COSC 4P78

Effectors and Actuators

Week 3

(Work based in part on Matarić)

Brock University

Intro to Effectors and Actuators

Affecting the Environment

If sensors give the robot information about the environment, then effectors are used to influence that environment

- Effectors: a device that directly affects the environment (legs, wheels, fingers, grippers, etc.)
- Actuator: the mechanism that enables an effector to execute an action; provides the force/motion/etc. (motors, hydraulics, pneumatics, etc)

Obviously the two are connected, and may seem similar (and are often used interchangeably), but they *are* different!

Actuators

Obviously, neither actuators nor effectors can be fully explained independently.

Howsabout we start with actuators?

- Outside of esoteric examples, actuators provide motion either rotationally or linearly
- A device that can provide one can often have that motion transformed into the other (e.g. via a cam)

DC Motors

- Convert electrical into mechanical energy
- Use magnets
  - *Usually* both permanent magnets and electromagnets
- Motors can be rated according to:
  - Speed
  - Voltage
  - Torque
  - Operating current
  - Stalling current
    - This will be higher than the normal operating current
  - Power takes into account both rotational velocity and torque

DC Motors
Speed and Torque

- Higher torques typically require higher current
- DC motors frequently spin faster than appropriate for their use in robotics
- We can use *gearing* to convert excess speed into additional torque
  - (Or vice-versa, but that isn't nearly as fun!)
- We can use *gearing ratios* to get the torque/speed we desire

DC Motors

Brushed

Brushed DC motors are (by far) the most common type of electric motor

- These are pretty much the only kind of motor you'll ever see in small toys/electronics

Handy terms to know:

- Field magnet: Creates a constant magnetic field
- Armature: Creates a magnetic field that may change
- Rotor: The assembly that turns
- Stator: The part that doesn't turn
- Commutator: The part that makes the electrical contact
  - As part of the rotor, turns with the armature
- Brushes: Make contact with the commutator to make the electrical connection
  - Can be made in different shapes and from different materials
    - Copper and carbon are common

DC Motors

Brushed (cont)

- Basically, a brushed motor is an electromagnet spinning between fixed permanent magnets
- The commutator acts as a series of switches
  - As the rotor (and thus commutator) turns, it automatically makes the next connection to turn on the next electromagnetic coil

DC Motors

Problems with Brushed Motors

- Typically lower efficiency
  - Have we talked about efficiency yet?
  - Efficiency in actuators refers to how much of the electrical energy gets converted into mechanical
  - The remainder is wasted (e.g. to heat)
- Brushes can wear out :(
  - Though brushes can also often be replaced
- Commutators can create electromagnetic interference!
- Creates noise in the line/power system
  - Not quite the same thing as the interference concern
  - May be mitigated with capacitors

DC Motors

Additional Concerns

- Stalling current may be unexpected
- Torque decreases as speed increases
- Induction can be a *severe* problem when current stops
  - Flyback diodes may be necessary
- How can we change the direction of the motor?
  - An *H Bridge* acts like a pair of switches to flip the direction of the current
  - Typically achieved with transistors, rather than mechanical switches
- Full torque requires full current. Suppose we wish to be able to vary speed, without sacrificing torque? (e.g. for differential drive)
  - Obviously gearing isn't an easy solution
    - Outside of complicated gear-shifting, we wouldn't be able to control speeds on the fly
  - Reducing voltage would slow it down, but at the expense of torque
  - ... wasn't there some other technique we discussed that might apply here?

DC Motors

Brushless

What if, instead of an electromagnet spinning between permanent magnets, we were to flip that around?

- The armature is part of the stator
- The rotor includes the field magnet
- We no longer have brushes or a physical commutator.
  - Instead, the rotor is *electronically commutated*
    - ⋆ One less source of electromagnetic interference!
    - ⋆ That means there are no brushes to wear out! :)
    - ⋆ It also means you need a more elaborate electronic controller :(
    - ⋆ The controller can allow for tighter speed controls
    - ⋆ Typically uses *Hall Effect* sensors when precision timing is necessary
- Usually more efficient than brushed motors
- Easier to maintain torque
- Can also be used for...

A Quick Question

How much torque do you think a typical DC motor has when it isn't spinning (or trying to spin)?

