mc375: intro to robotics
effectors and actuators.

- effectors
  - effectors and actuators
  - degrees of freedom
  - locomotion
  - manipulation
- actuators
  - what are actuators?
  - DC motors
  - gearing
  - electronic motors
  - servo motors
  - continuous rotation motors

effectors and actuators:
effectors.

- an *effector* is any device that affects the environment
- a robot's effector is controlled by the robot
- effectors can range from legs and wheels to arms and fingers
- controller has to get effectors to produce desired effect on the environment, based on robot's task

effectors and actuators:
actuators.

- an *actuator* is the actual mechanism that enables the effector to execute an action
- typically include:
  - electric motors
  - hydraulic cylinders
  - pneumatic cylinders

effectors and actuators:
terminology.

- terms are often used interchangeably to mean: "whatever makes the robot take an action"
- but they aren't the same thing

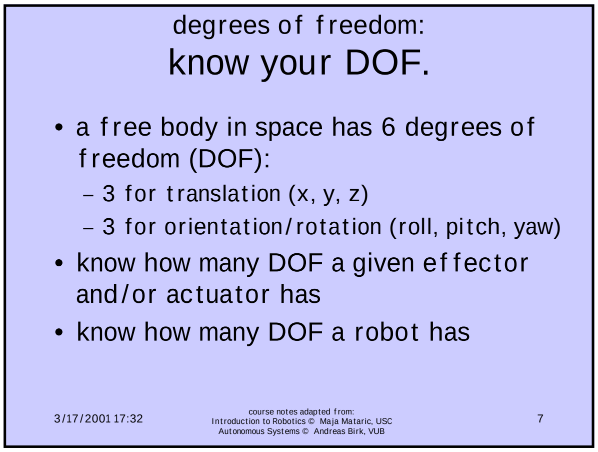
degrees of freedom:
and actuators.

- most simple actuators control one *degree of freedom*
- i.e., a single motion
- e.g., up-down; left-right; in-out
- example:
  - motor shaft
  - sliding part on a plotter

degrees of freedom:
and effectors.

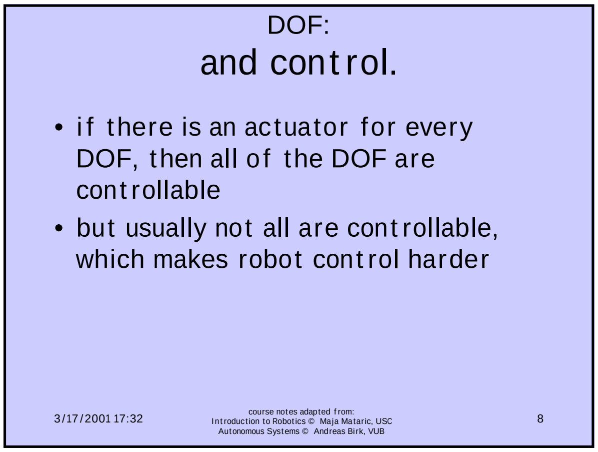
- how many degrees of freedom a robot has is very important in determining how it can affect its world, and therefor how well, if at all, it can accomplish its task

both sensors and effectors must be well-matched to the robot's task!!



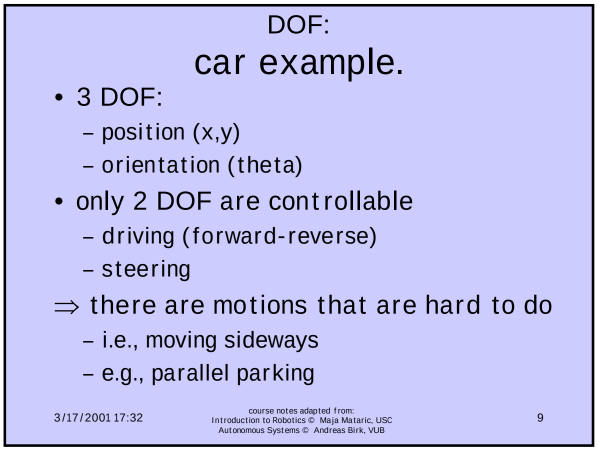
degrees of freedom:
know your DOF.

- a free body in space has 6 degrees of freedom (DOF):
  - 3 for translation (x, y, z)
  - 3 for orientation/rotation (roll, pitch, yaw)
- know how many DOF a given effector and/or actuator has
- know how many DOF a robot has



DOF:
and control.

