R/C SERVO-MOTORS – A TUTORIAL

1. INTRODUCTION

This tutorial is intended to provide the information necessary to understand the basic operation of a radio control or hobby type Servo motor, hereafter referred to for short as a "Servo".
Due to their affordability, reliability, and simplicity of control by microprocessors, R/C Servo's are often used in small-scale robotics applications.

2. WHAT IS A SERVO

A Servo is a small device with an internal DC motor, internal gearing and an internal feedback control loop circuitry. The Servo has an electrical input signal and an output shaft.

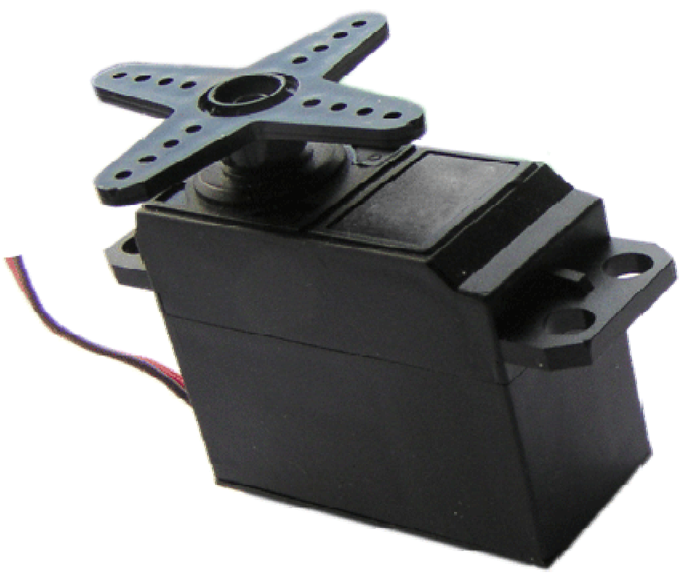
This shaft can be positioned to specific angular positions by sending the Servo a coded signal. As long as the coded signal exists on the input line, the Servo will maintain the angular position of the shaft. As the coded signal changes, the angular position of the shaft changes. In practice, Servos are used in radio controlled airplanes to position control surfaces like the elevators and rudders. They are also used in radio controlled cars, boats, planes, tanks, puppets, and of course, robots. These R/C Servos are sold at hobby stores and via E-bay for anywhere from $10 to $150.

R/C Servos come in standard "sizes" so that they have a more universal fit into models for replacement or upgrade and use similar control schemes. Unlike general purpose motors, R/C Servos are constrained from full rotation. Instead, as standard, they have a limited rotation such as 90, 120 or 180 degrees.

A typical Servo is used to control an angular motion of between 0 and 180 degrees. The Servo is mechanically not capable of turning any farther due to a mechanical stop built on to the main output gear.

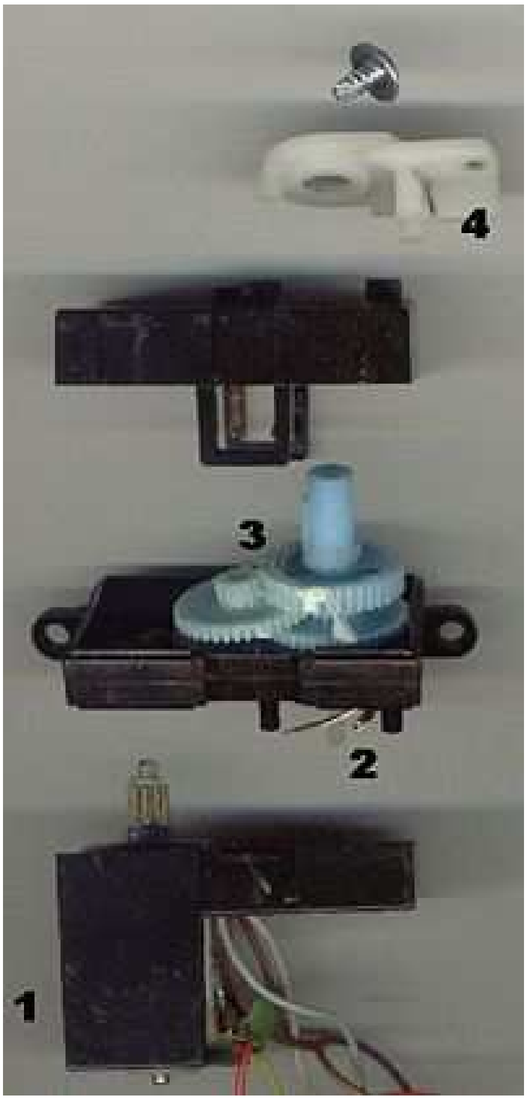
A typical R/C Servo looks like a rectangular box with three wires coming out of one end of the Servo body to a connector and a shaft coming out at the top or other end.

