w(t) = K_p r(t) + K_I \int_0^t r(\tau) d(\tau) + K_d \frac{dr(t)}{dt} \qquad 3.23$$

Where w(t) is the PID controllor response and r(t) is the input.

After taking the Laplace transform this equation produces the function:

$$W(s) = (K_p + \frac{K_I}{s} + K_d s) R(s) \qquad 3.24$$

or

$$\frac{W(s)}{R(s)} = K_p + \frac{K_I}{s} + K_d s \qquad 3.25$$

A digital PID controller also replicates the above equation in the base frequency band if the sampling rate is sufficiently high. This condition will be assumed in this discussion. Also, the multiplication, integration and differentiation are performed numerically. Several algorithms are available to perform numerical integration and numerical differentiation. Assume the sample period is T. After taking the z transformation, the digital compensator has the transfer function given below.

$$\frac{W(z)}{R(z)} = D(z) = K_p + \frac{K_I Tz}{z-1} + K_d \frac{(z-1)}{Tz} \qquad 3.26$$

ZOH

The zero-order-hold models the sampling especially the effects of sampling time. The ZOH holds the output signal value at the constant value of the input until after the next sampling time. The ZOH can be modeled by the following transfer function:

$$H\ (s)\ =\ \frac{1}{(1+\ s\ \frac{T}{2})} \qquad 3.27$$

3.27

For example, if the sampling time is 0.02 second, then H(s)=1/(1+0.01s)

Here we take the sampling time as T=0.001 Second, H(s) becomes:

3.28

$$H(s) = \frac{2000}{(s + 2000)}$$

In some cases, H(s), can be neglected or approximated as one.

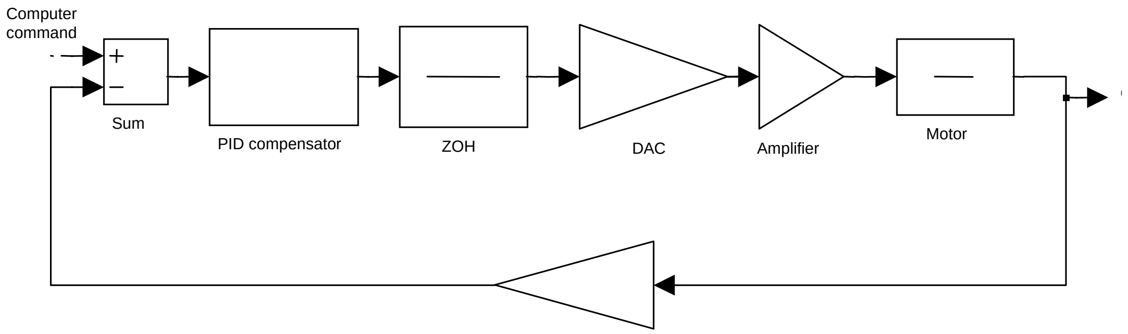
Having modeled the system, we now have to obtain the transfer functions with the actual system parameters. This is done for the system as follows.

3.9 Motion system mode selection

Among the three modes of the amplifier motor configuration, current mode was selected for the motor wheel control loop. Because the vehicle will run across rugged grasses and sometimes climb ramp, high torque will be needed during its operation. For the sonar rotation loop, voltage control will be used since its load is light.

3.10 Compensation design

According to the above description, a motion control system block diagram was constructed as shown in Figure 3.4



The transfer functions of each block can be get from product pamphlet as follows:

The torque constant of wheel motor 0728-39-003 K_t = 2.98 lb-in/amp = 0.336 Nm/A

Combined inertia of motor and load J = 0.00041 $Kg.m^2$.

The winding resistance 0.42 ohm

The AMC Amplifier gain Ka = 4 (adjustable)

The Encoder resolution on the wheel motor N=500

The sample time of the Galil board was set as 1ms or 0.001 second

It is desired that after the compensation, the system would have a phase margin of 45 degree, also a gain margin of 100 or above.