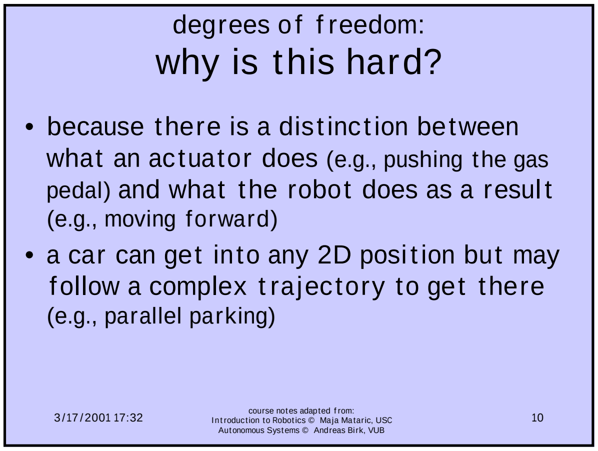
- if there is an actuator for every DOF, then all of the DOF are controllable
- but usually not all are controllable, which makes robot control harder



DOF:
car example.

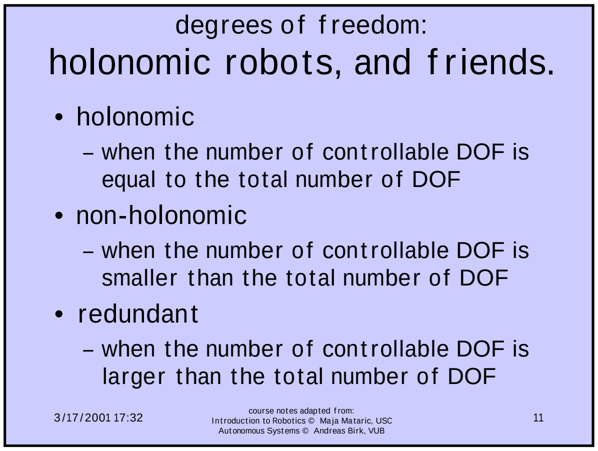
- 3 DOF:
  - position (x,y)
  - orientation (theta)
- only 2 DOF are controllable
  - driving (forward-reverse)
  - steering

⇒ there are motions that are hard to do
  - i.e., moving sideways
  - e.g., parallel parking



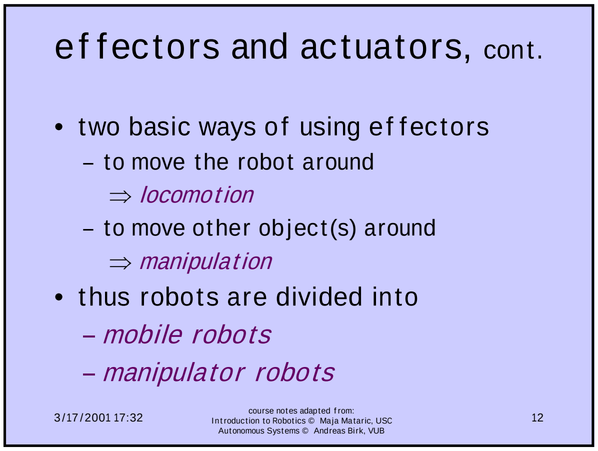
degrees of freedom:
why is this hard?

- because there is a distinction between what an actuator does (e.g., pushing the gas pedal) and what the robot does as a result (e.g., moving forward)
- a car can get into any 2D position but may follow a complex trajectory to get there (e.g., parallel parking)



degrees of freedom:
holonomic robots, and friends.

- holonomic
  - when the number of controllable DOF is equal to the total number of DOF
- non-holonomic
  - when the number of controllable DOF is smaller than the total number of DOF
- redundant
  - when the number of controllable DOF is larger than the total number of DOF



effectors and actuators, cont.

- two basic ways of using effectors
  - to move the robot around
    - ⇒ *locomotion*
  - to move other object(s) around
    - ⇒ *manipulation*
- thus robots are divided into
  - *mobile robots*
  - *manipulator robots*

locomotion.

- many kinds of effectors and actuators can be used to move a robot around
  - legs (walking/crawling/climbing/jumping/hopping)
  - wheels (rolling)
  - arms (swinging/crawling/climbing)
  - flippers (swimming)

---

locomotion:
legs vs wheels.

- while most animals use legs to get around, legged locomotion is a very difficult robotic problem, especially compared to wheeled locomotion

---

locomotion:
stability.

