ScienceDirect

Servo Motor

Related terms:

Grippers, Ultrasonic Sensor, Microcontrollers, Drones, Fault Diagnosis, Control Signal, Rotation Sensor

Controlling Motors

In Building Robots with LEGO Mindstorms NXT, 2007

Internals of NXT Servo Motor

Servo motors in industrial applications are different from regular motors because of their capability to precisely rotate the motor shaft. This is achieved by special electronics built into the motors. Similarly, the NXT servo motors are advanced in their capabilities and precision. Philippe Hurbain's Web site, NXT motor internals, is an excellent place to learn more about the internals of these motors (refer to Appendix A), and some of his material is included in the following section.

NXT servo motors have a built-in optical encoder that keeps count of rotations of the motor shaft (see Figure 3.2). This encoder is accurate up to 1 degree of motor rotation. You can use this property from your program for precise movement or positioning:

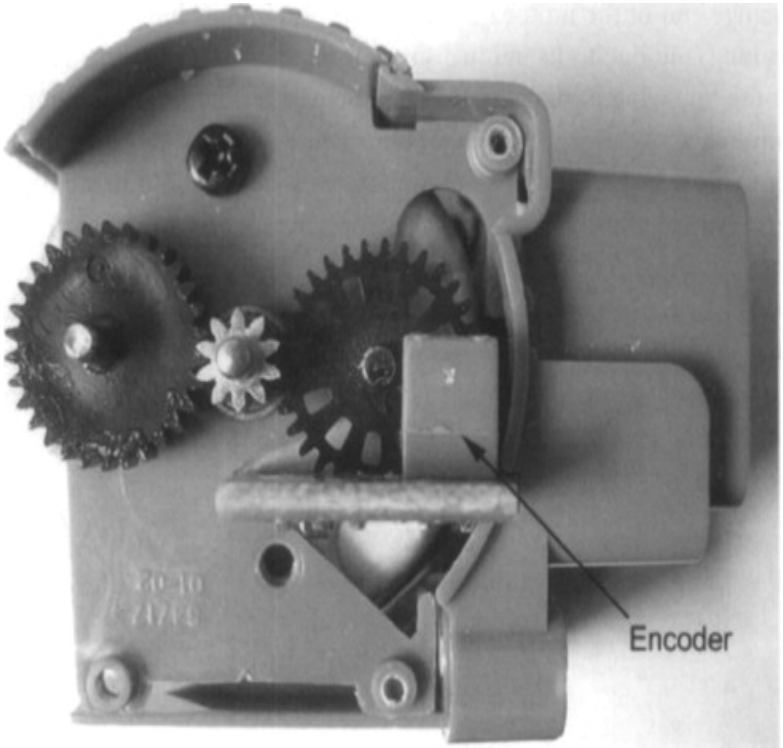


Figure 3.2. Optical Encoder in NXT Servo Motor

```
while (nMotorEncoder[motorA] < 1000)
// wait for motor to reach a specific location
{
. . .
}
```

This property can also keep two motors synchronized with each other and move your robot along a straight line.

The NXT servo motor also has built-in gears to reduce the rpms and increase the torque (see Figure 3.3). This desirable feature makes it easier to build robots without excessive geartrains, thereby reducing the complexity and size of your robot.

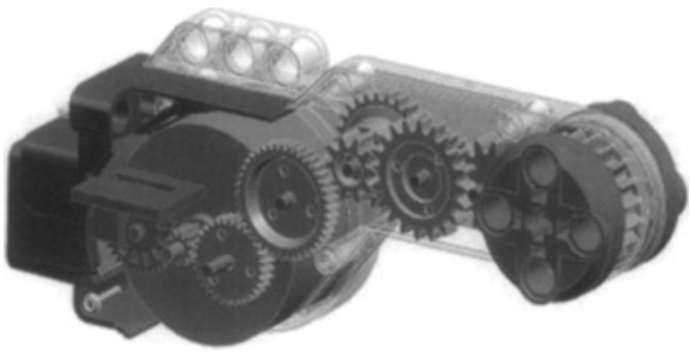


Figure 3.3. Internal Gears of NXT Motor

Read full chapter

URL: https://www.sciencedirect.com/science/article/pii/B9781597491525500085

Memory and I/O Systems

David Money Harris, Sarah L. Harris, in Digital Design and Computer Architecture (Second Edition), 2013

Servo Motor

A servo motor is a DC motor integrated with a gear train, a shaft encoder, and some control logic so that it is easier to use. They have a limited rotation, typically 180°. Figure 8.62 shows a servo with the lid removed to reveal the gears. A servo motor has a 3-pin interface with power (typically 5 V), ground, and a control input. The control input is typically a 50 Hz pulse-width modulated signal. The servo's control logic drives the shaft to a position determined by the duty cycle of the control input. The servo's shaft encoder is typically a rotary potentiometer that produces a voltage dependent on the shaft position.

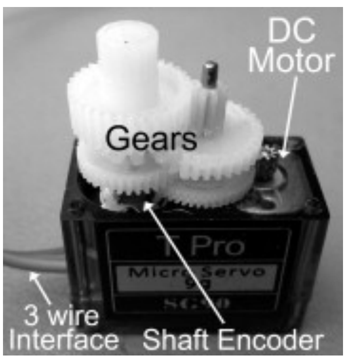


Figure 8.62. SG90 servo motor

In a typical servo motor with 180 degrees of rotation, a pulse width of 0.5 ms drives the shaft to 0°, 1.5 ms to 90°, and 2.5 ms to 180°. For example, Figure 8.63 shows a control signal with a 1.5 ms pulse width. Driving the servo outside its range may cause it to hit mechanical stops and be damaged. The servo's power comes from the power pin rather than the control pin, so the control can connect directly to a microcontroller without an H-bridge. Servo motors are commonly used in remote-control model airplanes and small robots because they are small, light, and convenient. Finding a motor with an adequate datasheet can be difficult. The center pin with a red wire is normally power, and the black or brown wire is normally ground.