Stepper Motors

Stepper motors address two desires:

- We would like to be able to tell a motor to *not* turn
- We would like control over angles, for finer control and repeatable actions
  - To that end, we would like to be able to divide each full rotation of a motor into individual *steps*

Our solution will be based on the same principle as a brushless motor

Stepper Motors

The Concept

- The rotor contains permanent magnets
  - It actually looks like a gear, where each "tooth" has a magnetic pole
- The stator contains several coils (e.g. 8)
  - Each coil has a plate against it that also has teeth
    - This allows each tooth to act as an attractive pole to the rotor
- By applying current (of a single direction, or multiple directions), the teeth of the stator can attract the teeth of the rotor

Stepper motors

Unipolar or Bipolar?

There are two basic approaches:

- Bipolar
  - Only four wires necessary
  - Apply current one direction, and a coil is North.
    - Reverse the current and it now repels
    - Between that and alternating which coils are on, allows easy advancement
  - Requires H Bridge or similar
  - Potentially higher torque than unipolar
- Unipolar
  - At least five or six wires necessary
  - Each coil has only one pole
    - Doesn't require flipping direction of current
    - Doesn't require H Bridge
  - Switch them on and off to incrementally advance
  - Simpler, potentially weaker torque

Stepper Motors

Uses and Notes?

- Actually driven by waveforms too complicated/tedious to cover within this scope
- Typical steps per revolution: 12, 24, 72, 144, 180, or 200
  - Allows for resolution of 30 degrees all the way to 1.8 degrees!
- Good for precise control; knowing how far you've moved (or your position, if you're counting)
- Half-stepping/microstepping possible
  - Half-stepping is using two coils at once to pull a pole between them
  - Microstepping is using using a sinusoidal waveform to *very* smoothly transition
  - Both can allow for smoother motion
- Outside of complicated techniques like microstepping, can be a bit "stilted"
- If motor is stalled, it can potentially miss one or more steps
  - Obviously this is bad if you're counting the steps

Open-Loop or Closed-Loop Control

Before We Continue...

This is a topic for later, but we need to basically understand:

- An open-loop system is one in which an actuator/effector acts without “knowing” if it’s succeeding or not. For example:
  - An automated car spinning its wheels without realizing
  - A robotic arm trying to lift a train, without knowing if it’s working or not
- A closed-loop system is one in which there is *feedback*
- Closed-loop systems allow for determination of error, and thus correction (as well as possibly learning)

Servo Motors

Servos are special motors that can be directed to a specific position (angle)

- Servo motors are simply DC motors that include a *servo mechanism*
- They are composed of:
  - A motor
    - Hobby servos might just be cheap brushed motors
  - Gearing
    - Though speed is important, strength is usually *more* important
  - Some form of encoder
    - A simple potentiometer for cheap hobby servos
    - More on this in a little bit
  - Electronics

Servo Motors

How They Work

The operation of a servo is pretty simple:

- A device/robot/microcontroller tells the servo to move to an angle
- The encoder knows where the servo currently is
- The electronics tries to move the motor from where the servo is to where it should be

Servo Motors

Additional Notes

- How does the device tell the servo an angle?
  - e.g. PWM (or PPM)
    - PWM sliced into discrete chunklets
    - *Full/highest value* could mean full 180 degrees
    - *Zero/none* could mean 0 degrees
    - (Typically requires calibration or correction; mapping function)
- How can a device know the position of a servo, or if it's moving?
  - Some servos return an analog signal from the potentiometer; others communicate digitally (e.g. from optical encoders)
  - To some extent, whether or not the servo's moving can be inferred from the amount of current being drawn
- Hobby servos can be modified to create continuous rotation motors with speed control
  - Largely just a neat tidbit

Servo Motors

Uses

- Anything where you need precise control over positions
  - Technically servos can be linear, but linear mechanisms can also often be attached to rotary servos
- Manipulators, Pan'n'Tilt, robotic arms
  - Note: If incorporated into arm, that can add to the weight quite a bit!
- Turning potentiometers to adjust much stronger motors
  - Sometimes used for hobby vehicles (e.g. boats/planes)
  - Allows complete electrical isolation between control and beefier actuators

Choosing a Motor

Which Best Suits a Task?

