Robo-ELF Software

Jonathan Kriss
JHU CiiS Lab
System Overview

- **Joysticks**
- **Galil**
  - Watchdog Timer
  - Digital/Analog I/O
- **Robot**
  - Motors/Encoders
  - Potentiometers
  - Limit Switches
- **PC**
  - TaskManager
  - GUI
  - RobotTask
    - Robot Control
    - Safety Mechanisms
  - Galil Controller cisst wrapper
  - EStop
Task-level Diagram

<table>
<thead>
<tr>
<th>Task Manager</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connects mtsTask and Device objects</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RobotTask</th>
<th>qtRobotDevice</th>
</tr>
</thead>
<tbody>
<tr>
<td>devGalilController Object</td>
<td>QT widget</td>
</tr>
<tr>
<td>Robot Control functions</td>
<td>Displays robot state</td>
</tr>
<tr>
<td>Safety checks</td>
<td></td>
</tr>
</tbody>
</table>
RobotTask Functional Description

• GalilController object
• ProvidesThroatRobot Interface
• Safety checks
  – Estop Connection
  – PC Watchdog Timer
  – Encoder/Pot checking
  – Joystick valid input check
• Control Functions
qtRobotDevice Functional Description

• Requires ThroatRobot Interface
• Gets state information from robotTask
• Displays position and switch state on GUI
devGalilController Functional Description

• Cisst GenericObject
• cisst wrapper for Galil C++ API
• Receives input from:
  – Potentiometers(analog)
  – Encoders(digital)
  – Joystick controls(digital)
  – Limit switches(digital)
System Startup/Run/Shutdown

Create Tasks
- robotTask
- qtRobotDevice

Configure() / Startup() tasks
- Init Galil
- Start Watchdog
- Connect to EStop
- Build Enc/Pot tables

Run() Tasks
- Robot Control
- Safety Checks
- Error Handling/Reporting

Kill() / Cleanup() Tasks
- Safely shutdown robot and programs
- Close GUI
RobotTask Run Loop

Robot Status Checks

Safety Checks OK

Yes

Get Joystick Input

Valid Input

Yes

Robot Control

No

Log Error

Critical Error

No

No

Stop Motion

Yes

Estop too long (>0.5s)

Invalid Input

No

Yes

Estop

Print Error

System Restart Required to Continue
Safety Checks

• Estop
  – Cuts power to motors
  – Activated by robotTask, Stop button

• Watchdog Timer
  – Ensures good connection between Galil and PC
  – 50ms interval, 75ms non-critical timeout, 125ms critical timeout

• Encoder/Potentiometer checking
  – Compares values to detect encoder failure

• Software limits
  – Prevents commanded movement outside of those limits
  – Implemented with Galil’s built-in software limits

• Joystick Input Check
  – Two switches per axis on joystick input to check consistency
Safety Errors

Critical Errors

• Errors that could present a danger to the patient
• Immediate stop of robot through Estop
• Some errors can be fixed with a restart and recalibration

Non-Critical Errors

• Errors that do not pose a danger to the patient or cause the system to stop working
Safety Errors

Critical Errors
- Communication failure
  - PC <-> Galil
  - PC <-> Estop
- Encoder/ Pot Error
- Galil System Error
  - Power error, Motion Error
- Invalid Joystick input
  - Over error threshold time

Non-Critical Errors
- Galil Command errors
  - Limit switch hit, ect.
- Non-critical watchdog timeout
- Invalid Joystick input
  - Under error threshold time
Emergency Stop

• Relay controllable by serial interface
• Each run loop, check connection
• EmergencyStop() function opens relays and cuts power to motors
• Can also be activated manually with button
Watchdog Timer

- PC and Galil send handshake signal every Run() loop
  - 50ms period
- Two timeout levels
  - 75ms non-critical
  - 125ms critical
- Non-critical timeout logs error, no Estop
- Critical timeout activates Estop
- Galil program contained in Watchdog.dmc
Maximum Effects of Timeout

- Maximum uncontrolled travel distance:
  - Axis A
    - ~.8 rot/s => .1 rot / timeout
    - ~.25 rot total range (scope tip has ~.6 rot total range)
  - Axis B
    - ~.4 rot/s => .05 rot/ timeout
    - ~.75 rot total range
  - Axis C
    - ~4.25 in/s => .5 in/ timeout
    - ~6 in total range
Encoder/Potentiometer Checking

• Build tables of values for each on startup
  – New values should always be monotonically increasing

• During run, compare current values to values in tables using interpolation

• If error > margin of error, stop robot
  – Margin of error = step size of the table
Joystick Switch Checking

• Double switches on each joystick axis provide redundant sensing
• The status of the joystick switches is constantly checked to ensure they are working properly
• If an invalid switch state lasts for more than .5s, an error is thrown and the Estop activated
Robot Control

- Velocity function based on how long joystick is held down.
- Linear acceleration from V_MIN to V_MAX
- Different min/max values for each axis
Velocity Function

Velocity

V_MAX

V_MIN

Time

AccelStartTime

AccelStopTime

1s
Error Handling

• Sources of errors
  – GalilController
  – Safety Checks in robotTask

• All exceptions thrown up to high-level functions

• All errors logged to cisstLog.txt

• Safety/System failures generate simple error messages to user
Galil Exception Classes

• All exceptions thrown up to robotTask functions
  – Only ExcpCommError and ExcpSystemError handled individually

• RobotException
  – Parent class for all exceptions

• ExcpCommError
  – Generated by SendCommand when Galil returns a timeout or command error

• ExcpSystemError
  – Generic System errors, non-critical

• Classes not caught by name in robotTask
  – ExcpMotionError, ExcpPowerError, ExcpMotorOff
robotTask Exception Classes

• EStopException
  – Thrown by Estop functions if connection is lost

• EncoderException
  – Thrown by encoder/pot checking functions if error is detected. Also by startup calibration functions

• WatchdogException
  – Thrown by watchdog timer function if it times out

• Joystick Exception
  – Thrown by input function if readings are inconsistent
Error Logging

- robotTask set to ALLOW_ALL
- Std::cout set to allow errors only
  - Only displays error messages relevant to user
- cisstLog.txt set to “VERBOSE” allowance
  - Creates a more detailed log file
Qt GUI

- Displays information about current state
  - Current position (red/green), encoder values (green), pot values (green)
  - Limit switch state (blue)
- Reads from state table in robotTask
- No commands sent from GUI to robot