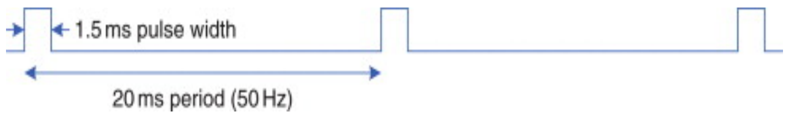


Figure 8.63. Servo control waveform

Example 8.29

Servo Motor

Design a system in which a PIC32 microcontroller drives a servo motor to a desired angle.

Solution

Figure 8.64 shows a diagram of the connection to an SG90 servo motor. The servo operates off of a 4.0–7.2 V power supply. Only a single wire is necessary to carry the PWM signal, which can be provided at 5 or 3.3 V logic levels. The code configures the PWM generation using the Output Compare 1 module and sets the appropriate duty cycle for the desired angle.

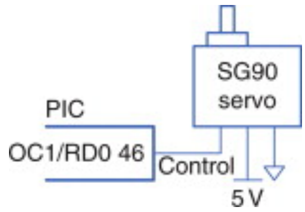


Figure 8.64. Servo motor control

```
#include <P32xxxx.h>
void initservo(void) { // configure PWM on OC1 (RD0)
 T2CONbits.TCKPS = 0b111; // prescale by 256 to 78.125
KHz
 PR2 = 1561; // set period to 1562 ticks = 50.016 Hz
(20 ms)
 OC1RS = 117; // set pulse width to 1.5 ms to center
servo
 OC1CONbits.OCM = 0b110; // set output compare 1
module to PWM mode
 T2CONbits.ON = 1; // turn on timer 2
 OC1CONbits.ON = 1; // turn on PWM
}
void setservo(int angle) {
 if (angle < 0) angle = 0; // angle must be in the
```

```
range of
 // 0-180 degrees
 else if (angle > 180) angle = 180;
 OC1RS = 39+angle*156.1/180; // set pulsewidth of
39-195 ticks //
 (0.5-2.5 ms) based on angle
}
```

It is also possible to convert an ordinary servo into a *continuous rotation servo* by carefully disassembling it, removing the mechanical stop, and replacing the potentiometer with a fixed voltage divider. Many websites show detailed directions for particular servos. The PWM will then control the velocity rather than position, with 1.5 ms indicating stop, 2.5 ms indicating full speed forward, and 0.5 ms indicating full speed backward. A continuous rotation servo may be more convenient and less expensive than a simple DC motor combined with an H-bridge and gear train.

Read full chapter

URL: https://www.sciencedirect.com/science/article/pii/B9780123944245000082

I/O Systems

Sarah L. Harris, David Money Harris, in Digital Design and Computer Architecture, 2016

9.4.4.2 Servo Motor

A servo motor is a DC motor integrated with a gear train, a shaft encoder, and some control logic so that it is easier to use. They have a limited rotation, typically 180°. Figure e9.37 shows a servo with the lid removed to reveal the gears. A servo motor has a 3-pin interface with power (typically 5 V), ground, and a control input. The control input is typically a 50 Hz pulse-width modulated signal. The servo's control logic drives the shaft to a position determined by the duty cycle of the control input. The servo's shaft encoder is typically a rotary potentiometer that produces a voltage dependent on the shaft position.

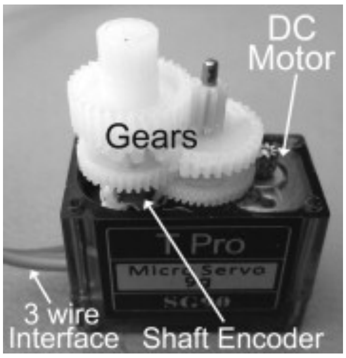


Figure e9.37. SG90 servo motor

In a typical servo motor with 180 degrees of rotation, a pulse width of 0.5 ms drives the shaft to 0°, 1.5 ms to 90°, and 2.5 ms to 180°. For example, Figure e9.38 shows a control signal with a 1.5 ms pulse width. Driving the servo outside its range may cause it to hit mechanical stops and be damaged. The servo's power comes from the power pin rather than the control pin, so the control can connect directly to a microcontroller without an H-bridge. Servo motors are commonly used in remote-control model airplanes and small robots because they are small, light, and convenient. Finding a motor with an adequate datasheet can be difficult. The center pin with a red wire is normally power, and the black or brown wire is normally ground.

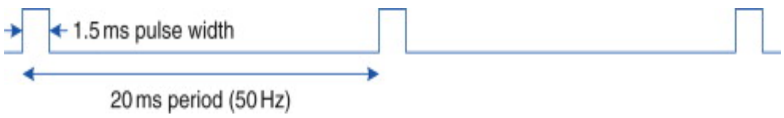


Figure e9.38. Servo control waveform

Example e9.13

Servo Motor

Design a system in which a Raspberry Pi drives a servo motor to a desired angle.

Solution

Figure e9.39 shows a diagram of the connection to an SG90 servo motor, including the colors of the wires on the servo cable. The servo operates off of a 4.0–7.2 V power supply. It can draw as much as 0.5 A if it must deliver a large amount of force, but may run directly off the Raspberry Pi power supply if the load is light. A single wire carries the PWM signal, which can be provided at 5 or 3.3 V logic levels. The code configures the PWM generation and computes the appropriate duty cycle for the desired angle. It cycles through positioning the servo at 0, 90, and 180 degrees.