The transfer functions of each component could be set as below

ZOH:

$$H(s) = \frac{2000}{(s + 2000)} \qquad 3.29$$

DAC:

$$K_d = \frac{20}{2^{16}} = 0.0003 \quad V / count \qquad 3.30$$

Encoder:

$$K_f = \frac{4 \times 500}{3.14} = 636 \qquad 3.31$$

Motor and Amplifier:

$$M(s) = \frac{0.336}{0.00041 s^2} = \frac{819.5}{s^2} \qquad 3.32$$

From above, the Open loop transfer function is

$$L(s) = Kf \cdot Kd \cdot M(s) \cdot H(s)$$

$$L(s) = \frac{128.6}{0.00041 s^3 + 0.82 s^2} \qquad 3.33$$

Suppose we need the crossover frequency ω_c=50, we can get the magnitude of the uncompensated system as

$$\left| L(j50) \right| = 0.0612 \qquad 3.34\ 3.33$$

The phase margin of the open loop function is

$$\arg[L(j50)] = -180 - \tan^{-1}[\frac{50}{2000}] = -181.432^{\circ} \qquad 3.35$$

Suppose we use G(s) as the compensation function, A(s) as the system after the compensation. Then we have

$$|A(s)|=|G(s)|\,|L(s)| \qquad 3.36$$

and

$$\arg[A(s)]=\arg[G(s)]+\arg[L(s)] \qquad 3.37$$

According to the design goal, |A(s)|=1 and arg[A(s)] = -135° 3.38

So,

$$G(50) = |A(50)|/|L(50)| = 1/0.0612 = 16.34 \qquad 3.39$$

$$\arg[j50] = -135+181.432 = 46.43^\circ \quad (3.40)$$

thus, we get the compensator function

$$G(s) = 16.34*\tan(46.43^\circ)j+16.34*\mathrm{ctg}(46.43^\circ) \quad (3.41)$$

$$G(s) = 17.17j+15.54 \quad (3.42)$$

The bode plot of the system was given as Figure 3.5

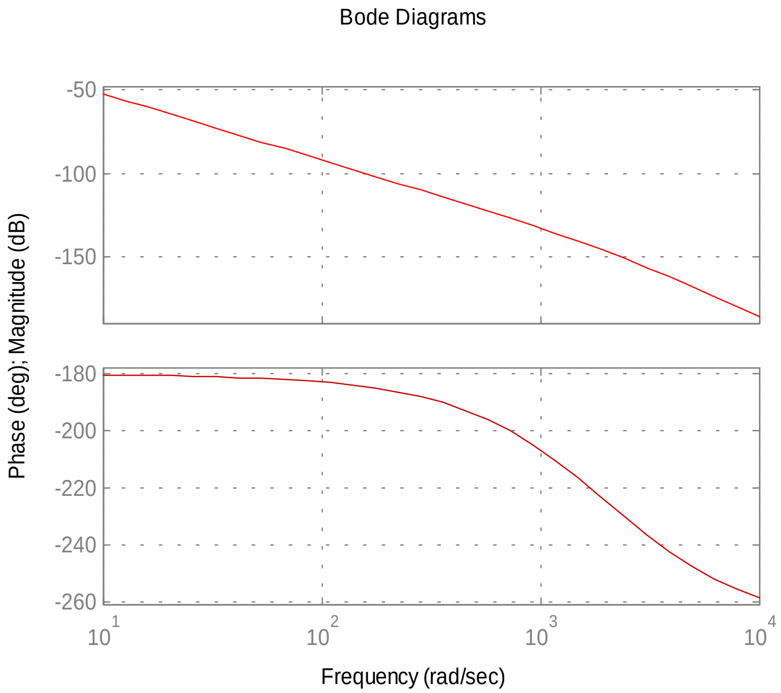
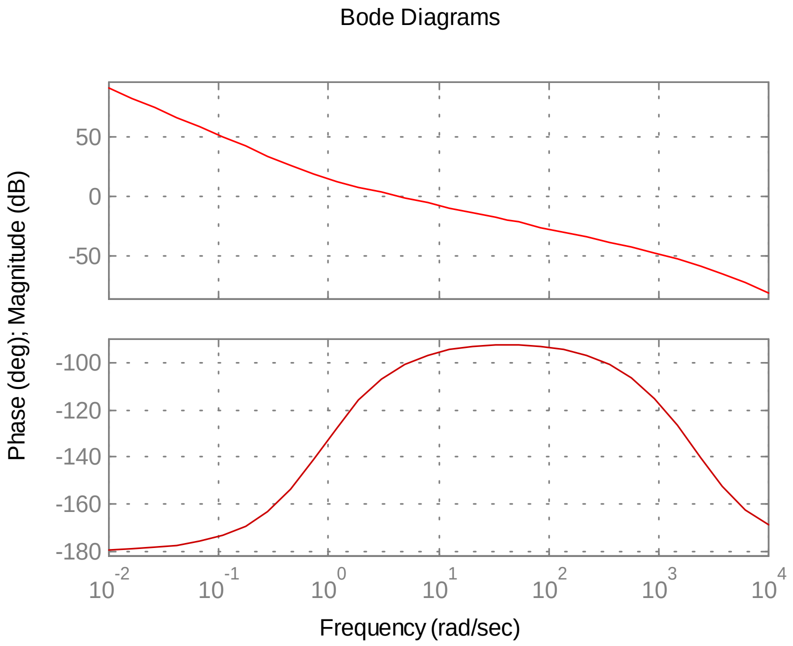