- robot needs to be stable
  - i.e., not wobble and fall over easily
- two kinds:
  - static: a *statically stable* robot can stand still without falling over
  - dynamic: a *dynamically stable* robot is stable only while moving

---

locomotion:
static stability.

- difficult to achieve
- *people are NOT statically stable!*
  - standing appears effortless
  - but requires active control of balance through nerves, muscles, tendons
- requires enough legs/wheels to provide sufficient static points of support
- center of gravity (COG) must fall under robot's polygon of support

---

locomotion:
polygon of support.

- the projection between of all robot's support points onto the surface
- on a 2-legged robot, this is a line
- on a 3-legged robot, this is a tripod
- when a statically stable robot lifts a leg to move, does its COG stay within the polygon of support?
  - sometimes, depending on robot's geometry.

---

locomotion:
static gait.

- statically stable gait
- assumption is that weight of leg is negligable (compared to body) so that COG is not affected by leg swing
- gait designed to keep COG inside the support polygon

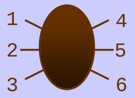
locomotion:
dynamic stability.

- allows robot to be stable while moving
- e.g., one-legged robot that can hop in place, but cannot stop (!)
- an inverse pendulum problem
  - pendulum problem: how to keep pendulum stable while moving
  - inverse: how to keep robot stable while not moving

---

locomotion:
static and dynamic stability.

- a statically stable robot can use dynamically stable walking patterns (e.g., to be fast) or statically stable walking patterns
- example: 6-legged robot
  - very stable walking gait
  - alternating tripod gait
    - 1+3+5,2+4+6
    - e.g., cockroaches
  - rippling tripod gait
    - 1,4,2,5,3,6
    - e.g., centipedes, millipedes



---

locomotion:
energy efficiency.

- statically stable walking is very energy inefficient
- dynamically stable walking enables robot to stay up while moving
  - requires active control
  - allows for greater speed
  - but is harder to control

---

locomotion:
balance.

- balance and stability are very difficult problems in control and robotics
- most existing robots have many legs/wheels
- researchers are looking at 1- and 2-legged and other dynamically stable robots

---

locomotion:
wheels.

- more efficient than legs
- most popular for robots
- statically stable
- vary in size, shape
- made of simple tires, complex tire patterns, tracks, within cylinders, within other wheels

---

locomotion:
wheels, 2.

- having wheels does not imply holonomicity
- 2- or 4- wheeled robots are usually non-holonomic
- popular 3-wheeled design uses 2 *differentially-steerable* wheels and a passive caster
  - i.e., 2 wheels can be controlled separately and thus differently from each other

locomotion:
nature.

- wheels appear in nature (in certain bacteria), though legs appear more frequently
- evolution favors lateral symmetry
- legs are easier to evolve
- insects are most populous animals -- they have 6 or many more legs!

---

locomotion:
tasks.

- following a particular trajectory
  - following an arbitrary given trajectory is harder and even impossible for some robots
  - for robots with discontinuous velocity, it is possible
- getting to a particular location
  - practical robots may not be so concerned with specific trajectories as with just getting to the goal location

---

locomotion:
following trajectories.

- a large area of traditional robotics is concerned with following arbitrary trajectories
- because planning can be used to compute optimal (and thus arbitrary) trajectories for a robot to follow to get to a particular goal location

---

locomotion:
trajectory planning.

- computationally complex process
- all possible trajectories must be found (using search) and evaluated
- robots are not points
- geometry must be taken into account
- e.g., turning radius, steering mechanism, holonomicity properties
- also called *motion planning*

---

manipulation.

- manipulator moves itself, typically to get the *end effector* (e.g., hand, finger, fingertip) to a desired 3D position and orientation
- touch your nose with your eyes closed!
- tasks: grasp and move objects

---

manipulation:
arm movement.

- get *end effector* to goal position efficiently and safely
- end effector is connected to whole arm
- have to worry about whole arm!
  - cannot violate its *joint limits*
  - cannot hit itself or rest of robot
  - cannot hit other obstacles in environment

autonomous manipulation.

- very challenging!
- first used in tele-operation where human operators move artificial arms to handle hazardous materials
- very hard for humans to control!
- duplicate human arms -- have 7 DOF!
- one method today: exo-skeleton

---

manipulation:

why is it hard?

- because there is typically no direct and obvious link between what the effector needs to do in physical space and what the actuator does to move it

---

kinematics.

- correspondence between actuator motion and resulting effector motion
- in order to control a manipulator, we have to know its kinematics
  - what is attached to what
  - how many joints there are
  - how many DOF for each joint
- this can be formalized mathematically

---

inverse kinematics.

