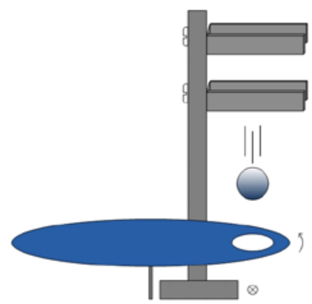
The Tunneling Ball Device

EECS c149 Project Ideas

Jeff C. Jensen
jeffcjensen@cal.berkeley.edu

February 17th, 2010

The Tunneling Ball Device



The Tunneling Ball Device is a cyber-physical system whose operation demands hardware and real-time embedded computing that deliver high-precision sensing and actuation. Computations are event driven, and signals present reflect those in an automotive engine control unit for control of fuel injection, ignition timing, and valve retraction of an automotive engine. The system is naturally extensible to a distributed platform, presenting an interesting example for modeling of distributed cyber-physical systems.

Steel ball bearings are dropped one at a time at sporadic intervals towards a fixed drop target located below a spinning disc. The disc has been bored through at two opposite ends, and the ball will pass ("tunnel") through untouched if the disc is correctly aligned at the time of impact. Should the disc be improperly rotated, the ball will impact the disc and the device has failed. The device must sense when a ball is dropped, track the position of the disc, and adjust the trajectory of the disc so that the ball tunnels through the disc untouched. Only one ball will be above the disc at any time, and between drops the disc should return to a default speed. The disc must not stop at any time, and changes in rate should be minimal.

http://chess.eecs.berkeley.edu/tbd/

1. System Modeling

Currently, the theoretical models of the physical dynamics of the Tunneling Ball Device are inconsistent with experimental results. This project will identify sources of these inconsistencies and possible solutions. Potential aspects of this project may include:

- Use LabVIEW and a data acquisition device to perform measurements of the system that can be compared to theoretical models (i.e. Bode plots, step response, controller output, etc.).
- Refine plant models, and using simulation tools like Simulink or LabVIEW, revise the current control algorithm.
- Propose and/or enact revisions to the device that mitigate dynamics that have not been modeled, such as friction.
- Use an FGPA or other embedded controller (such as a cRIO) to perform hardware in the Loop (HIL) testing.

2. Code Generation in Ptolemy II

Using Ptolemy II and its code generation framework, synthesize software to control the Tunneling Ball Device. Potential aspects of this project may include:

- Reason about software written in the PTIDES model of computation.
- Extend the Ptolemy II code generation framework to include new PTIDES actors.
- Measure performance differences between synthesized and hand-written code.

3. Upgrading the Microcontroller

The Tunneling Ball Device uses a Luminary Micro microcontroller for embedded computing, but it lacks a floating-point processor. Replace this with a more advanced controller that supports floating-point calculations. Potential aspects of this project may include:

- Replace controller with an FPGA-based CompactRIO (cRIO).
- Synthesize code and program an FPGA from LabVIEW.
- Develop a dataflow solution to the Tunneling Ball problem.