Figure 3.5 (a) Open loop system without



3.5(b) Bode plot of the system with PID

According to this graph, after the compensation, they system got

Phase margin Pm= 45.91°

while the gain margin Gm= 347.74 dB

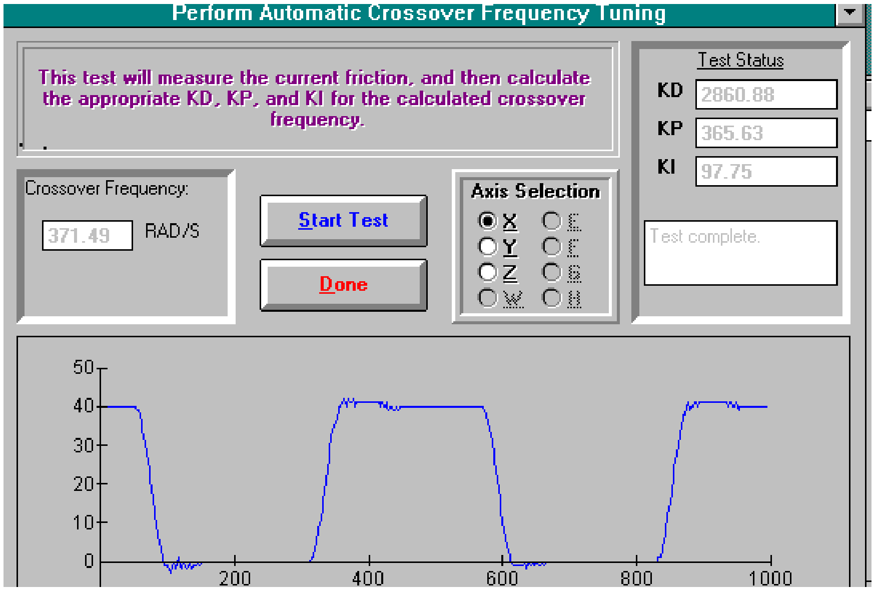
which meet the desired requirement.

3.11 The BEARCAT II system PID compensation:

In the actual system, it is hard to determine the road friction factor and other system parameters. So, it is hard to find PID value by theoretical calculation only. Fortunately, the Galil controller has a associated program called WSDK which has the ability to automatically find the PID value that is right for the real life control system.