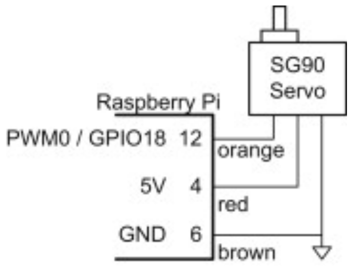


Figure e9.39. Servo motor control

```
#include "EasyPIO.h"
void setServo(float angle) {
 setPWM(50.0, 0.025  +  (0.1 * (angle / 180)));
}
void main(void) {
 pioInit();
 pwmInit();
 while (1) {
  setServo(0.0); // Left
  delayMillis(1000);
  setServo(90.0); // Center
  delayMillis(1000);
  setServo(180.0); // Right
  delayMillis(1000);
 }
}
```

It is also possible to convert an ordinary servo into a continuous rotation servo by carefully disassembling it, removing the mechanical stop, and replacing the potentiometer with a fixed voltage divider. Many websites show detailed directions for particular servos. The PWM will then control the velocity rather than position, with 1.5 ms indicating stop, 2.5 ms indicating full speed forward, and 0.5 ms indicating full speed backward. A continuous rotation servo may be more convenient and less expensive than a simple DC motor combined with an H-bridge and gear train.

Read full chapter

URL: https://www.sciencedirect.com/science/article/pii/B9780128000564000157

Reading S'ensors

In Building Robots with LEGO Mindstorms NXT, 2007

The Servo Motor Encoder (Rotation Sensor)

The legacy RCX rotation sensor was always known for its lack of reliability with readings when turning at both low and high speeds. Robot makers had to play with code to provide stability to readings returned from this sensor.

LEGO decided to integrate an encoder (rotation sensor) directly within its new NXT servo motors (Figure 4.10). There are two benefits to this: The encoder functionality was improved, and the NXT received three rotation sensors built right into the motors that don't require additional sensor ports! The interactive servo motor (as it's also referred to) allows you to measure both speed and distance in a variety of formats, including degrees, rotations, and seconds. It acts as both a motor and a rotation sensor, and has a dedicated block for each of these in NXT-G. In RobotC, you would simply set the parameters for driving the motor as you normally would while using other commands to read the encoder values to measure rotation.

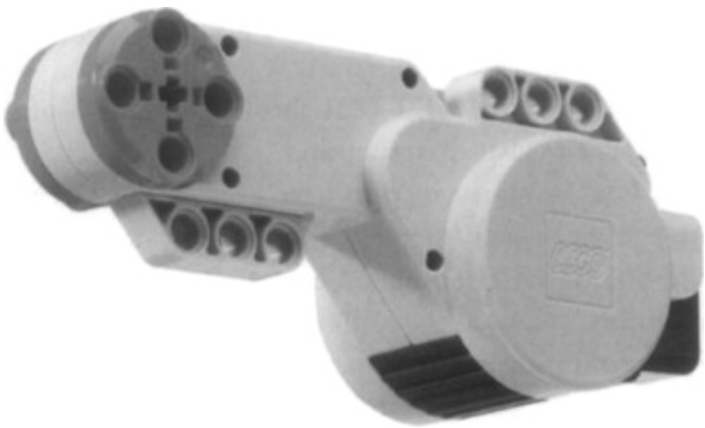


Figure 4.10. The NXT Servo Motor with Built-in Encoder

Figure 4.11 shows an internal view of the servo motor with the encoder (in blue) located to the left of the larger orange drum (the motor). In reality, the encoder is actually a black wheel that contains 12 holes which allow the optical sensor to read 24 on/off states with each full rotation. This provides the NXT with a great deal of resolution to detect position down to the nearest degree. From the image, you can also see how the NXT motor is internally geared. There is enough torque to drive wheels/tracks directly. Even though the RIS motors are also internally geared, they have limited torque that usually required an additional geartrain—especially in sumo competitions!

Figure 4.11. The NXT Servo Motor—Internals

Having an encoder built directly into the servo motor allows robot designers to develop more sophisticated drive mechanisms that enable your robot to do things such as drive straight, even over rough terrain. This functionality works out of the box with NXT-G. When you program your robot, the *move block* pairs two motors together, enabling the NXT to monitor the encoders of both motors while correcting them on the fly to ensure that the robot is tracking straight. The general idea is that the program monitors rotations on both motors. If one falls behind, it adjusts the speed of one motor to compensate for the lag, which keeps the robot driving straight.

You can try this yourself by creating the TriBot from within Robo Center (sample robots in NXT-G). Following the programming guide, you will use a *move block* to allow for both drive motors to be synchronized. Once built, run the robot and follow along beside it. Press your finger to one of the wheels and then let go. Note how at first you slow down one side of the robot, but then it speeds that side up to bring the robot back to driving in a straight line.

As mentioned earlier, you can use a single servo motor to both move an elevator as well as determine which floor it is on. With the new level of accuracy in these motors, you can determine the position of the elevator by performing some simple tests to find which angle values represent each floor. To do this, create the elevator unit and manually rotate the motor while viewing the encoder rotation values in NXT-G (or the RobotC *poll brick* window). Jot down the rotation angle for each floor. Then, simply identify in your program these angles as stop points for the elevator unit.

The encoder functionality is very powerful for the future of NXT robots, as it opens the door by enabling your robots to be “location-aware” by performing tasks such as room mapping. The sky is the limit here.