Attached to the Servo shaft is usually (but not always) a mechanical lever or disk which is referred to as a "horn". This is a plastic piece with holes in it for attaching push rods or other mechanical linkages to the Servo.



3. R/C SERVO CONSTRUCTION

Servos are constructed from four basic pieces, a motor, a feedback device, a gear train and a control board.



A small R/C Servo mechanism
1. electric motor
2. position feedback potentiometer
3. reduction gear
4. actuator arm (or “horn”)

In R/C Servos the feedback device is typically a potentiometer (variable resistor). The motor, through the gear train, turns the output shaft and the potentiometer simultaneously. The potentiometer is fed into the Servo control circuit and when the control circuit detects that the position is correct, it stops the motor.
The electronics are pretty much all the same and so not an issue. In the motor department however you can get smaller and larger motors which dictates the overall size of the Servo. "mini" Servos are generally more expensive than "standard" Servos in part for this reason.

The potentiometer is the feedback device and often the first thing to fail. If it gets dirty, or the contacts get oxidized, the Servo will fail to work properly, sometimes by "jittering or hunting" since the feedback is inaccurate, or turning completely to one side and drawing lots of current since the Servo doesn't know where its output shaft is pointing. More expensive Servos have "sealed" potentiometers, cheaper ones do not. The life of cheaper Servo potentiometers can sometimes be extended a bit by some judicious application of silicone sealant around the edge. You can do this with a syringe if you are careful. Be sure and not to get it on the gears though as it will cause them to bind.

The gears also vary from Servo to Servo. Inexpensive Servos have plastic gears that will wear out after less than 100 hours of use. More expensive Servos have metal gears which are much more durable.

Nylon Gears - Nylon gears are most common in Servos. They are extremely smooth with little or no wear factors. They are also very lightweight, but lack in durability and strength.

Karbonite Gears - Karbonite gears are relatively new to the market. They offer approximately 5 times the strength of nylon gears and also better wear resistance. Cycle times of well over 300,000 have been observed with these gears with virtually no wear. Servos with these gears are more expensive but what you get in durability is more than equalled.

Metal Gears - Metal gears have been around for sometime now. Although the heaviest and having the highest wear rate of all gear types, they offer unparalleled strength. With a metal output shaft, side-loads can be much greater. Ever had a nylon output shaft crack? I have. In applications that are jarred around, metal gears are best. Unfortunately, due to wear, metal gears will eventually develop slight play in the gear-train. Accuracy will slowly be lost.

The last feature is the output shaft bearing. Cheap Servos invariably have a plastic sleeve bearings that will not take much load. Medium priced Servos generally have metal sleeve bearings that stand up better under extended use and the expensive Servos have ball bearings which work best. Many places also sell "ball bearing upgrades" for cheap Servos which consist of a new top cover and ball based bearing for the output shaft.

The last major difference is in the **spline**. The spline is the output shaft of the Servo. This is where you would attach your Servo horn or Servo arm. Standard Hitec splines have **24** teeth while standard Futaba splines have **25** teeth. What makes this important is that Servo horns built for one will not work with the other.

