COMP-765B Mobile Robotics

(version 1.1 of this specification, revised from the hardcopy.)

1 Paper presentation(s)

In the coming weeks, as announced, we will be doing in-class paper presentations.

These presentations will be based on papers selected for either influence, novelty, or to establish breadth in the field. Papers are selected on a first-come first-served basis.

`http://www.cim.mcgill.ca/~dudek/765/Resources/`

The papers include:

- Real-Time Simultaneous Localization and Mapping with a Single Camera (Davison, ICCV'03)
- Distributed Algorithms for Multi-Robot Observation of Multiple Moving Targets (Parker, Auto. Rob. '02) .
- Multi-Robot Task Allocation in Uncertain Environments (Mataric, Auto. Rob. '05)
- Directed Diffusion for Wireless Sensor Networking (Intanagonwiwat, Trans. Net. '03)
- Behavior-based Formation Control for Multi-robot Teams (Balch, Trans. Rob. Auto. '99)
- Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images (Geman, PAMI '84)
- Sold!: Auction Methods for Multirobot Coordination (Gerkey, Trans. Rob. Auto. '02)
- On the Consistency of Multi-robot Cooperative Localization (Trawny, RSS '09)
- Range synthesis for 3D Environment Modeling (Torres-Mendez, ICRA '03)
- Learning Sensor Network Topology through Monte Carlo Expectation Maximization (Marinakis, ICRA '05)

- Statistical Inference and Synthesis in the Image Domain for Mobile Robot Environment Modeling (Torres-Mendez, IROS '04)
- Probabilistic Roadmaps for Robot Path Planning (Kavraki, '98)
- View-Based Maps (Konolige, RSS '05)
- Precise Positioning Using Model-Based Maps (MacKenzie, ICRA '94)
- Online Searching with an Autonomous Robot (Fekete, Comp. Geom.'04)
- Randomized Algorithms for Minimum Distance Localization (Rao, WAFR '04)
- Exponential Family PCA for Belief Compression in POMDPs (Gordon, NIPS '02)
- Bridging the Gaps between Cameras (Makris, CVPR '04)
- Minimizing Localization Travel Distance (Tovey, Trans. Rob. Auto.)
- Vast-scale Outdoor Navigation Using Adaptive Relative Bundle Adjustment (Sibley, IJRR '10)
- Underwater Human-Robot Interaction via Biological Motion Identification (Sattar, RSS '05)
- Incremental Sampling-based Algorithms for Optimal Motion Planning (Karaman, RSS '10)
- The Bayes Tree: An Algorithmic Foundation for Probabilistic Robot Mapping (Kaess, WAFR '10)
- Visual-Inertial Navigation, Mapping and Localization: A Scalable Real-Time Causal Approach (Jones, IJRR '10)
- Cross-Entropy Randomized Motion Planning (Kobilarov, RSS '11)
- Identification of Homotopy Classes of Trajectories for 3D Search-based Path Planning (Bhattacharya, RSS '11)
- Appearance-only SLAM at large scale with FAB-MAP 2.0 (Cummins, IJRR '10)
- Gaussian Processes for Signal Strength-Based Location Estimation (Ferris, RSS '02)

2 Written & practical assignment 2

This assignment has 2 parts, The first part is based on the implementation of a hardware system to do vision-based tracking using a Kalman filter, and using ROS. It is worth the bulk of the marks. The second part will be delivered next class, just to limit the amount of material that must be discussed.

2.1 The inverted pendulum

You must construct a servo-controllor that balances a vertical rod modelled as an inverted pendulum.

This system has several components: hardware, embedded software, and main computer software.

2.1.1 Principles

As discussed in class, the problem of balancing a level arm can be modeled as a pendulum, as dbut one that is positioned upside down relative to the normal arrangement of a weight on a string (hence inverted). In this case, the string supporting the pendulum is replaced by a rigid rod, and the weight can be omittted since the rod itself provides the mass. In class we discussed the model for linear inverted pendulum, which is used as a simple model for bipend locomotion, since a person walking can also be crudely modeled as a rigid rod that must be balanced by walking motion.

