CS-417 INTRODUCTION TO ROBOTICS AND INTELLIGENT SYSTEMS

Underwater Robotics
What are robots best suited for?

- Environments that are dangerous.
- Environments that are inaccessible.
- Environments that are taxing.
- Environments are expensive to access.
- Environments that are inhospitable.

Undersea: inaccessible, dangerous, costly, demanding.

As we all know, most of the world is undersea, yet it’s the environment on earth we understand the least well!
Coral Reefs

Oceans: 70% of earth’s surface.
Reefs: Greatest diversity / area of any marine ecosystem

4-5% of all species (91,000) found on coral reefs

Significant to the health of the planet:

1/2 of the calcium that enters the world’s oceans/year is taken up and bound into Coral Reefs as Calcium Bicarbonate
Coral reefs are found in polar, temperate and tropical waters
Highest diversity of species in tropics
Found in 20 degree C surface isotherm
Optimal temperature for coral is 23-25 degrees C.
Atlantic

More common in Atlantic:

- Branching coral (3 sp)
- Fire Coral

Dominant coral types:
- Branching coral (3 sp)
- Fire Coral

Sea fan

Sea Whip
Why Study Coral Reefs?

• Most biologically diverse and sensitive marine ecosystem
• Dramatically altered by humans
• By 1998, 27% of reefs were destroyed
  – 16% was from coral bleaching event (El Nino)
Coral Reefs

• Reefs are regions of *exceptional* biodiversity.
• 20% of the world’s reefs have been destroyed.
• 24% of reefs are under imminent threat of collapse due to human pressure, 26% under longer term threat of collapse!
  
  Dec. 2005 there was a terrible coral bleaching (and destruction) in the Caribbean.

  95% of Jamaica’s reefs are dead or dying.
• If we want to make things better, we need to be able to measure the changes!
• This is taxing, error-prone, tiring and dangerous.
Underwater vehicles

Autonomous Benthic Explorer (ABE)
1200 pounds and a little over 2 meters long.
Lobster like Robot
Glider UW Robot
Enabling Autonomous Capabilities in Underwater Robotics

• This work was presented at the International Conference on Intelligent Robots and Systems (IROS), 2008, at Nice, France
Overview

Technologies to increase the level of autonomy

• AQUA description
• Guidance and Control
  – Hovering
• Terrain Classification
• HRI
• Underwater Sensor Nodes
  – Video Mosaics
About Aqua

- Legged swimming vehicle
  - Hexapod with flippers, descendant of RHex
  - High mobility (can also walk, hover, etc)
- On-board cameras, IMU, computers
- Power autonomous for ~5+ hours
- Application: surveillance and monitoring of coral reefs, working in conjunction with marine biologists(s).
AQUA Components

Computation
AQUA operates with a Pentium CPU on a PC/104 stack and relays command and sensor information via a fiber optic tether.

Power
Two MIL-spec NiMH batteries allow AQUA to operate for over 5 hours underwater. Tool-less battery replacement allows quick and easy swaps for rapid redeployment.

Shell
Rugged shell design provides ample seal for up to 20m water depth and heavy impact protection.

Propulsion
Multiple iterations and tests have brought the biologically inspired flippers to generate optimal thrust. Experimentation with new swimming gaits has allowed for further improvement of AQUA's underwater performance.

Vision
2 front board cameras and 1 rear allow for remote operation of the robot. Future work will allow for visual servoing and stereoscopic 3D terrain mapping

(AQUA version 1)

mass = 18.5kg (ballasted for salt water)
AQUA objectives

- AQUA is about developing a portable robot that can walk and swim, and which exhibits the ability to use vision and/or sound to know where it is and what is near it.

- The robot could be used, for example, to survey and monitor the conditions on a coral reef. By being able to land on the bottom and move around, the robot can make regular observations without disturbing the natural organisms.

- The ability to walk, swim and use vision underwater is unique to AQUA (derived from RHex [Buehler et al.])

- Allows for efficient station-keeping and surveillance.
Project objectives

• Survey and monitor the conditions on coral reefs
• Ability to walk on land, swim, and use vision underwater
• Ability to land on the sea floor
Autonomy

Operation Methods
• Tele-operation
• Partial Autonomy-HRI
• Full Autonomy
Guidance

• Small, light, moderate-cost robot
• Learn trajectories by (initially) following a diver
• Diver specifies specific actions as desired
• Diver specifies where and how data is collected
Alternative Entry Technique
Hovering combines two distinct leg motions.

Can also selectively tune thrust direction to minimize disturbances.

Combining hovering with motion can lead to interesting planning issues.