4. WIRING CONNECTIONS

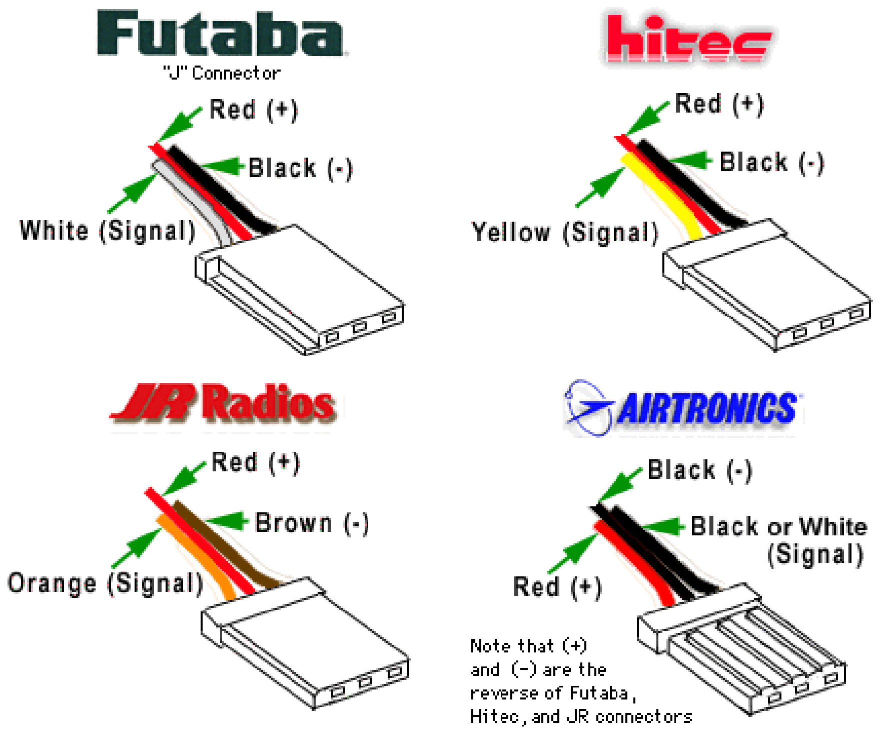
The three wires are:

- Ground (**black** or **brown**),
- Power (**red** in colour), and
- Control (**white**, **yellow**, **orange** or other colour).

Unfortunately, this colour scheme is not consistent for all Servos, so it is worth checking the datasheet for the Servo purchased.

In most cases, the wiring order is the same, just different colours. Note that Airtonics and compatibles have the reverse order.

Below are the four standard connectors that are used by the 'big' four radio manufacturers (other manufacturers or third-party Servo makers, such as FMA Direct or Cirrus, use the Hitec standard):



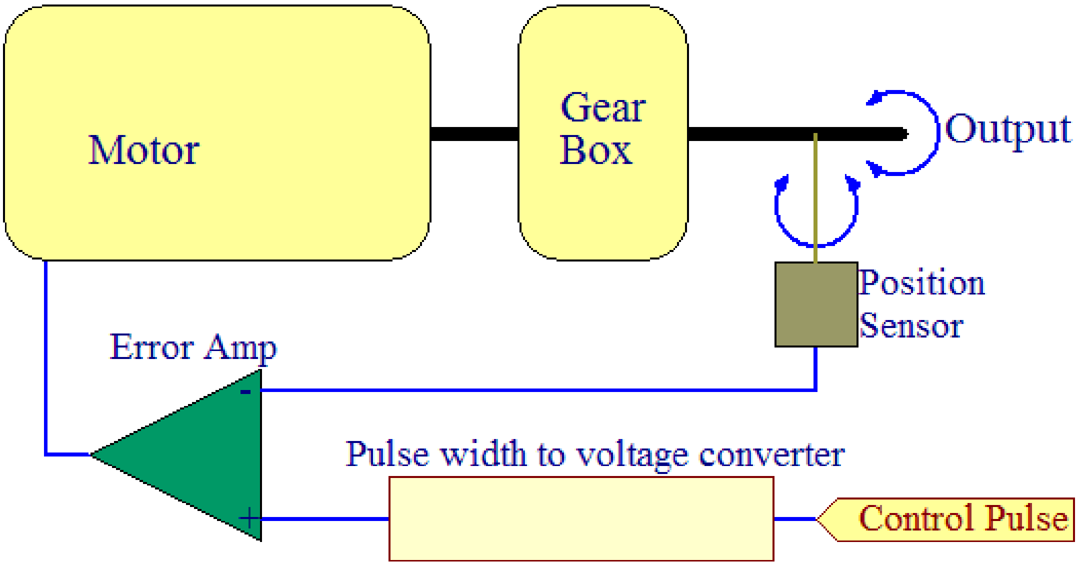
Note that the Futaba (J type) has a special flange while Hitec (S type, for universal) does not. Futaba has the extra flange to help the user plug in the Servo correctly, although there are only two ways to do it and connecting a Servo in the wrong way will not actually damage anything. If you want to connect a Futaba Servo to a Hitec device, just clip the flange off and use sandpaper to file it down until it fits. If you ever need to connect a Hitec connector to something Futaba, just use sandpaper to decrease the connector width until it fits.