This software allowed us to perform conservative tuning and valued for the PID controller that is suitable for the system. The Bode plot frequency response was measured by supplying a sinusoidal input signal to the open loop system and recording the response through the encoder. A phase margin of 40 degrees and a gain margin of 10 decibels was achieved. The step response was tested to minimize the overshoot and select a critically damped response. The torque for real life conditions were measured. Tuning the amplifier parameters especially loop gain and selection of the PID parameters were very important and required iterative adjustments. The tuning effects of these PID values would also be shown on the screen. One example of WSDK test is shown in Fig 3.6

The values get from the WSDK tuning would be further input into the master motion control program.



Perform Automatic Crossover Frequency Tuning

This test will measure the current friction, and then calculate the appropriate KD, KP, and KI for the calculated crossover frequency.

Test Status

KD 2860.88

KP 365.63

KI 97.75

Crossover Frequency: 371.49 RAD/S

Test complete.

Axis Selection: ● X ○ E ○ Y ○ F ○ Z ○ G ○ W ○ H

Figure 3.6 WSDK tuning example

Chapter 4. Motion control programming and test

4.1 Functional behavior Analysis

1. The wheel drive system

As constructed, the AGV is driven by two independent motors. These motors drive the left and right wheel respectively through gearboxes attached to them. An encoder is mounted on each motor shaft to feedback position signals. A castor wheel in the rear of the vehicle is free to swing when the vehicle has to negotiate a turn. The encoder position signal is numerically differentiated to provide velocity feedback signal. The variation of left and right motor speed will account for the turning motion of the mobile robot body.

2. The sonar rotating system.

The sonar rotating system is comprised of a small-sized motor and a small-capacity amplifier. It is independent from the wheel power system.

4.2 Motion behavior identification

In order to give the BEARCAT II ideal ability to handle its motion during the contest, the robot should have at least three motion status, i.e. turn left, turn right and stop. In addition to that, the robot should also have the ability to go back whenever it found no way to proceed. Also, one important feature of this design is its ability of making Zero Turning Radius (ZTR), which is gaining popularity in the U.S. mowing market. ZTR means the ability to turn around the center of the axis of the drive wheels. This design offers exceptional manuverbility with sharp turns. For the sonar system, the rotation of sonar would be limited to several positions at a limited range. In this design, the robot needs to sense an area of +/-30 degree in front of it. Five positions were selected as the sonar sensor reading.

All in all, the robot should have the following motion abilities:

1) Left turn
2) Right turn
3) ZTR_left
4) ZTR_right
5) Reverse
6) Sonar positioning

4.3 The Galil motion control programming

The motion controller plugs into the PC bus and accepts several high level ASCII commands and motion profiles. A terminal emulator program is provided enabling the designer to send commands from the PC keyboard. The Galil controller has its own programming language to control the motors moving. In general there are three positioning strategies to follow:

Independent Axis Positioning: In this mode, the desired absolute position (PA) or relative position (PR), along with the acceleration rate (AC), deceleration rate (DC), and slew speed (SP) are specified. Once begin, the DMC-1000 generates a trapezoidal or triangular velocity profile and position trajectory. The user specifies Position can be interrogated at any time using the Tell Position (TP) command.

Jogging: The jog mode allows the user to command each motor to run at a prescribed jog speed until a stop command is assigned. The user specifies the jog speed (JG), the acceleration rate (AC), and the deceleration rate (DC). On begin (BG), the motor accelerates up to the jog speed and continues at that speed until a new speed or stop command (ST) is issued. The direction of motion is specified by the sign of the JG parameter. The JG, AC, and DC parameters can be changed at any time during motion. Average speed can be interrogated at any time using the tell velocity (TV) command.

Linear Interpolation: The DMC-1000 provides a linear interpolation mode for any combination of 1, 2, 3, axes if the DMC-1080 is used. Here, motion between the axes is coordinated to maintain the prescribed vector speed (VS), acceleration rate (VA), and deceleration rate (VD) along the specified path.

4.4 Programming Mode Selection

During its operation, the wheel motor motion is a continuous process, **Jogging** mode was selected for this kind of motion. For the sonar motion, it requires the motor to move the motor to a specified position to sample obstacle distance, so the **independent axis-positioning** mode would be a good solution. Programs to achieve these kinds of motion behavior were coded in Turbo C++, which is listed in Appendix A, B, and C.