Not surprisingly, when it comes to DC motors, servos, or steppers, the best choice will often be very task-specific

- Stepper motors are often more powerful
- Servos are great for *fast* repositioning and accurate angles
- Both servos and steppers require a controller
- Positional feedback can be *directly* incorporated into servos

Linear Motion

- *Linear motion* is simply the alternative to *rotary motion*
- Rather than a “turning” force, uses a pushing and/or pulling force
- Often includes some form of track or rail to maintain axis of motion

Solenoid

Solenoids are very similar to DC motors, but *much* simpler

- Electricity applied to an armature causes it to attract a permanent magnet or ferrous core
- When electricity is halted, a spring pushes the core to return to its original position
- (Of course, this can be reversed so the magnet/core is what moves)
- Obviously, the spring could potentially wear out
- Solenoids typically can't move very far
- They have many of the same properties/concerns as DC motors (e.g. induction)
- Often used for quickly moving small switches/valves back and forth

Linear Actuators

Linear actuators are any actuators that move along a single linear axis

Within the scope of this course, we're mostly concerned with:

- Screw
  - As a threaded spoke rotates, it forces another threaded or toothed object (or similar) along it
    - Uses a *worm gear*
    - e.g. a vice
- Wheel/belt driven
  - If a linear assembly is in place, a pulley can move a block along rails
- Rack and pinion
  - Uses a *rack gear*
- Of course, there are many variations, as well as other options
  - Pneumatics, hydraulics, cams and pistons, etc.

Linear Actuators

Uses and Notes

Typically, linear actuators are used in semi-static scenarios where motion can be limited

- Opening a door
- e.g. CNC, 3D printing, etc.

How could you know the current position (e.g. for CNC)?

Actuator Wire

Shape memory alloys are metals that can regain particular configurations when heated/cooled.

- Actuator wires (e.g. nickel-titanium) are thin wires that mimic the action/function of muscles
- Apply a current, and they contract
- e.g. Flexinol®:
  - Takes 1 second to contract
  - 0.0010" wire ($<$0.03mm) can lift 7g (with current draw of 20mA) – Takes 0.1 second to cool
  - 0.02" wire (approx. 0.5mm) can lift 3562g (7.8 lbs) (with current draw of 4 freaking amps) – Takes 17 seconds to cool
- Still limited uses
  - Smaller gauges may be suitable for moving flaps, etc.
  - Some work into reproducing human-like hands

Effectors

As previously indicated, *effectors* are used to affect the environment.

To do this, they include one or more *actuators*, as well as some additional hardware

Actuators typically control only a single *degree of freedom* each, so effectors often have to combine them to control more (and achieve their task)

Effectors

Goals

Effectors are primarily used for one of:

- Locomotion
  - Changing the position/orientation of the robot (i.e. to move itself)
- Manipulation
  - Pushing
  - Gripping
  - Turning an object
  - Somehow changing the position or orientation of some external object
  - i.e. to move *something else*

Locomotion

Actuators (and additional hardware) are mounted to the robot in a configuration that allows it to move the robot
e.g.

- Wheels (and treads)
- Legs
- Propellers
- Arms
- Flippers

Though *most* biological lifeforms aren't born with wheels, they're actually normally far easier to add and control than legs

Stability

Robotics is concerned with two forms of stability

- Static stability
  - A *statically stable* robot can stand still without falling over
  - Typically connected to the number of points making contact with the ground and the robot's *centre of gravity*
    - ★ A statically stable robot's centre of gravity must be within its *polygon of support*
  - Are humans statically stable?
- Dynamic stability
  - Dynamically stable robots can maintain stability... so long as they keep moving
- A statically stable robot might choose a *gait* that's only dynamically stable, or it may choose one that's statically stable
  - Silly example: A hexapod could move only *one* leg at a time, or could choose to leap and bound, with at most two legs on the ground at a time

Degrees of Freedom

- A *degree of freedom* describes a single property of orientation or position (pitch, yaw, roll, x, y, z).
- A single rigid body in free space has 6 degrees of freedom (all three of the above).
- The degrees of freedom of a mechanical system (the robot in its environment) is the number of values necessary to reproduce that position and orientation.
- The number of *controllable degrees of freedom* is the number of DOF that the robot can directly control with its actuators

Holonomicity

Robots, depending on their controllable DOF relative the environment of operation, may be categorized as of the following:

- Non-holonomic
  - $\#$ Controllable DOF $<$ $\#$ DOF
- Holonomic
  - $\#$ Controllable DOF $=$ $\#$ DOF
- Redundant
  - $\#$ Controllable DOF $>$ $\#$ DOF

Holonomicity

The Classic Example

Consider an automobile:

- How many controllable DOF does it have?
  - (Ignore action movies)
- How many DOF does its system have?
- How can it parallel park?

Holonomicity

What about a tank?

What if we were to use *treads* instead?

- How many DOF would it have then?

Holonomicity

SuperCar!