You **cannot** reverse the direction of a Servo simply by swapping (+) and (-). If you do, you'll burn out the Servo internal controller board. If you're **really** good at soldering very small wires, you can reverse the normal direction of Servo by swapping the wires that connect directly to the Servo motor inside the Servo case **as well as** the little Servo wiper that moves as the Servo moves.

As of December 1997, Airtronics is now shipping Servos with an optional "Z" connector that matches the connector found on JR Radio or Hitec Servos. This means that if you have a Airtronics Servo with one of the "Z" connectors, you won't have to switch (+) and (-) wires to use that Servo with Futaba, Hitec, or JR Radio receivers.

5. R/C SERVO's - FUNDAMENTALS OF OPERATION

The diagram below contains a block diagram of a typical Servo motor unit.
The radio control receiver or microcontroller generates a Control Pulse of varying length approximately every 20 milliseconds. The length of the pulse is used by the Servo control board to determine the position it should rotate to.



Starting from the control pulse we will work though each part of the diagram and explain how it all fits together. Once we have gone through how the Servo works we will investigate how the control pulses can be generated with a PICAXE microcontroller.

5.1 Pulse width to voltage converter

The control pulse is feed to a pulse width to voltage converter. This circuit charges a capacitor at a constant rate while the pulse is high. When the control pulse goes low the charge on the capacitor is fed to the output via a suitable buffer amplifier. This essentially produces a voltage related to the length of the applied pulse.

The circuit is tuned to produce a useful voltage from a pulse with duration of approximately 1ms to 2ms. The output voltage is buffered and so does not decay significantly between control pulses so the length of time between pulses is not critical, within reason (more on that later).

5.2 Position Sensor

The actual rotational position of the Servo output shaft is read by a sensor. This is normally a potentiometer (variable resistor) which produces a voltage that is related to the absolute angle of the output shaft.

The position sensor then feeds this proportional voltage for the Servo shaft position into the Error Amplifier which compares the actual shaft position with the commanded position supplied by the pulse width to voltage converter.

5.3 Error Amplifier

The error amplifier is an operational amplifier with negative feedback. It will always try to minimise the difference between the inverting (negative) and non-inverting (positive) inputs by driving its output is the correct direction.

The output of the error amplifier is either a negative or positive voltage representing the difference between its inputs. The greater the difference between the input voltages, the greater the output voltage from the error amplifier to the motor.

The error amplifier output is used to drive the motor; If it is positive the motor will turn in one direction, if negative the other. This allows the error amplifier to reduce the difference between its inputs (thus closing the negative feedback loop) and so make the Servo go to the commanded position.

The Servo normally contains a single integrated circuit and a hand full of discreet components to implement the entire control system.

5.4 Motor and Gearset

As mentioned above, the motor rotates in a direction set by the polarity of the output voltage from the Error Amplifier. As the output shaft approaches the desired position, the error reduces and the error amplifier voltage changes so that the motor slows down as it nears the desired position.