Bricks & Chips …

How the Servo Motor Encoder Works

The NXT motor encoder detects movement similar to the way an older computer mouse (with the ball) works. One of the first things Philippe Hurbain (Philo) did when he got his NXT set was to dismantle the servo motors to have a look under the hood. His site (see Appendix A) provides detail on this. Figure 4.12 shows the motor and encoder components cut away from the rest of the motor. The encoder wheel (the black wheel to the right) is driven directly off the motor. The wheel has a number of holes in it which allow the optical sensor to detect on/off states as it spins. A beam of light is generated from the optical sensor (the gray square box covering the encoder wheel) on one side of the encoder wheel and shines through to the other, which falls upon a photocell. As the motor spins, the encoder rotates and causes light to alternate through a series of on/off states. The sensor picks this up and passes the information to the NXT for processing. Unlike the older rotation sensor, because the motor is directly coupled to the encoder wheel, direction is automatically handled, as the NXT always knows which way it is driving the motor based on the programming done in the software.

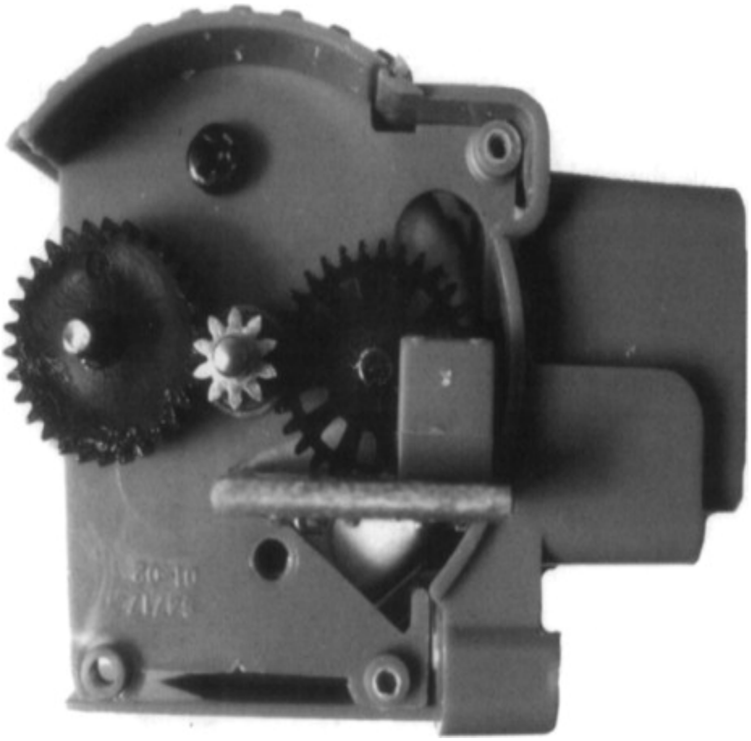


Figure 4.12. The NXT Servo Motor Encoder

Through RobotC, it is possible to detect motor stall conditions by monitoring the motor encoder rotation as your motor turns. Knowing the power and expected output of the motor, you can match this with the speed at which the encoder is actually turning and detect when the motor has stopped turning. This has an added bonus, as you could conceivably create a robot that does not need a touch sensor to detect when it has hit a wall. You can simply monitor the motor rotations and judge it this way—when you hit a wall, you can detect that the motors have either slowed or stopped, and after a period of, say, one or two seconds, force a decision to back up or turn.

NOTE

The NXT supports three types of sensors: passive, active, and digital. The main difference is that passive sensors do not require a current generator to supply power to the sensor, whereas active sensors do. Digital sensors use I^2C communication and typically have a microcontroller to handle sampling of the environment.

Passive sensors include the touch sensor (NXT and RCX), light and sound sensors (NXT), and the RCX temperature sensor. Active sensors include the RCX light and rotation sensors. Digital sensors include the NXT ultrasonic sensor and numerous third-party sensors such as color, compass, pressure, gyro, and acceleration sensors.

Read full chapter

URL: https://www.sciencedirect.com/science/article/pii/B9781597491525500097

Solving a Maze

In Building Robots with LEGO Mindstorms NXT, 2007

Constructing the Maze Runner

To construct the Maze Runner, we used two servo motors, a ball caster, and one ultrasonic (US) sensor. You can replicate the whole robot with parts solely contained in the MINDSTORMS NXT set (see Figure 17.5).