- process of converting Cartesian (x,y,z) position (for each end effector) into a set of joint angles for the arm (thetas)
- computationally intense
- harder if manipulator is redundant

---

manipulation: recap.

- involves
  - trajectory planning (over time)
  - inverse kinematics
  - inverse dynamics
  - dealing with redundancy

---

manipulators.

- manipulators are effectors
- joints connect parts of manipulators
  - rotary (rotation around a fixed axis)
  - prismatic (linear movement)
  - robot manipulators can have 1 or more of each
- end effectors
  - simple: pointers (e.g., a stick); 2D grippers; screwdrivers
  - complex: hand with multiple fingers/joints

manipulators:

DOF.

- any free body has 6 DOF
- so in order to position a robot's end effectory to any arbitrary position in space, robot arm must have a minimum of 6 joints
- human arm has 7 DOF!
  - sufficient and redundant

what are actuators?

- mechanisms for getting things done
- in particular, getting robots to move
  - themselves (locomotion)
  - objects (manipulation)
- include:
  - pneumatics (air pressure)
  - hydraulics (fluid pressure)
  - motors (electric current)

DC motors.

- most common actuator for mobile robots
- what you will use
- DC = direct current
- simple, cheap, easy to use
- come in many sizes, to accommodate different robots and different tasks

DC motors:

how they work.

- convert electricity into mechanical energy
- consist of permanent magnets, loops of wire
- when current is applied, wire loops generate a magnetic field, which reacts against the field of the static magnets
- interaction of the fields produces movement of the shaft/armature

⇒ *electromagnetic energy → motion!*

DC motors:

efficiency.

- not perfectly efficient, as with any physical system
- i.e., energy is not converted without waste
- some energy is wasted as heat generated by friction of mechanical parts
- performance ranges from 90% (expensive motors) to 50% (cheap motors)

DC motors:

power source.

- operating voltage
  - recommended voltage range for best efficiency of the motor
  - lower will still turn motor but generate less power
  - higher may increase power but decrease life of motor
  - e.g., revving car engine makes car die sooner

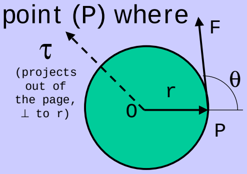
DC motors:
work.

- DC motor draws current in the amount proportional to the work it is doing
- a robot pushing against a wall draws more current (drains batteries) than when moving freely
- because wall introduces a resistance to the motor and makes it work harder

---

DC motors:
torque.

- angular acceleration of a body with rotational motion, i.e., “turning effect”
- torque (τ) exerted by force (F) about point (O), where vector ($\mathbf{r}$) extends from reference point (O) to point (P) where force acts:



$\tau = \mathbf{r} \times F$

$\tau = r\ F\ \sin\theta$

$(r = |\mathbf{r}|)$

DC motors:
torque, 2.

- equations:

$\tau = \mathbf{r} \times F$

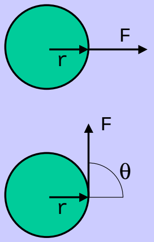
$\tau = r\ F\ \sin\theta$

- recall:

$\sin 0° = \sin 180° = 0$

$\sin 90° = 1$

$\sin 270° = -1$



DC motors:
operating current.

- within a motor’s *operating current* range, the more current is used, the more *torque* or *rotational force* is produced at the shaft/armature
- i.e., the strengths of the magnetic field generated by in the wire loops is directly proportional to the applied current and thus the produced torque at its shaft/armature

---

DC motors:
stalling.

- if resistance is very high (e.g., pushing against a solid wall), motor draws maximum amount of power and then stalls
- stall current
  - the most current a motor can draw at a specified voltage
- stall torque
  - the amount of rotational force produced when motor is stalled at its operating voltage

---

DC motors:
power.

- the amount of *power* a motor generates is the product of its shaft’s *rotational velocity* and its *torque*
  - if there is no load on the shaft (motor spins freely), rotational velocity is highest but torque is 0, ⇒ output power = 0
  - if motor is stalled, torque is maximum but rotational velocity is 0, ⇒ output power = 0

DC motors:
power, 2.

- in between, motor does useful work
- motor produces most power in middle of its performance range

---

DC motors:
speed.

- unloaded speeds range from 3000 to 9000 RPM (revolutions per minute) or 50 to 150 RPS (revolutions per second)
- DC motors are *high-speed/low-torque*
- good at driving something very light that rotates very fast
  - e.g., fan

DC motors:
adjustments.