Thus the voltage and power applied to the motor is proportional to the distance it needs to travel. So, if the shaft needs to turn a large distance, the motor will run at full speed. If it needs to turn only a small amount, the motor will run at a slower speed.

The gearset reduces the motor shaft speed by a significant factor (of the order of 100:1 ratio) so that the Servo output shaft rotates relatively slowly (compared to the motor shaft).

The gearset in reducing the Servo output shaft speed, also increases the torque available at the output shaft. The maximum amount of force the Servo can exert is the torque rating of the Servo.
For example, Jaycar sell standard sized Servos with torque ratings of 11kg/cm at 6 Volts (YM2765) and 13kg/cm at 6 Volts (YM2763). 13kg/cm equates to a force of 13 kg at 1cm (or 180 oz at 1 inch) from the shaft centreline.

A Servo draws power proportional to the mechanical load. A lightly loaded Servo, therefore, doesn't consume much energy.

6. SERVO CONTROL SIGNAL CHARACTERISTICS

R/C Servos are controlled by sending them a square-wave "pulse" of variable width and approximately 3-5 Volts on the control wire. The parameters for this pulse are that it has a minimum width, a maximum width, and a repetition rate. These values are not "standard" but there are conventions that are generally accepted. The convention is that a pulse of approximately 1.5 ms is the "neutral" point for the Servo. Given the rotation constraints of the Servo, neutral is defined to be the position where the Servo has exactly the same amount of potential rotation in the counter clockwise direction as it does in the clockwise direction.

It is important to note that different R/C Servos will have different constraints on their rotation but they all have a neutral position, and that position is always around 1.5 ms.

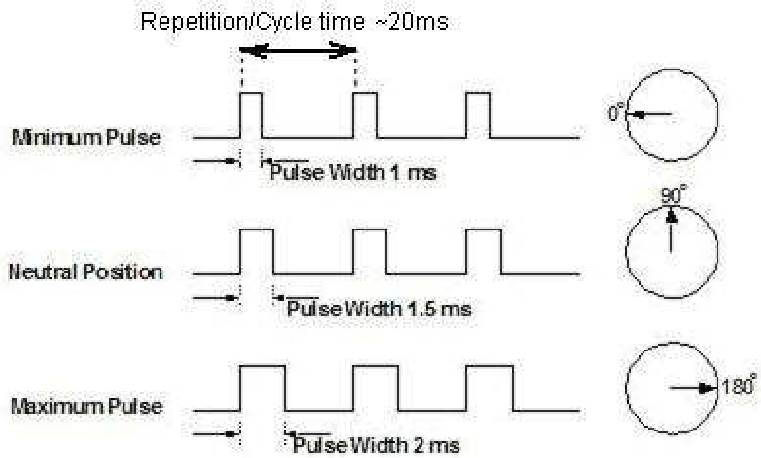
R/C Servos are "active" devices, meaning that when commanded to move they will actively hold their position. Thus, if a Servo is commanded to the neutral position and an external force is present to push against the Servo (presumably through the mechanical linkage) the Servo will actively resist being moved out of that position.

Servos will not hold their position forever though, the position pulse must be repeated to instruct the Servo to stay in position. The maximum amount of time that can pass before the Servo will stop holding its position is the command repetition rate. The command repetition rate can be in the range 10 - 30 mS and most control systems typically use a 20ms period. You can repeat the pulse more often than this, but not less often. When this timeout expires and there hasn't been another pulse the Servo de-energizes the motor. In this state in can be pushed out of position and it will not return to the commanded position.

When the pulse sent to a Servo is less than 1.5 ms, the Servo positions and holds its output shaft some number of degrees counter-clockwise from the neutral point. When the pulse is wider than 1.5 ms the Servo positions and holds its output shaft some number of degrees clockwise from the neutral point. The minimal width and the maximum width of pulse that will command the Servo to turn to a valid position are functions of each Servo. Different brands, and even different Servos of the same brand, will have different maximum and minimum durations. Generally the minimum pulse will be about 1.0 ms wide and the maximum pulse will be 2.0 ms duration. However, these are just guidelines and should be checked on the Servos you use. In particular if you attempt to command a Servo past its maximum or minimum rotation it will use the maximum amount of current trying unsuccessfully to achieve that position.