4.5 The motion behavior tests and results

Motion behaviors were programmed and tested the motion character of the final design in a variety of running conditions, including the test grounds at the University of Cincinnati and Oakland University. The observations of the motion systems performance are summarized below:

- Wheel motion control system:

The wheel motion system is the final computer command execution unit. The WSDK will help us obtain right PID parameters that achieve a real life stable motion response with very little overshoot. Under the manual control mode, the control system could follow the keyboard command to do all five motions as identified before. The maximum speed of the motion system is about 27 meters/minute. However, the go straight behavior may have right hand side errors, the rate is about 10.5 feet to 1 foot, i.e., for every 10.5 feet of straight line motion, the robot may run 1 foot off the desired track. The corresponding error angle is about $\tan^{-1}(1/10.5) = 5.48°$. This is because the two independent wheel systems have different unpredicted mechanical non-linear properties. Fortunately, this error could be corrected when combined with the vision system.

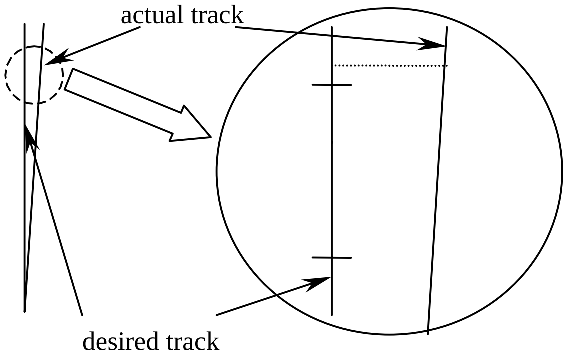


Figure 4.1 Wheel motion and off track error

- Sonar motion behavior

The sonar motor was programmed to move among 5 positions continuously. The resolution of this motion is limited by the encoder, which is 2000 counts/circle. So, the resolution of sonar motion is about 360/2000=0.18°. However, this accuracy is good enough for the contest environment.

- Safety and Emergency Stop System

For several times, the fuses built in the motor power circuits has burned themselves and protected the motor and amplifier. The over current is usually caused by the short cut on the metal frame. The manual E-stop is the most reliable device. Each time it is off, the motion system will surely be shut down. The remote device is highly capable, the maximum distance to function the remote E-stop is 65 feet. Because of its coded signal, the remote device was never mistakenly stimulated by environmental noise. The two batteries power supply can sustain the power system for 1 hours of continuous work.

Chapter 5 The AUVS competition and the results

5.1 The competition introduction

The 7th AUVS competition was hold at the scenic campus of Oakland University in Rochester Michigan. Fourteen Unmanned Ground Vehicles (UGVs) pulled up to the starting line for the sixth consecutive year. The purpose of this contest is to offers a design experience that is at the very cutting edge of engineering education. It is multidisciplinary, team implemented, theory-based, hands-on, outcome assessed, and is based on product realization. It encompasses the very latest technologies impacting industrial development and taps subjects of high interest to students.

There are four kinds of competitions involved: Vehicle performance competition, Vehicle Design Competition, Road Debris Bonus Event and Follow the Leader Bonus Event. Two of them will be discussed.

5.2 VEHICLE PERFORMANCE COMPETITION

- OBJECTIVE:

1. Robotic vehicle to autonomously navigate around an outdoor obstacle course under a prescribed time while avoiding the obstacles on the track.

2.Judges will rank the entries which complete the course based on shortest adjusted time taken. In the event that a vehicle does not finish the course, the judges will rank the entry based on longest adjusted distance traveled

Other requirements including:

1.Speed: For safety, a maximum vehicle speed of five miles per hour (5 mph) will be enforced. All vehicles must be governed not to exceed this maximum speed. No changes to maximum speed control software and hardware are allowed after qualification.

2.E-Stop: Each vehicle must be equipped with both a manual and a wireless (RF) remote emergency stop (E-Stop) capability. The wireless E-Stop must be effective for a minimum of 50 feet. The manual E-Stop must be easy to identify and activate. Activating the E-Stop must bring the vehicle to a quick and complete stop (stopping distance not to exceed 12ft on inclines up to 15%).

