Automatic Pouring Robot

Akilah Harris-Williams
Adam Olmstead
Philip Pratt-Szeliga
Will Roantree
Overview

- Objective and Motivation
- Mechanical System
- Modeling, Simulation and Verification
  - Tilt
  - Pan
- Pouring Trajectory Generation
- Control Design
  - Tilt
  - Pan
- Webcam Image Processing
- Uncertainty Analysis
Objective and Motivation

- To develop an Automatic Pouring Robot
- Industrial Applications
  - Casting Robot
  - Hazardous Materials Robot
Purpose

- Smoothly pour water from a bottle into a cup without spilling
- Consists of a bottle, a pan/tilt mechanism, a computer controller, a cup and a webcam
- The bottle will be mounted on the mechanism
- Computer controller will rotate the bottle towards the cup
- Webcam will view the height of water in the cup.
Mechanical System
Tilt Modeling

- Mass Properties change with tilt angle and water volume. System dynamics are variable.
- Visual Basic program extracts Center of Mass, Inertial Tensor and Mass from Solidworks API at varying angles and volumes.
Tilt Friction/Gravity Model

- Upper Level System
- Tilt Dynamics
- Equation of Motion
Tilt Friction/Gravity Results

- Experimental system response versus Simulated system response to force of gravity.
Tilt Friction/Gravity Cancellation
Pan Modeling

\[ \ddot{\theta} + a_1 \dot{\theta} + a_2 \text{sgn}(\dot{\theta}) = a_3 V \]

- Used full parameter identification method
- Chirp input
- \( a_1 = 1.9080 \)
  \( a_2 = 6.0656 \)
  \( a_3 = 8.4429 \)
- 0.08 rad
  Steady-State Error
Basic Trajectory

500mL Simple Forward Trajectory

Angular Position (rad)

Time (s)
Basic Trajectories
Polynomial Trajectories

Forward Trajectories

Angular Position (rad)

Time (s)

500mL
400mL
300mL
200mL
100mL
Basic Return Trajectories
Return Trajectories

The graph shows the return trajectories over time. The blue line represents a return position of -1, while the red line represents a return position of -2. The angular position is plotted against time in seconds.
Tilt Control

- Needs to follow a smooth trajectory
- Specifications:
  - $M_0 \leq 0\%$, $t_r \leq 0.75$ s, $e_{ss} \leq 1\%$
- Used PID controller
  - System behaved similarly to model
  - Linear model assumes constant volume
  - Due to modeling error, needed an easily tweaked controller
Tilt Controller – Simulated

- **Simulated**
  - $M_o = 2.87\%$
  - $t_r = 0.327 \text{ s}$
  - $e_{SS} = 0.81\%$

- **Actual**
  - $M_o = -0.13\%$
  - $t_r = 0.104 \text{ s}$
  - $e_{SS} = 0.138\%$
Tilt Controller - Actual

- Simulation was run assuming a constant volume
- Values will change due to:
  - Constantly changing amount of liquid
  - Any modeling error
- Values tweaked from simulation to dampen overshoot
Tilt Controller on Actual Trajectory
Pan Control

- Used for Disturbance Rejection
  - Needs to have a quick rise time and low steady-state error

- Specifications
  - $M_o \leq 5\%$, $t_r \leq 0.4$ s, $e_{ss} \leq 2\%$

- Lead-Lag Controller

$$D(s) = 30 \frac{(s + 6.493)(s + 1.266)}{(s + 19.486)(s + 0.9)}$$
Pan Controller - Design
Pan Controller - Simulation

- Simulated:
  \( M_o = 0.367 \% \)
  \( t_r = 0.336 \text{ s} \)
  \( e_{ss} = 0.037 \% \)

- Actual:
  \( M_o = 5.23 \% \)
  \( t_r = 0.133 \text{ s} \)
  \( e_{ss} = 0.936 \% \)
Tweaked Pan Controller

- **Actual:**
  - $M_o = 1.55\%$
  - $t_r = 0.134s$
  - $e_{ss} = 0.016\%$
  - $t_s = 0.356s (2\%)$
  - $t_s = 1.548s (1\%)$
Sensor Subsystem - Webcam

- Objective: Pour a specified amount of liquid into a cup
- Need to know when to stop pouring to get the desired amount
- Our solution: Webcam
Image Processing Algorithm

- Want to find the volume of liquid in the cup
  - Find highest pixel row occupied by the colored liquid (determined by green and blue thresholds)
  - Correlate these pixels to actual volumes via experimentation
Webcam Timing

- Average frame rate with processing is 13 fps.
- When the volume in the cup approaches a certain percentage of the desired volume, MATLAB triggers xPC target to stop pouring.
- Percentage at which the trigger is sent is a function of:
  - Return trajectory
  - Communication delay between MATLAB and xPC Target
  - Flow rate before stopping
Webcam Data of Sample Pour

Stop Triggered
Uncertainty Analysis

- Modeling error
  - Tilt: Friction Identification
  - Pan: Steady-State Error
- Webcam error
  - Noisy data
  - Slow feedback
- Other sources of error
  - Occasional encoder spikes
Modeling Error – Tilt Friction ID

- Velocities never converge to a steady-state value, due to water sloshing in the bottle
- Different values in each direction
  - Average was taken
Modeling Error - Pan

- Identified parameters work fairly well with data used to generate them.
  - $e_{ss}$ of only 0.08 rad = 4.6°
- Does not work as well for all inputs
Assessment

- **Mechanical System**
  - Sturdy apparatus key to entire project

- **Model of Dynamic Tilt Parameters**
  - Works well after initial calibration of mass

- **Webcam Image Processing**
  - Good response but prone to inaccuracies

- **Disturbance Rejection**
  - Functions well, but with large settling time for large disturbances

- **Pouring Control**
  - Occasionally spills due to stationary cup

- **Overall Success!**
Conclusion

Areas of Improvement

- Third degree of motion
- Faster and more precise cup volume feedback
- Implement fuzzy logic controller based on constant webcam data
- Improve models and parameter identification