From the above we can determine that we need to generate a pulse approximately every 20ms although the actual time between pulses is not too critical. The pulse width however must be accurate to ensure that we can accurately set the position of the Servo.

The typical **time vs. angle** is represented in this diagram:



Note that the times here are illustrative and the actual timings depend on the Servo manufacturer. The principle, however, is the same.

Another parameter that varies from Servo to Servo is the slew rate. This is the time it takes for the Servo to change from one position to another. The worst case slewing time is when the Servo is holding at the minimum rotation and it is commanded to go to maximum rotation. This can take several seconds on very high torque Servos. Typically it takes less than two seconds.

7. CONTROL WITH A PICAXE CONTROLLER

Firstly, as stated in the PICAXE manuals:

Servo’s require a large current (up to 1A) and also introduce a large amount of electrical noise on to the power rail. Therefore in most cases the Servo should be powered from a separate power supply. Remember that when using two power supplies the two 0V rails must be joined to provide a common reference point.

7.1 Using the SERVO and SERVOPOS Commands

The PICAXE has two commands specifically intended for Servo control. These are the best commands to use.

PICAXE BASIC has the SERVO command to establish initial control of a Servo.
The command syntax is: **SERVO pin, pulse**

Where **pin** is an output pin from 0 to 7 and **pulse** is the duration in tens of microseconds.
Divide the **pulse** value by 100 to get milliseconds –(eg 150 = 1.5 ms)

Once the Servo has been initialized, the SERVOPOS command is used to further change the position without re-initialising internal PICAXE timer which could cause "jitter" of the Servo shaft.
The command syntax is: **SERVOPOS pin, pulse**

Note that these commands are different to most other BASIC commands in that the pulsing mode **continues** until another SERVO, HIGH or LOW command is executed. HIGH and LOW commands stop the pulsing immediately.

An example program to initially centre the Servo and then move it to the extremities, before returning to the centre position, is as follows:

```
Init: SERVO 1,150        ; servo on output 1 to centre/neutral position

Main:
      SERVOPOS 1, 75     ; move servo to one end
      PAUSE 5000         ; wait 5 seconds
      SERVOPOS 1, 100    ; move Servo toward anticlockwise limit - can try lower values like 75
      PAUSE 5000         ; wait 5 seconds
      SERVOPOS 1, 150    ; return the servo to centre/neutral position
      PAUSE 5000         ; wait 5 seconds
      SERVOPOS 1, 200    ; move Servo toward the clockwise limit - can try lower values like 225
      PAUSE 5000         ; wait 5 seconds
      SERVOPOS 1, 150    ; return the servo to centre/neutral position
      PAUSE 5000         ; wait 5 seconds
      GOTO Main          ; loop back to the Main label
```

7.2 Using other PICAXE Commands

Many microcontrollers including many PICAXE chips are equipped with PWM generators and some people initially consider using these to generate the control signals. Unfortunately they are not really suitable.

The problem is that we need a relatively accurate short pulse then a long delay; and generally you only have one PWM generator to share between several Servos which would require switching components outside the microcontroller and complicate the hardware.

The PWM generator is designed to generate an accurate pulse between 0% and 100% duty cycle, but we need something in the order of 5% to 10% duty cycle (1ms/20ms to 2ms/20ms). If a typical PWM generator is 8 or 10 bits say, then we can only use a small fraction of the bits to generate the pulse width we need and so we loose a lot of accuracy.

For the Low frequencies required for the Servo control, that is 50Hz, the values needed are outside the range of the PWMOUT command – try looking for 5% duty cycle at 50Hz in the PWM wizard.

Another PICAXE chip option is the **PULSOUT** command. The command syntax is:

PULSOUT pin, time

Where **pin** is an output pin from 0 to 7 and **time** is the duration in 10microsecond increment.

