S.A.T.I.R.E. Milestone 3

Software Team

Taylor McRae
Sean Small
Robert Booth
Clayton Esposito

Faculty Sponsor

Dr. Phil Bernhard

Client

Dr. Stephen Wood

Milestone 2 Task Matrix Update

Task	**Complete %**	**Taylor**	**Sean**	**Robert**	**Clayton**	**To do**
1. Implement, test, and demo GPS navigation controller	80%	10%	10%	10%	50%	Implement drift capability into moos
2. Implement, test, and demo AUV dead reckoning	90%	25%	25%	25%	25%	Moos dead reckoning is functional. Will need occasional input from gps to verify.
3. Implement, test, and demo motor control	30%	5%	15%	5%	5%	Interface Moos app with the arduino control code

Milestone 2 Updates

- Connect/interface the Arduino with the MOOS app
- Dead reckoning handled by MOOS IvP helm is in place only additions that will be needed if periodic checks of current position differ from GPS.
- Motor control through arduino board was functional, but has been down since removal from the design center. We have access to the arduino code and are interfacing the arduino with the MOOSDB.
- Working on a simplified motor simulation

Milestone 3 Task Matrix

Task	Complete %	Taylor	Sean	Robert	Clayton	To do
1. Implement, test, and demo sensory input	50%	5%	30%	5%	10%	Demo pressure sensor and AHRS Sparton
2. Implement, test, and demo collision avoidance system	25%	10%	5%	5%	5%	Moos App created and capable of fetching messages from MOOSDB implemented
3. Implement, test, and demo Emergency System	40%	5%	5%	20%	10%	Moos App created and capable of fetching messages from MOOSDB implemented
4. Implement Mission Planner	55%	5%	5%	5%	40%	Finish adding the .moos config blocks and .bhv behaviors, implement GUI

Milestone 3 Discussion

- Basic arduino code is done but have to be modified
- No access to hardware so have worked on simulations
- Interface sensor controller MOOS Apps with the arduino code.
- Collision avoidance is still in the early stages. Support in place but needs more tweaking.
- The emergency system is in progress, the system needs to receive and eval sensor input. Response will be based on mission type. (Scuttle or call for help).
- Mission planner will generate MOOS missions based on user input (GPS coords, etc.)

Sensory Input

- Basic arduino code is done but have to be modified
- Testing the sensors using an arduino simulator
- Successfully tested some of the sensors needed for the vessel

Collision Avoidance System

- MOOS has functions that support avoiding or traversing difficult or impassible terrain.
- Obstacles had to be hard-coded, actual field testing isn’t guaranteed.
- Support for the collision avoidance is in place, but still needs to be improved
- Not completely done, but will be finished by Milestone 4

Emergency System

- Fetch sensor data from the MOOSDB for each update.
- Check the sensor data against safe ranges and make a decision on course of action in event of damage.
- Decisions are based on mission type with two basic outcomes, send a message with a report to the operators or scuttle the AUV.

Mission Planner

- User interface to select what behaviors are needed for the mission.
- Input of behavior specific variables. For each behavior such as target location and area of operation.
- Input mission type for emergency system to determine response in event of system damage.
- Select which MOOS Apps will be needed for the mission and input configuration values.
- Generate the .moos and .bhv files for transfer to the device.

Demo - Gas Sensor

https://circuits.io/circuits/3349347-demo-gas-sensor

Milestone 4 Task Matrix

Task	Taylor	Sean	Robert	Clayton	To Do
1. Implement, test, and demo more sensors	20%	40%	20%	20%	Add more sensors, and test.
2. Implement, test, and demo collision avoidance system	40%	25%	25%	10%	Implement object recognition and decision process.
3. Implement, test, and demo Emergency System	20%	20%	40%	20%	Continue work on emergency system, separate response based on mission type.
4. Finish implementation of the Mission Planner and implement the GUI	10%	10%	10%	70%	Finish adding config blocks and behaviors, implement gui.

Milestone 4 Discussion

- Add more sensors to the controller and arduino. Test sensor to MOOSDB communications. Test MOOSDB to client app communication on the sensor channels.
- Collision TODO
- Continue work on the emergency system. Implement responses. Implement disk wipe and interface with comms system.
- Finish adding the Config blocks and Behaviors to the planner and implement the GUI.

Questions?