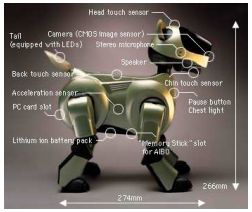
CSE-490DF
Robotics Capstone

Motion

(Based on slides by Manuela Veloso)

Sony AIBO Robot



AIBO Actuators

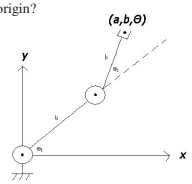
- 18 degrees of freedom with a continuously controllable range of motion
  - 3 DOF in each leg (12 total)
  - 3 DOF in the head
  - 2 DOF in the tail
  - 1 DOF in the jaw
- Each joint is controlled by specifying to a desired *joint angle* to OVirtualRobotComm.
- 2 binary motors for the ears
- A speaker for general sound production

Forward Kinematics

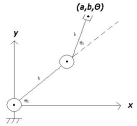
- Determines position in space based on joint configuration
- Denavit-Hartenberg Convention
- Conceptually simple (follow convention)

A simple example

What is the position and orientation of the tool (end effector) relative to the origin?



Solution



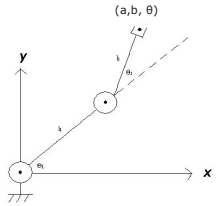
Can be solved trigonometrically!

$$a = l_1 \cos\theta_1 + l_2 \cos(\theta_1 + \theta_2)$$
$$b = l_1 \sin\theta_1 + l_2 \sin(\theta_1 + \theta_2)$$
$$\theta = \theta_1 + \theta_2$$

Inverse Kinematics

- Going backwards
- Find joint configuration given position & orientation of tool (end effector)
- More complex (path planning and dynamics)
- Possibility of no solution, one solution, or multiple solutions

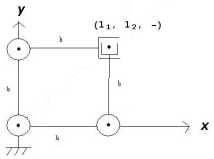
Another example



Let's assume $l_1 = l_2$

What is the configuration of each joint if the end effector is at $(a,b,\theta)=(l_1, l_2, -)$?

Solution



$\theta_1 = 0, \theta_2 = 90$

or

$\theta_1 = 90, \theta_2 = -90$

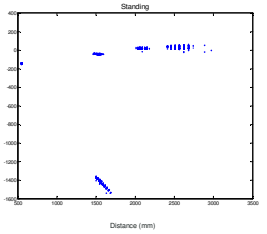
Motor Control

- Each message to OVirtualRobotComm contains a set of target angles for the joints
  - Each target is used for a PID controller (part of the AIBO robot) that controls each motor
  - Each target angle is used for one 8ms motor frame
- Each message contains at least 4 motor frames (32ms)

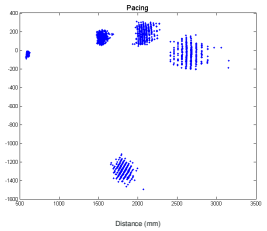
Walk Engine

- Lowest level: desired angles for joints
- Typical interface to walk engine:
  - One level of abstraction
  - More natural settings to specify desired positions
  - Includes all the kinematics computation
- Provides estimate of performed motion

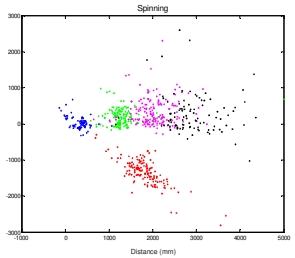
Accuracy of Object Detection Varies -- Robot Standing --



Accuracy of Object Detection Varies -- Robot Pacing --

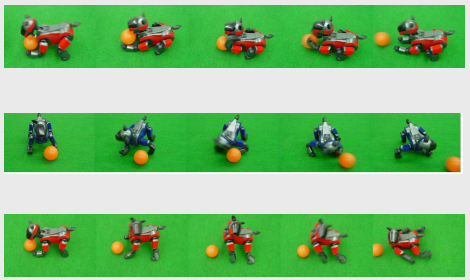


Accuracy of Object Detection Varies -- Robot Spinning --



Kicking

- A series of set positions for the robot
- Linear interpolation between the frames
  - Kinematics and interpolation provided by CMWalkEngine
- Set robot in desired positions and query the values of the joints



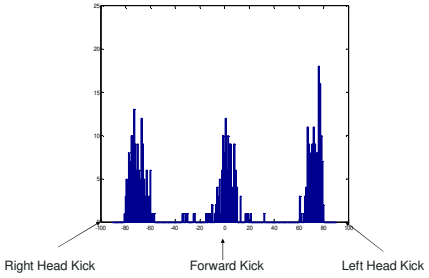
Approaches for Parameter Setting

- Trial and error
  - Tedious, but controlled, and provides knowledge of parameters
- Search
  - Large parameter space, local vs. global optima
- Adaptation
  - Controlled change by feedback

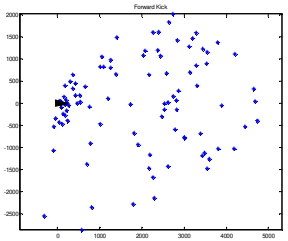
Use of Kicks in Behaviors

- Modeling *effects* of kicking motions
  - Ball vision analysis
  - Ball trajectory angle analysis
  - Kick strength analysis
- Kick selection for behaviors
  - Selection algorithm
  - Performance comparison

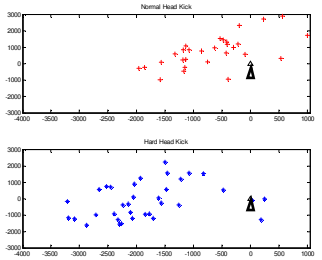
Ball Trajectory Angle Analysis



Forward Kick Distance Analysis



Head Kick Distance Analysis



Kick Selection

- Incorporate the kick models into the selection algorithm
  - The robot knows its position on the field relative to the goal and the desired ball trajectory
  - The robot selects appropriate kick by referencing the kick model
  - If no kick fits desired criteria, robot selects closest matching kick and turns/dribbles ball to appropriate position

Summary

- Effectively moving a four-legged robot is challenging
- Effectiveness of motion is highly sensitive to motion parameters
- Walk engine provides the kinematics computations, so parameter setting can be at a high level of abstraction.
- Ideally, we would like to set parameters automatically.