So for 1.0 ms duration the time value is 100, for 1.5ms duration the value is 150 and for 2.0 ms the value is 200.

In this case, the pulse is only sent once per execution of the command and not repetitively. So a routine with a loop is needed. For example:

```
Main:
     FOR b0 = 0 to 255       ; repeat 255 time  - need a word variable for up to 65535 times
          PULSOUT 1, 150
          PAUSE 19
     NEXT b0
```

Even this will only continue the command for approximately 255 * 0.020 = 5.1 seconds with a byte variable.

While other commands can be undertaken in place of the 19 ms pause, it is limiting in what can be done seemingly in parallel.

From this, it should be clear that if ongoing position control is required while doing other activities, then the **SERVO** and **SERVOPOS** commands are the best method for most circumstances.

8.0 OTHER SERVO FACTORS TO CONSIDER

8.1 Servo Current

Servo current operates the same as in a DC motor, except that you now also have a hard to predict feedback control system to contend with. If your DC motor is not at the specified angle, it will suddenly draw huge amounts of current to reach that angle. But there are other peculiarities as well. If you run an experiment with a Servo at a fixed angle and hang precision weights from the Servo horn, the **measured current** will not be what you expect. One would think that the current would increase at some fixed rate as the weights increased linearly. Instead you will get unpredictable curves and multiple rates.
In conclusion, Servo current draw is very unpredictable.

8.2 Stall Torque, Stall Current,

Since Servos contain DC motors, when the motor first starts to turn it is effectively in a stall condition. This causes a very high current to flow until the motor starts to move. Likewise, when the Servo output shaft is at a minimum or maximum position limit the motor cannot turn when commended to move further (past the limit) and will draw a very high current when attempting to do so.

If the load on the output shaft is greater than the Servo torque rating, the motor will be unable to move the shaft. Again a stall condition applies.

8.3 Velocity

The Servo turn rate, or transit time, is used for determining Servo rotational velocity. This is the amount of time it takes for the Servo to move a set amount, usually 60 degrees. For example, suppose you have a Servo with a transit time of 0.17sec/60 degrees at no load. This means it would take nearly half a second to rotate an entire 180 degrees. More time is taken if the Servo is under a load. This information is very important if high Servo response speed is a requirement of your robot application. It is also useful for determining the maximum forward velocity of your robot if your Servo is modified for full rotation. Remember, the worst case turning time is when the Servo is at the minimum rotation angle and is then commanded to go to maximum rotation angle, all while under load. This can take several seconds on a very high torque Servo.

8.4 Efficiency and Noise

Due to noise and control circuitry requirements, Servos are less efficient than uncontrolled DC motors. To begin with, the control circuitry typically drains 5-8mA just on idle. Secondly, noise can more than triple current draw during a holding position (not moving), and almost double current during rotation.

Noise is often a major source of Servo inefficiency and therefore should be avoided. Ever notice your Servo **jitter** or vibrate? This is because your Servo is rapidly jumping between two different angles due to interference. What causes this interference? Well the control wire is effectively an short antennae, capable of accepting unwanted foreign signals and sending them straight to your Servo as a command. A common interference source is usually from other nearby Servos and/or Servo wiring. To prevent this problem, keep your control wire short, meaning do not add say 1 metre of extension cables to your Servo. If you have many Servo wires going through one area, and it isn't feasible to keep them apart, then try twisting them together. You can also buy something called a Servo **booster extension** which buffers and amplifies the signal.

8.5 Regulating Voltage to a Servo

As you should already know, Servos have a voltage rating. Go above that voltage and your Servo overheats and may burn out. So suppose you have a 7.2V battery and you want to use a 5V regulator to power your Servos, is that a good idea?

Short answer: **No!**

Longer answer . . . it will work, but it is a huge waste of battery power.

Lets say you have your 7.2V regulated to 5V and the Servos draw a total of 1.5A of current.