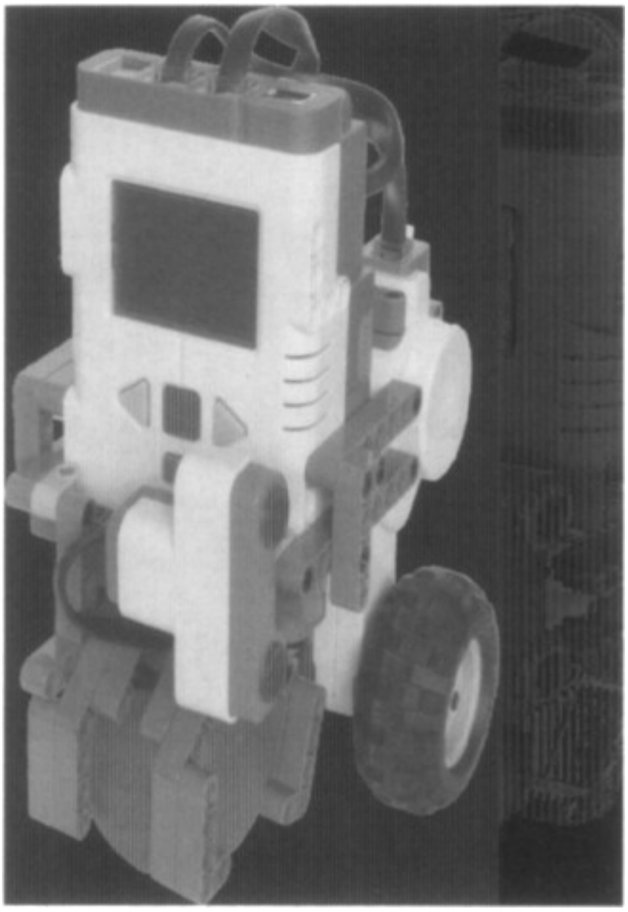


Figure 17.5. The Maze Runner

It works on a simple principle: The US sensor keeps the robot at a fixed distance from the left wall. When the distance changes abruptly, it considers this to be an opening to the left. If the gap is large enough for the robot to pass, it turns left. This covers the case of straight walls and left turns, but the robot will also have to face situations in which it hits a wall in front of it and must turn right. For this the robot monitors another encoder, and detects whether it is stalled by comparing the previous encoder reading to the current encoder reading.

We designed the Maze Runner to be as small as possible in the planar dimensions so that it can move through narrower mazes. We mounted the US sensor in front vertically to measure the distance to the left wall, while keeping the robot design compact. Figure 17.6 shows the front view of the Maze Runner, and Figure 17.7 shows the left-side view.

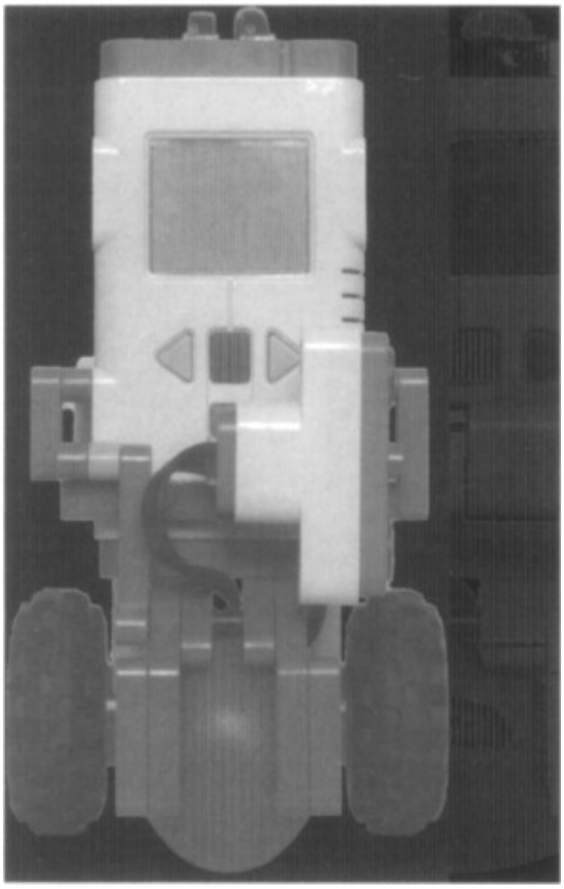


Figure 17.6. Front View

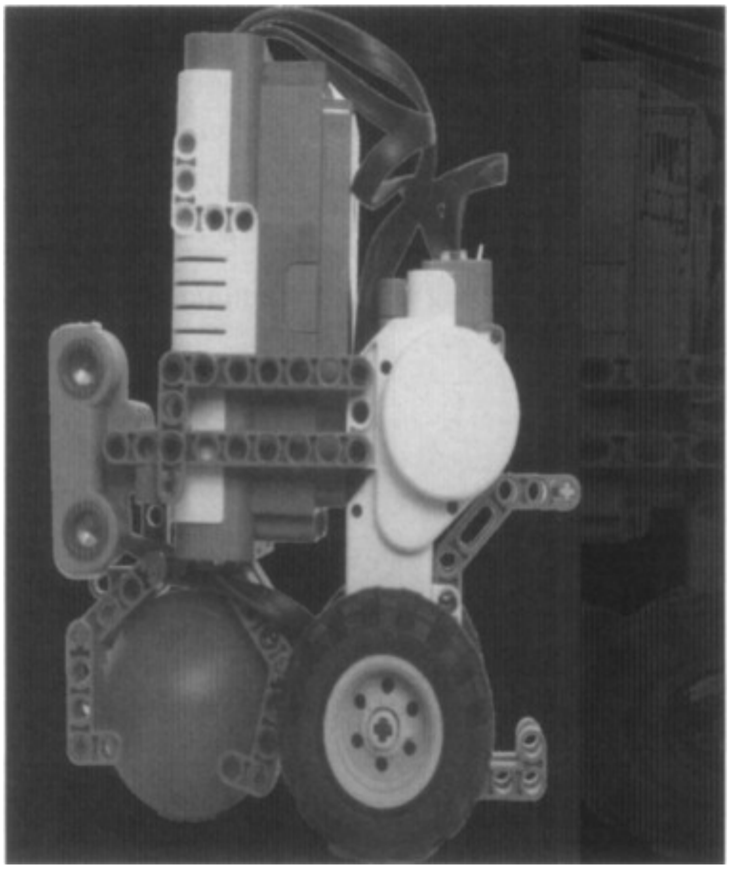


Figure 17.7. Left-Side View

As we discussed in Chapter 6, the motors are an inherent part of the robot's "chassis." In our Maze Runner, the robot is divided into three "modules": the motors, the ball caster, and the NXT brick with the US sensor. As we emphasized in

Chapter 6, building a robot in a modular fashion allows you to disassemble and fix each part without reconstructing the whole structure.