- BEARCAT II performance

In the first day, the BEARCAT II has the remote E-stop burnt. The next day, new remote E-stop unit was bought and in serial connection with the motor power control circuit, which is controlled by a solenoid. This effort helped the BEARCAT II to pass the qualification test. During the formal contest, the computer motion system closely followed the commands from the computer and the sonar rotation system the BEARCAT II made a distance of 153.3 feet during its second efforts, which is the fourth best among other teams. Figure 5.1 shows the track that the vehicle traveled during the contest.

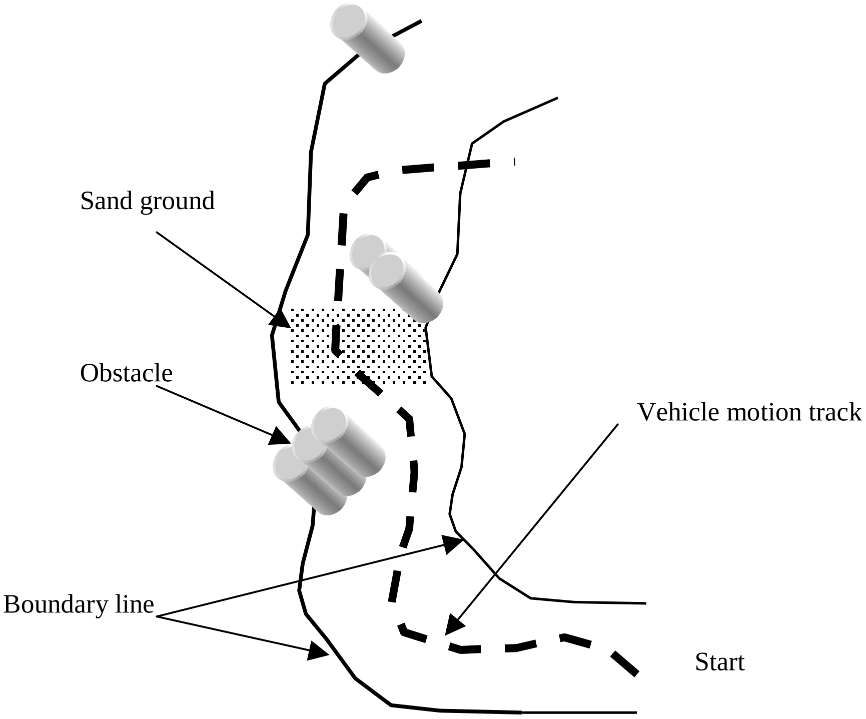


Figure 5.1 The Bearcat II in the vehicle

5.3 Road Debris Bonus Event

In this competition, the robot vehicle will encounter with irregular obstacles that could be seen during regular highway or roadway driving. This can include, construction signs, barrels and cones, hubcaps, tire treads, tailpipes, barricades, parked cars and pedestrians. Each autonomous vehicle system (AVS) entering the Bonus Competition Event (BCE) will be given 2 attempts at negotiating a 150'-250' stretch of road.

Again, Bearcat II moved on the complicated track for 65 feet before it runs out of track, which is the 3rd best performance in the Road Debrit Event. Both the wheel motion and the sonar rotation acted accurately to support the robot moving in the right direction.

5.4 Motion system performance during the contest

The robot was transported to Oakland University in Detroit to participate the AUVS competition in the early June of 1999. Extensive field tests were performed on the contest track. The motion system was tested on a variety of ground conditions including dry and wet grass, sand, plywood ramps and asphalt. The performance of motion systems is as following

1. It can successfully follow the computer command, demonstrated all five motion behaviors as defined before. Two ZTR were observed.
2. Both the remote E-stop and manual E-stop can stop the motion of the robot. The maximum effective distance for the remote E-stop is about 65 feet.
3. No run away situation occurred.
4. The overall quality of the motion system was proved by its performance during the contest. During the Vehicle Performance Competition, the robot successfully followed the track and avoided obstacles and also traveled over 155 feet. This score ranked 4th among 14 competitors. Also, during the ROAD DEBRIS competition, the robot moved 65 feet without breaking rules, which is the 3rd best performance.