- robots need to pull loads (their bodies, other objects), so they need *more torque* and *less speed*
- need to use *gears* to adjust performance of DC motors

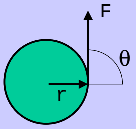
---

gears.

- torque (τ) along the line perpendicular to the circumference of a gear is equal to the product of the radius of the gear (r) and force (F) generated at the edge of the gear

$$\tau = r\ F\ \sin\theta$$

$$\tau = r\ F$$



---

gears:
combining.

- by combining gears (with different radii), we can manipulate the amount of force/torque a mechanism generates

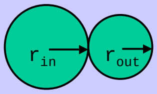
$$F_{in} = \tau_{in}/r_{in} \qquad F_{out} = \tau_{out}/r_{out}$$

- meshing input gear with output gear:

$$F_{out} = \tau_{in}/r_{in} = \tau_{out}/r_{out}$$

- and:

$$\tau_{out} = \tau_{in}\ r_{out}\ /r_{in}$$

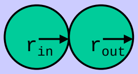


---

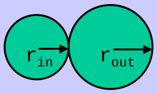
gears:
combining, 2.

$$\tau_{out} = \tau_{in}\ r_{out}\ /r_{in}$$

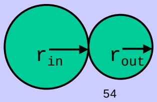
- if $r_{out} = r_{in}$, then the torque generated at the output gear is equal to the torque on the input gear ($\tau_{in} = \tau_{out}$)



- if $r_{out} > r_{in}$, then $\tau_{out} > \tau_{in}$

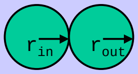


- if $r_{out} < r_{in}$, then $\tau_{out} > \tau_{in}$

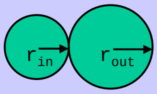


gears:
speed.

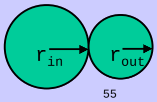
- when torque changes, speed changes (inversely proportional)
- if $r_{out} = r_{in}$, then $\tau_{in} = \tau_{out}$
  ⇒ no change in speed



- if $r_{out} > r_{in}$, then $\tau_{out} > \tau_{in}$
  ⇒ speed *decreases*



- if $r_{out} < r_{in}$, then $\tau_{out} > \tau_{in}$
  ⇒ speed *increases*



gears:
teeth.

- number of teeth is not arbitrary
- gears must mesh properly
- *backlash* = looseness between meshing gears
- *friction* = resistance between gears

gears:
ratios.

- example: 3:1 gear reduction
  - input = small gear with 8 teeth
  - output = large gear with 24 teeth
  - output speed is 1/3 slower, torque is increased 3 times
- example: 1:5 gear multiplication
  - input = large gear with 25 teeth
  - output = small gear with 5 teeth
  - output speed is 5 times faster, torque is decreased by a factor of 5

gears:
ganging.

- more than two gears can be organized in series: *ganged*
- e.g., two 3:1 gears in series result in 9:1 reduction; three 3:1 gears in series result in 27:1 reduction

this is why/how DC motors are useful!

electronic control of motors.

- motors require more battery power (i.e., current) than electronics
- e.g.,
  - 5 milliamps for a 68HC11 processor
  - 100 milliamps for a small DC motor

servo motors.

- *servo motors* = motors that can turn to a specific position (and stop)
- basic DC motors cannot do this
- servo motors are constructed out of DC motors by adding:
  - gear reduction
  - position sensor for motor shaft
  - electronic circuit to control motor’s operation

servo motors:
uses.

- toys
  - steering on RC cars
  - wing position on RC planes

    (RC = remote controlled)

servo motors:
control.

- most have movement reduced to 180º (instead of full 360 º)
- motor driven with a waveform that specifies the desired angular position of the shaft within that 180 º range
- waveform is given as a series of pulses, within a pulse-width modulated signal
- thus the width (i.e., length) of the pulse specifies the control value for the motor (i.e., how the shaft should turn)

servo motors:
control, 2.

- exact width/length of pulse is critical
  - noise w/in msec or µsec: motor will jitter
  - integrity should be checked empirically
- duration between pulses doesn't matter
  - noise w/in msec is okay
  - but when there is no pulse, motor doesn't move
  - timing should be consistent

continuous rotation motors.

- can use a regular DC motor
- can modify a servo motor:
  - remove mechanical limit (of 180º)
  - remove position sensor
  - apply 2 resistors to fool the servo into thinking it is fully turning

reading.

- Introduction to Robotics, ch 2.1-2.6, by McKerrow.