The front ball caster was designed to be as small as possible, and to provide clearance for the US sensor (see Figure 17.8). Now, you may ask why the two L-shaped five-stud pieces are in the back. Well, when the robot moves forward, these two liftarms are in the air. However, when the robot hits a wall, due to the vertical design we found it can easily fall back. These two liftarms prevent this from happening.

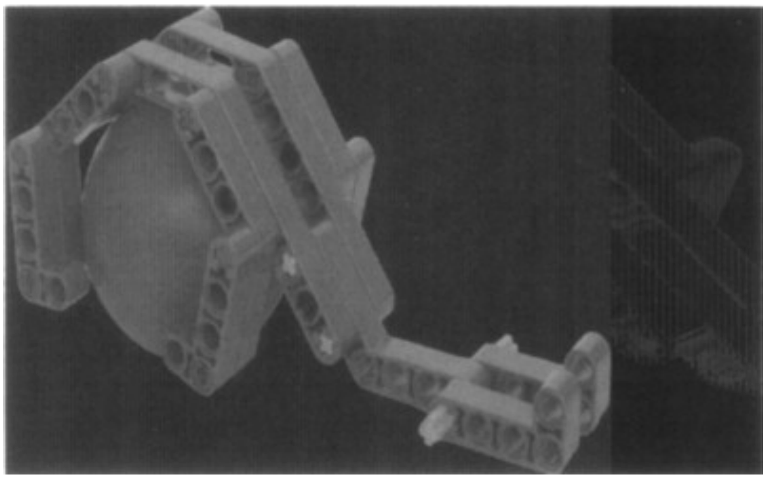


Figure 17.8. Ball Caster Design

Read full chapter

URL: https://www.sciencedirect.com/science/article/pii/B978159749152550022X

First Results of Automatic Matching System on Tore Supra ICRH Antennas; Fast Matching Network for ICRH Systems*

L. Ladurelle, ... G. Lombard, in Fusion Technology 1996, 1997

1.1 Hardware equipment

Error signals derived from line directional voltages and phase measurements are used to drive fast brushless servo motors coupled to the internal matching capacitors C_1 and C_2 of each half antenna (*figure 1*).

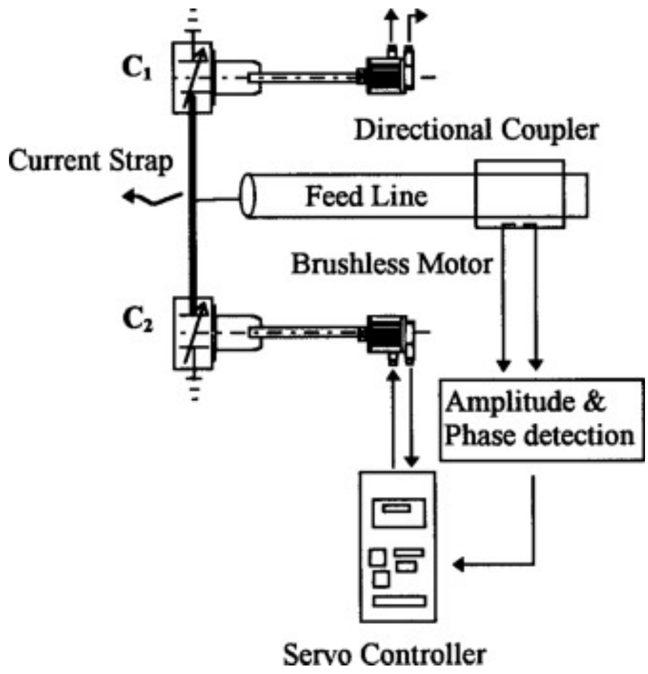


Figure 1. Hardware Equipment

- **Measurements :**

  Forward and reflected power are directly measured through a directional coupler and fast diodes supplied by SPINNER(*). Reflection coefficient phase is given by phase detectors TS35164.

- **Capacitors driving system :**

  Each antenna is equipped with a motion coordinator which drives four brushless servo motors. This multi tasking controller is provided with four analog inputs fed with reflection coefficient amplitude and phase to compute error signals. Digital inputs are used to allow capacitors motion according to antenna safety and power thresholds.

The USASEM05A brushless servo motors with optical encoders have been chosen for their reliability in magnetic environment. Tests up to 0.2 T have been successfully performed.

Read full chapter

URL: https://www.sciencedirect.com/science/article/pii/B978044482762350118X

Ball on plate modeling

Nassim Khaled, ... Affan Siddiqui, in Digital Twin Development and Deployment on the Cloud, 2020

4.4 Failure modes and diagnostics concept for the ball on plate

In this chapter, we will focus on detecting failures that affect the acceleration of the ball. These failure modes include servo motor failure such as wiring issue, aging servo, or mechanical failure in the rotating mechanism. Failure could also include a stuck ball (due to dust or other obstacles). Lastly, failure in the sensing mechanism can also affect the ball acceleration.

Regardless of the servo type or sensing mechanism, the process outlined to diagnose the system should still be applicable. The basic idea relies on acceleration measurement or estimation. If the acceleration of the physical asset deviates significantly from the acceleration of the digital twin, then a fault will be set. Usually the art of designing the diagnostic lies in implementing a logic that can look for the conditions where the deviations of the physical asset from the digital twin are magnified. This ensures a robust logic that can overcome uncertainties pertaining to modeling errors, slight aging of the hardware, uneven surface where the plate is place, and unmeasured disturbances (such as the air condition in the room blowing on the ball on plate setup). When implemented properly, implementing diagnostic enable conditions produces a detectable signal-to-noise ratio.