Chapter 6 Conclusions

Motion control design is a critical factor in the modern AGV systems. It needs arduous pre-scheduling, cooperative design and great experiments. In this design, a digital motion control system for an AGV was configured, constructed and tested. The control systems was modeled, simulated and analyzed. The theory behind the motion control and PID compensation was outlined. After realization, the system demonstrated reliable motion control under the computer command. It is because of the high reliability and accuracy of the motion system that higher level controller could make effective judgement with the vision line tracking and ultrasonic information.

One problem of the motion system design is the castor mounted on the backside of the robot. It is too small to handle battery weight that it often trapped into the grass. One good solution is to reduce the number of battery from four to two and also move the battery in the middle of the robot instead of the four in the backside. In doing this, the robot get a balanced weight and thus make a stable motion. However, it would be better to mount four large wheels instead of one little castor before the robot can do gentle motion.

The field testing of the robot indicates that the motion system can successfully follow the computer command, and at the same time, functionally behaves as commanded. Also, the motion system can provide enough power for the vehicle to move across sands, ramp and other complicated situation. The safety system helped to reduce possibility of system crash down, which happened several times before. The ranking of BEARCAT II in the list of the AUVS competition proved that the motion control system of BEARCAT II is a successful design.

The future design of the system may incorporate two large wheels on the backside of the robot instead of one small castor. This may reduce the change of misdirection when the front motor is powered up. Velocity feedback could also be applied to the system so that the motion system could get a better stable reaction. Higher level control strategies could be developed. Other eminent features of Galil controller, such as contour mode, could be utilized to give BEARCAT II better capacity. This feature is particularly useful to apply the model-based algorithm, which is another popular approach in today's AGV navigation. Higher level control strategies need to be done to show the effects of new the contour feature of the Galil controller could be explored to apply the model based algorithm, which is another popular approach in today's AGV navigation.

Appendix A. Assembly code to achieve three position sonar motion

```
//  The sonar motor encoder has a rate of 2000/circle
DP ,,0 //define current position as 0
SP,,35000 // initialize speed
AC,, 35000 //initialize acceleration
SH //serveo is ready
PA,,166 // move 30 degree
AMZ // non stop until the movement finished
BGZ // start the motion
PA,,0 // move back to original position
AMZ
BGZ
PA,,-166 //move to –30 degree
AMZ
BGZ
PA,,0  // move back to the original position
AMZ
BGZ
GOTO  4 // execute beginning at the 4th line
EN //program end
```

Appendix B. C code to achieve 5 position sonar motion

```
Const int Enc_counts 2000; // the encoder is 2000 counts/revolution

void initiSonar()
{
//this function, when invoked, will initialize the settings for sonar motion
            submitDMC("AC ,,35000");
            submitDMC("DC ,,35000");
            submitDMC("DP ,,0");
            submitDMC("SH");
}

void positionSonar(int sonarAngle)
{
       //This function when invoked will rotate sonar to the desired angle position
      //the angle value should be in degree

       char position[10],send[15];
            int Angle;
       Angle=Enc_counts/360*sonarAngle; //Change angle value into the actual encoder
                                                                  //counts that need to be rotated.
       gcvt(Angle,4,position);
       char *command="PA,,";
       strcpy(send,command);
       strcat(send,position);
       submitDMC(send);
       cout<<send<<"\n";
       submitDMC("BGZ");
       submitDMC("AMZ");
 }

void rotateSonar()
{
//This function, when invoked, will rotate sonar to five positions and read obstacle  //information in those
directions
   int i;
   int posSonar;
  float obstacleDistance;
   initSonar();
   for ( i=-2;i<=2;i++)
            {
            posSonar=15*i;   //rotate sonar in five positions -30,-15,0,15,30
            positionSonar(posSonar);
            sleep(250);                       //wait 0.25 seconds for the motion to be completed
          obstacleDistance=sonl();      // get the sonar reading
            sleep(250);                        //wait for another 0.25 seconds for the reading complete
            }
}
```