Well, okay, but what if we were to add *omniwheels*?

- How many DOF would it have then?

Trajectories

Trajectories allow a robot to plan its path

- This is particularly important for non-holonomic robots
- This may require discontinuous velocities
  - e.g. stop to re-orient
- This could also require some notion of error correction, but we'll return to that soon

Manipulation

- *Manipulation* is the act of effecting change on an external member of the environment
  - i.e. It's "doing something to something else"
- Manipulators are effectors that are designed for this purpose
  - e.g. Grippers, arms

Manipulation and DOF

- Consider simulating a human picking up a dry erase marker
- How many paths are there to move the robot to the marker?
- How many parameters could describe the arm's position when gripping the marker?
  - How many DOF does an arm have?
- This should hint at both why it can be difficult to create a suitable human interface for manipulators, and especially why it can be hard to program the automation of such tasks
- The relationship between actuator motion and the resulting effector motion is *kinematics*
  - Examining the reverse process to determine needed actuator motions is *inverse kinematics*
  - We might come back to both of these, but for now, know that it's icky math stuffs

Knowing One's Self

Close your eyes

Are your arms flailing about? How do you know?

- *Proprioception* refers to knowledge of the positions of effectors/actuators
- For comparison, *exteroception* is perception of the environment (i.e. what we were using sensors for last week)
- Proprioception can be very important for some tasks, and entirely superfluous for others
  - For many tasks, it's only required for specific actuators

Intermission

Let's look at a sample problem

Suppose I wanted to build a flatbed scanner, and used a stepper-driven, pulley-based linear actuator. Assume it's counting the steps for every motion, and then returning to the start each time.

What happens if the power goes out half-way through a scan?

Shaft Encoding

The position and/or distance travelled of a rotary mechanism can be determined via a *shaft encoder*.

- Shaft encoders can be indirectly used to determine speed/velocity

(Note: Most of what's coming up can also be applied to linear motion; it's just often a trickier problem for rotary mechanisms, due in part to the typically more confined spaces and lower flexibility of placement)

Shaft Encoding

Incremental

- Incremental shaft encoders are used to keep track of *relative* movement
  - i.e. It provides 'ticks' for every measureable unit of motion/rotation
- Optical encoding is based on the idea of a *break-beam* (remember that? that dealie from last week?) and a disc with slits in it, or reflectance
- Can be included in wheels to determine how far the robot has travelled
- When used for positioning of a joint, *must* assume a specific initial starting position!
- Picture time!

It should be easy to see how we can determine speed this way, but how do we determine **direction**?

Let's take a look!

Shaft Encoding

Additional thoughts on incremental shaft encoders

- Of course, it doesn't *have* to be optical. Remember what we said about fancier DC motors?
  - Hall effect sensors can also be used to judge whether or not a magnet has passed a specific point
  - Depending on various factors, you might get a cleaner signal, or otherwise find it easier (though it will still behave effectively the same)
- Where should the encoder actually go? On the axel? On the motor's shaft?
- Whenever adding encoders, besides the obvious concerns (wiring, physical space, etc.), also consider whether or not you'll need additional dedicated electronics, and what additional computational burdens they might impose

Shaft Encoding

Absolute

Absolute shaft encoders use a more elaborate disc that can be used to determine the precise (well, within the precision of the encoder) angle.

- It requires a separate emitter/detector for *each bit* of precision
- Motion (and thus speed) is calculated based on what the absolute position is, relative to where it was
- Typically more expensive
- It's usually more suited for positioning of a joint, than for something intended to rotate continuously
  - It will certainly work for both; it's simply overkill for the latter

How could we choose a binary encoding to map to angles?

Absolute Shaft Encoding

One Possible Approach

No matter what encoding we choose, we'll have to assign each value to a *range* of angles

- e.g. 6 bits (64 values) would mean each value would cover a range of 5.625 degrees

Perhaps we could simply... number them? Let's assume 4 bits:
0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 (i.e. counting to 15)

- Each value would cover a range of 22.5 degrees (e.g. 1 covers the range (22.5,45])

Is this a good idea? Why or why not?

Absolute Shaft Encoding

Gray Codes

What about 0,1,3,2,6,7,5,4,12,13,15,14,10,11,9,8?

- Sounds intuitive, right?
  - ...right?

Controllers

Let's just briefly talk about motor/servo controllers for a moment.
(Or not? I dunno)

Questions?

Comments?

Something something... date with a cup of noodles?

(Also, do we want to briefly discuss anything lab-related?)