We will enable the diagnostic when either the servo command or the acceleration of the ball is big. For few samples of time, the digital twin and physical asset average accelerations will be compared. If the difference is bigger than a threshold, then a fault will be set. Fig. 4.4 shows the diagnostic process.

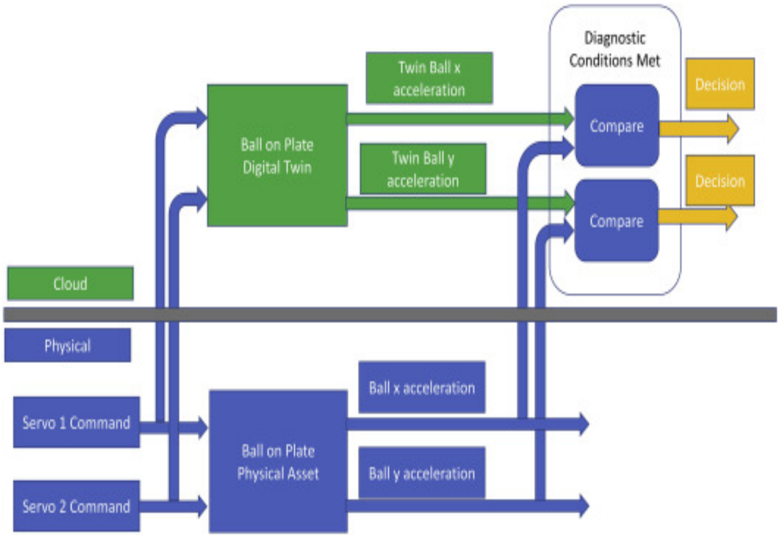


Figure 4.4. Off-BD diagnostic process for the ball on plate.

Read full chapter

URL: https://www.sciencedirect.com/science/article/pii/B9780128216316000049

Electronic and Firmware Design of the HP DesignJet Drafting Plotter

Alfred Holt MebaneIV, ... Anne P. Kadonaga, in Readings in Hardware/Software Co-Design, 2002

Print Engine Control

Print engine control is a much less demanding task than vector-to-raster conversion. Print engine control includes two-axis servo motor control, front-panel control, keyboard scanning, front-panel display update, optical sensor scanning, and thermal inkjet pen service station control.

The servo motor control functions of position decoding and pulse width modulation of the motor voltage were added to the processor support ASIC. The other functions required mostly pins with very little logic, so alternatives were investigated that could implement them more efficiently. The result of this investigation is an approach that surprised most of the design team. A single-chip processor, the Intel 8052, was able to perform these functions with a lower production cost than an ASIC and for a fraction of the development cost It also performs all of the real-time servo control, offloading this from the 80960KA.

The 8052 is designed into the architecture as a slave processor to the 80960KA. A bidirectional command and mailbox port is implemented in the processor support ASIC to allow processor-to-processor communication. Through this port the 8052 is able to return data to the 80960KA in response to commands or to generate one of several interrupts. Among these interrupts is the operating system time slice interrupt. A vast portion of print engine control is the electronics required to support the thermal inkjet pens. The most difficult function performed in this area is the mapping from the image generated in memory by the vector-to-raster converter to the series of timing pulses sent to fire the pens. This task is made difficult by the fact that the DesignJet plotter uses two 50-nozzle pens that are not accurately aligned mechanically with each other. The mapping and alignment compensation are performed by two ASICs, one located on the main board and a companion part located on the circuit board that travels on the pen carriage. The two ASICs are connected by a serial link that runs through the trailing cable. The pen interface ASIC on the main board is initialized with the measured distances between the two pens and is able to select from image memory all the dot positions that are covered by pen nozzles at a given carriage position. As the carriage scans across the page, this ASIC sends groups of 100 bits up the serial link to the carriage ASIC at 1/300-inch intervals. The carriage ASIC buffers the 100 bits and creates the timing patterns used as inputs to the drivers that generate the firing waveforms for the thermal inkjet pens.

Read full chapter

URL: https://www.sciencedirect.com/science/article/pii/B9781558607026500612

Distributed Control of Hyper-redundant Manipulator with Expansion and Contraction Motion for Obstacle Avoidance

Kenji Inoue, ... Yasushi Mae, in Human Friendly Mechatronics, 2001

5 Experiment

Fig.6 shows the experimental planar hyper-redundant manipulator consisting of 5 joint units. The link length is 80[mm]. Each joint is driven by a DC servo motor with a reduction gear and a rotary encoder. Each unit is controlled by one personal computer (Pentium processor 100[MHz], 133[MHz] or 166[MHz]), and the computers of neighboring units communicate through shared memory. The sampling period including control and communication is 2.0[ms], A infrared sensor with about 80[mm] detecting distance is attached to the tip of the unit U_3. As shown in the experimental result **Fig**.7, the end-effector can reach its desired position while the manipulator avoids the wall.