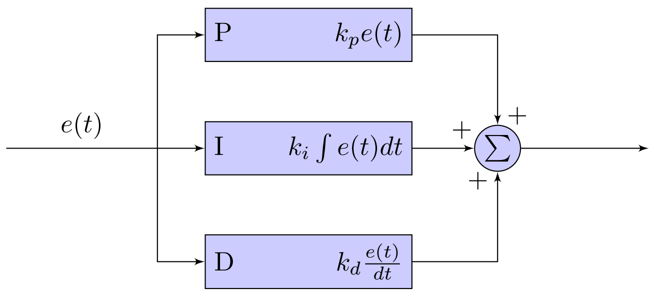
We can approximate the linear arrangement of the pendulum by putting the rod at the end of a level arm connected to a rotary joint. Teh balancing of a rod is achieved by moting the base of the rod to compensate for tipping (change of orientation) of the rod. To measure the orientation we can use potentiometer. For the purposese of this assignment, is is enough to see potentiometer ("a pot") as a rotary device that produces a voltage whose value is in proportion to a rotation angle.

A feedback controller uses a system model (also known as a plant model) to measure (or infer) the state of a system, and change the inputs to produce a desired output. When we want to output to remain fixed or follow a specific set of values, the control problem is known as tracking. This is closely related to the fucntion of a Kalman Filter, as we have discussed.

A simple design for a feedback controller is to produce changes in the input in proportion to errors in the output. This would be a "P" controller. When the input changes are proportional to the size of the tracjing error, as well as it's derivative and integral, we have the PID controller.

This is outlined in the notes from lecture 22 (see the L22 files in the class resources directory).

Further details of PID control can be found in both summary and detailed form at the web site for the classic control text **Feedback Systems: An**



Introduction for Scientists and Engineers by Karl J. Åström and Richard M. Murray:

`http://www.cds.caltech.edu/~murray/amwiki/index.php/PID_Control`

2.1.2 The hardware

This is

a computer of your choice (typically your own laptop),

a small microcontroller that takes serial commands (sent via USB) and produces hardware motor control signals,

a controllable motor (servo-controller),

and potentiometer to be mounted on the servo.

You will also need to "build" vertical pendulum (i.e. a vertical rod connected to a level arm) which can be moved using the servo, and whose orientation can be measures using the potentiometer.

The microcontroller board you will used is an **Arduino**. This term refers to a family on physically interchangeable devices that use cheap hardware, an open source design, an open source cross-platform development environment, and which can be driven with a USB cable (in most cases). The Arduino family devices all use an Atmel ATMega microcontroller chip, which is a highly integrated one-chip device with built-in memory (but not much of it).

The Arduino model you will use is the Duemilanove (or equivalent). It has about 14 digital input/output pins as well as a ground pin and a reference pin that operates as voltage reference for the analog pins. Pins 0 and 1 are devoted to serial IO. The board also has both a 3.3V and 5V output pins.

2.1.3 The methodology

The "servo" is driven using PWM commands generated using the Arduino, and usually has just 3 wires. The Arduino will produce PWM output based on serial commands that you send to it via the USB cable. These command, in turn will be produced by your software based on the output of your main program. You will also need to write code for the Arduino that converts serial commands to PWM signals. You may choose to use ROS, but it is not required.

To control the servo motor, you will need to connect wires from the servo to the $+5$, ground and one (digital) control output of the Arduino. You can use the Arduino "Servo" library to make the coding easy. See:

`http://arduino.cc/en/Reference/Servo`

For details about potentiometers, specifically in the Arduino context, see:

`http://www.arduino.cc/en/Tutorial/Potentiometer`

2.1.4 Suggested Approach

You know the motion model for the object from the lectures in class. You can use this ,combined with a PID controller to smooth noise in the tracking process and optimize performance.

The get the Arduino working. You will need to install the Arduino development environment. See

`http://arduino.cc/`

Make sure you can move the motor back and forth.

One you have these parts working, you can start putting it all together.

2.1.5 What to submit

You need to explain your approach in writing, in a paper document submitted both as hardcopy and as an email message with a video attachment. Specify what kinds of models you used (i.e. what equations), describe the code architecture, and provide illustrative results to show it all works. Include one of two sampel videos preferably encoded using MP4 codecs.

Describe the controller you use, making sure to present all the relevant equations that relate to what you are doing.' How important is the choice of controller paramaters to the correct performance of your system? (Be sure to justify your answers.)

Estimate how good your pendulum detector is.

Estimate how fast your contollers works in tems of processor capacity, range on pendulum properties or motion (which might depend on the geometry of your experimental setup).

Discuss the strengths and weakness of the system in terms of design, model, hardware and software. What shortcomings of deficiencies are there to your approach?