Wasted power is:
(7.2V-5V)*1.5A = <u>3.3W</u>

Percentage wise, its
(7.2V-5V)/7.2V = <u>30.6%</u>

That’s the battery energy percentage wasted to thermal heat - almost 1/3rd!!!

Speaking of heat, your voltage regulator probably has thermal shutdown, meaning that if it overheats it will cut out until it cools down or at least limit the current to your Servos - meaning lower torque and lower speed. If your voltage regulator doesn't have thermal shutdown, it will just fry instead (not a good thing).

But if you still really really need to regulate for Servos, get a switching regulator (like ~83% efficiency on average).

9.0 ANALOGUE VERSUS DIGITAL SERVO’s

Digital Servos, from the user perspective, are controlled no differently than analogue Servos. The difference is in how the Servo motor is controlled via the circuit board (amplifier).

The motor of an analog Servo receives a signal from the Servo amplifier (inside the Servo) at 30 times a second. This pulsing signal tells the Servo motor when to start rotating and which way to rotate. Since it only happens 30 times a second, that's the minimum reaction time. Digital Servos use a higher frequency amplifier that sends a signal to the Servo motor 300 times a second. Since this signal is received by a digital Servo motor more often, it is able to react much faster and hold its position better. This means better centering and considerably higher holding power. The quick refresh also allows the digital Servo to have a tighter deadband. This power comes at a cost, however, as digital Servos tend to draw a lot more power from the on-board battery which means your battery just won't last as long. The digital micro processor is **10 times faster** than an analogue Servo. This results in a much quicker response from the beginning with the Servo developing all the rated torque 1 degree off of the center point. With the exception of a higher **cost**, there are only advantages for digital Servos over analogue Servos.

Digital Servos **can be programmed** for direction of rotation, center and end points, failsafe option, speed, and dead bandwidth adjustment. This is great for matching sets of Servos for deadband width, center and end points in giant scale aircraft applications, and for reversing a digital Servo when two are used on a "Y" harness.

If you do not want to deal with the added complication of programming, no worries! Hitec digital Servos will perform like standard Servos out of the box. It is not required to program them before use.

The standing torque of a digital Servo is 3 times that of its analog counterpart. This means digital Servos are typically smaller and have more torque.

10. SERVO MODIFICATION FOR CONTINUOUS ROTATION

When used with robots, R/C Servos can be employed as sensor pointers, leg lifters, steering wheel turners, etc. But without modification they can not be the main drive system. Since a Servo is, at its heart, a DC gear motor with enough torque to move a small platform, Servos are often modified to become drive motors.

Modifying a Servo to be a drive motor can use one of two strategies, breaking the feedback loop, or lobotomy.

The most brutal way of modifying a Servo is the full lobotomy. You open up a Servo, remove the electronics, bringing out the power lines to the motor and remove the potentiometer or modify it so that it can rotate 360 degrees. What you are left with is a DC motor, a gear train, and an output shaft on which you can mount plastic pieces that can be used as wheel mounts. This gives you complete control of the mechanics, but you do have to have a motor driver circuit to drive the DC motor in the Servo housing.

Breaking the feedback loop is generally the easier way to modify a Servo since it takes advantage of the power switching circuit already present on the Servo to turn the motor on and off. This modification involves removing/disabling the potentiometer and replacing it with a voltage divider that convinces the Servo electronics that the Servo is in the neutral position. (You can figure this out by using the old pot, turned to the neutral position and measuring the resistance.) Now to turn the motor clockwise you send the Servo a pulse that is wider than 1.5 ms and the motor turns (and never stops because there is no potentiometer to tell the Servo circuit it has gone far enough). Or to turn the motor counter- clockwise you send it a pulse less than 1.5 ms wide.

This latter technique is fine except that the motor driver circuit in the Servo may not be able to handle driving the motor continuously. In normal operation, the motor would be driven for a moment and then idled when the Servo reached its position. The intermittent nature of turning the motor on and off allows the Servo to use a motor driver that is smaller than one that would be needed for 100% duty cycle operation. If this turns out to be the case, the motor electronics will eventually burn out and you'll end up with the full lobotomy case by default.