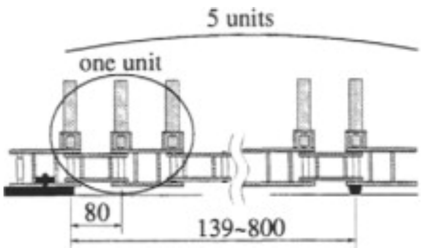


Figure 6. Experimental manipulator

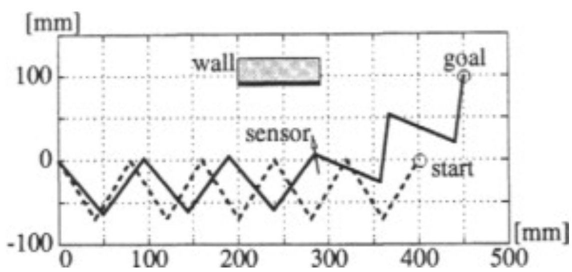


Figure 7. Experimental result

Read full chapter

URL: https://www.sciencedirect.com/science/article/pii/B978044450649850053X

Encoders and Resolvers

George Ellis, in Control System Design Guide (Fourth Edition), 2012

14.8 Choosing a Feedback Device

There are a number of feedback devices from which to choose. This section will provide an overview of some of the more popular feedback devices. The most popular rotary position feedback device is certainly the optical encoder. In the past, these sensors suffered from reliability problems, especially from the heat sensitivity of the optical electronics. This caused motor manufacturers to *derate* motors when using optical encoders, reducing the output torque of the motor to reduce the housing temperature. These problems are largely cured, and many industrial encoders can run at high enough temperatures that they have little or no effect on motor ratings. Optical encoders are available in a wide range, from kit encoders, which are inexpensive encoders that share the shaft of the motor and have a resolution of a few hundred lines per revolution and a price of about $10 in volume, to fully housed encoders with 50,000 or 100,000 lines per revolution costing several hundred dollars. As discussed before, sine encoders raise the resolution to the equivalent of 500,000 lines per revolution or more. Encoders are available in the resolution, accuracy, and temperature range required by most servo applications.

One shortcoming of optical encoders is that they rely on a masked transparent disk, which can be clouded by contaminants. In environments where the feedback device is exposed to contamination, magnetic techniques offer higher reliability. A class of encoders called *magnetic encoders* operates on principles similar to those of optical encoders but with less sensitivity to contaminants.

Resolvers are generally thought to be among the most reliable motion sensors. First, the sensing is based on magnetic techniques and so is insensitive to most contaminants. For example, in aerospace applications, resolvers are sometimes immersed in jet fuel as part of normal operation. Because resolvers have no internal electronic components, they can be exposed to extremely high temperatures. Also, because resolvers consist of copper wound around steel laminations, they can be built to be quite rugged, able to absorb high levels of shock and vibration compared with encoders. Resolvers are also inexpensive compared with encoders with equivalent resolution (that is, between 1000 and 16,384 lines).

For applications with a long distance between the motor and drive, the maximum cable length of the feedback device must be considered. For resolvers, the problem is that cable capacitance can create a phase lag for the modulated signal. Resolver signals are usually modulated at between 4 and 8 kHz. For cables that are in excess of 100 meters, the phase shift of the resolver's signals can degrade the RDC performance. Some drive manufacturers provide adjustments for the RDC circuit so that it can accommodate the phase shift generated by the long cables.

For encoders, cable length is sometimes limited by *IR* drop in the cable. Encoders usually draw between 50 and 250 mA. If the supply voltage is 5 V (as it often is), the encoder may allow only a few hundred millivolts drop in the cable. This limits the maximum resistance of the cable which can indirectly limit cable length. At least two companies, Hengstler and Stegmann, have addressed this problem by allowing users to apply an unregulated voltage to the encoder, say 7–12 VDC, which allows much higher IR drops in the cable. Encoder cable length is also limited by transmission-line effects, especially when the encoder output frequency is more than a few hundred kilohertz.

Tachometers are still used in servo applications. A DC motor, an analog drive, and a DC tachometer constitute a well-tested technology that is still cost-effective, especially in low-power applications. Also, tachometers are used in some very high-end applications, especially when the speed is very slow, such as when controlling a radar dish or telescope. Because a tachometer feeding an analog drive has no explicit resolution limitation, it can control motors rotating at small fractions of an RPM.

14.8.1 Suppliers

There are numerous suppliers of encoders. A recent search of the Thomas Register (www.thomasregister.com) showed over 100 companies that supply encoders, although only some are for servo motors. Suppliers of encoders include:

BEI (www.beiied.com)

Canon (www.usa.canon.com/cusa/semiconductor)

Computer Optical Products (www.opticalencoder.com)

Danaher Controls (www.dancon.com)

Gurley Precision Instruments (www.gurley.com)

Hengstler (www.hengstler.com)

Heidenhain (www.heidenhain.com)

Ono Sokki (www.onosokki.co.jp/English/english.htm)

Renco (www.renco.com)

Stegmann (www.stegmann.com)

Sumtak (www.sumtak.com/en/company/)

Tamagawa (www.tamagawa-seiki.co.jp)

US Digital (www.usdigital.com)

Sine encoders are supplied by a few manufacturers, including:

Heidenhain (www.heidenhain.com)

Hengstler (www.hengstler.com)

Stegmann (www.stegmann.com)

Resolvers are supplied by several manufacturers, including:

Artus (www.psartus.com)

Harowe (www.danahermotion.com)

Moog (www.moog.com/about/components-group/)

Tamagawa (www.tamagawa-seiki.co.jp)

Hardware R/D converters are manufactured by:

AnalogDevices (www.analog.com)

Control Sciences Inc. or CSI (www.controlsciences.com)

Data Device Corporation (www.datadevicecorp.com)

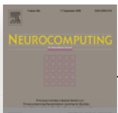
North Atlantic Instruments, Inc. (www.naii.com)

Read full chapter

URL: https://www.sciencedirect.com/science/article/pii/B978012385920400014X

Recommended publications

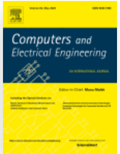
Neurocomputing



Journal

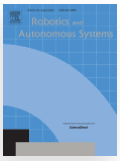
Computer Methods and Programs in Biomedicine

Journal



Computers & Electrical Engineering

Journal



Robotics and Autonomous Systems

Journal