Appendix C Programs to achieve Basic Car Behavior

```
   /********************************/
 /****** basic car behavior **********/
/********************************/

long int spdx, spdy;

void stopCAR()
            {
            submitDMC("ST XY");
            submitDMC("MO XY");
            submitDMC("SH XY");
            cout << " Stopped Running\n";
            }

void stopSonar()
            {
            submitDMC("ST Z");
            submitDMC("MO Z");
            submitDMC("SH Z");
            cout << " Sonar rotation stopped\n";
            }


void speedCARx(long int spx)
{
            //This command, when invoked, will set the x axis speed
            //of bearcat II.It is called with a value between 0 - 360000.

            char inx[10],sendx[15];
            gcvt(spx,6,inx);
            char *commandx="JG";
            strcpy(sendx,commandx);
            strcat(sendx,inx);
            cout<< " Left-motor:                \n";
            cout << sendx;
            cout << "X-motor -->     ";

            submitDMC(sendx);
            submitDMC("BGX");
}

void speedCARy(long int spy){
            //This command when invoked will set the speed of bearcat II.
            //It is called with a value between 0 - 250000.

            char iny[10],sendy[15];
            gcvt(spy,6,iny);
            char *commandy="JG,";
            strcpy(sendy,commandy);
            strcat(sendy,iny);
```

```
        gotoxy(28,21);
        cout<<"Right-Motor:                 ";
        gotoxy(42,21);
        cout << sendy;
        gotoxy(1,24);
        cout<< "Y-motor -->     ";
        submitDMC(sendy);
        submitDMC("BGY");
}

void steerCAR(float val)
{       double dtime;
         time_t second=time(NULL);
         gotoxy (25,12);
         cout << "                 ";
         gotoxy(20,16);
         cout << "Inside steercar:   -->VAL: " << val << "\n";
        //This function when invoked will put the steering wheel to the         //absolute angle given.
        This angle ranges between +-20.
        dtime=difftime(second,first);
        cout << "Time data:    first = "<< first << "     second = "<< second;
        cout << "     dtime = " << dtime  << endl;
        first=second;
        dtime = 0.25;

        if (val < 5 && val > -5)
        {
                spdx = 18000;
                spdy = 18000;
        }
        else
        /*if (val >= 5 && val <= 30 || val <= -5 && val>=-30)*/
        {
                spdx=18000-((134.5*val)/1);
                spdy=18000+((134.5*val)/1);
        }

        if (spdx > 36000)spdx =36000;
        if (spdx < -36000)spdx =-36000;
        if (spdy > 36000)spdy =36000;
        if (spdy < -36000)spdy =-36000;


        cout << "\t\tspdx = " << spdx  << "   ";
        cout << "spdy = " << spdy << "   ";
        cout << "increment = " << (int)((134.5*val)/dtime) << " ";
        speedCARx(spdx);
        speedCARy(spdy);

}

leftZTR()
{
        cout << "ZTR Left Side \n\t";
        if(spdx+spdy==0)
        {
          spdx-=STEER_INC;
          spdy+=STEER_INC;
          speedCARx(spdx);
          speedCARy(spdy);
         }
         else
        {
          cout << "Stopping Car  \n\t";
          spdx= -1*STEER_INC;
          spdy=  1*STEER_INC;
          speedCARx(spdx);
          speedCARy(spdy);
         }

}

rightZTR()
{
        cout << "ZTR Right Side\n\t";
        if(spdx+spdy==0)
         {
         spdx+=STEER_INC;
         spdy-=STEER_INC;
         speedCARx(spdx);
         speedCARy(spdy);
         }
         else
         {
          spdx=  1*STEER_INC;
          spdy= -1*STEER_INC;
          speedCARx(spdx);
          speedCARy(spdy);
         }

}
```