[Diagram showing pressure drag and thrust from oscillation]
Controllers: Objectives

• Provide trajectory tracking capabilities to the vehicle
  – Determine the required paddle force
  – Determine the appropriate paddle motion

• Stabilize the vehicle in the presence of disturbances
**Linear Model**

- Nonlinear model is linearized to allow use of linear systems theory

\[
\dot{x} = f(x, \tau)
\]

\[
\dot{x} = Ax + B\tau
\]

- State vector \( x = [u \ v \ w \ p \ q \ r \ x \ y \ z \ \phi \ \theta \ \psi]^T \)
- Force vector \( \tau = [f_{x1} \ \ldots \ f_{x6} \ f_{z1} \ \ldots \ f_{z6}]^T \)
Model Based Control

• PID controllers used
• Both Linear and Non-Linear models used to augment the PID controller
Stability Augmentation System

- Linear system is weakly unstable in yaw
- SAS aims to return state perturbations to zero

\[
\dot{x} = Ax + B\tau
\]
Experimental Validation

- Forward velocity of approximately 0.5m/s
- Roll and pitch impulse disturbances introduced by a swimmer
- Inertial Measurement Unit (IMU) data logged

**Stability Augmentation and Model Based Control** improved performance
Results – Roll and Pitch Disturbance

\[ K_p = 0, \ K_\phi = 4, \ K_q = 1, \ K_\theta = 8, \ K_r = 0, \ K_\psi = 0 \]
Terrain identification

- Vehicle is capable of using contact forces to identify terrains
- This allows gaits to be selected or adapted as a function of terrain type
Vision-Based HRI

- Easier than conventional methods (e.g. type, touch screens)
- Requires no extra input mechanisms or sensors other than a camera
- Advantages of machine vision
  - Problems lie in interpreting 'gestures'
  - Fiducials as tokens
Visual Language

• Gestural robot programming language
• Real-time interpreter
• Low-level constructs: robot action commands (e.g. MOVE_FORWARD)
• High-level constructs: loops, iterators, functions
• Commands coded in scripting language (Lua)
Features

for (i = 0; i < 4; i++) {
    angle = 90;
    duration = 2;
    Turn_Left(angle, duration);
    Move_Forward(duration);
}

4 REPEAT
    9 0 ANGLE
    2 DURATION
    TURN_LEFT
    MOVE_FORWARD
END
EXECUTE

C-like Pseudocode
(38 input tokens)

RoboChat snippet
(11 input tokens)

• Use of Reverse Polish notation to minimize unnecessary syntax artefacts (e.g. then, {...} etc)
Vehicle/Sessile Multirobot network

• Sessile sensor nodes
  – Some close to one another (metric relations)
  – Some well separated (metric or topological relations)

• Moving vehicle(s)
  – Vehicle-carried odometry (VCO: topological -> metric)
Sensor nodes
Noisy data collected from an underwater node
Conclusions

• Autonomy in underwater scenarios is challenging
• Model based control increases the operational capabilities of our vehicle
• AQUA-Diver communication increased the autonomy capabilities of the vehicle

Future Work

• Cooperation between AQUA and the Sensor Nodes
• Develop Image based Localization
• HRI employing the Microsoft Robotics Studio

AQUA ROBOT is available to other labs
http://www.aquarobot.net
Questions
Controllers: Inputs

• The inputs to the controller are actual and desired trajectory:

  Desired velocity: \( \mathbf{v}_d = [u_d, v_d, w_d, p_d, q_d, r_d]^T \)
  Desired position: \( \mathbf{s}_d = [x_d, y_d, z_d, \phi_d, \theta_d, \psi_d]^T \)
  Actual velocity: \( \mathbf{v} = [u, v, w, p, q, r]^T \)
  Actual position: \( \mathbf{s} = [x, y, z, \phi, \theta, \psi]^T \)
  Velocity error: \( \mathbf{e}_v = \mathbf{v}_d - \mathbf{v} \)
  Position error: \( \mathbf{e}_s = \mathbf{s}_d - \mathbf{s} \)
Controllers: PID

Controller form: \( f = K_d e_v + K_p e_s + K_I \int e_s dt \)

\( K_d, K_p \) and \( K_I \) are diagonal matrices with positives entries.

Equation of motion: \[
\begin{align*}
M\ddot{v} + C(v)v + D(v)v + g(n_2) + b(n_2) \\
- K_d e_v - K_p e_s - K_I \int e_s dt = 0
\end{align*}
\]
Controllers: Model-based Linearizing

• The objective of this controller is to remove every nonlinear term in the equation of motion of the robot

• This gives a linear system with decoupled degrees of freedom

Controller form:

\[ f = M \ddot{v} + C(v)v + D(v)v + g(n_2) + b(n_2) \]
\[ + K_d e_v + K_p e_s + K_I \int e_s dt \]

Equation of motion:

\[ M \ddot{e}_a + K_d e_v + K_p e_s + K_I \int e_s dt = 0 \]
Controllers: Model-based Linearizing

• The objective of this controller is to remove every nonlinear term in the equation of motion of the robot

• This gives a linear system with decoupled degrees of freedom

Controller form:

\[
f = M\dot{v} + C(v)v + D(v)v + g(n_2) + b(n_2) + K_d e + K_p e_s + K_I \int e_s \, dt
\]

Equation of motion:

\[
M \ddot{e}_a + K_d e + K_p e_s + K_I \int e_s \, dt = 0
\]

Also a more complex Non-Linear controller is used
Controllers: Model-based Nonlinear

- The objective of this controller is to input the ideal force that would be required to achieve trajectory tracking.
- The proportional, integral and derivative gains were added to account for uncertainties in the model.

Controller form:

\[ f = M \dot{v}_d + C(v_d)v_d + D(v_d)v_d + g(n_2) + b(n_2) + K_v e_v + K_p e_s + K_I \int e_s dt \]

Equation of motion:

\[ Me_a + (C(v_d)v_d - C(v)v) + (D(v_d)v_d - D(v)v) + K_v e_v + K_p e_s + K_I \int e_s dt = 0 \]
Simulation results: Maneuvers

Surge maneuver:

\[ x_d = \frac{200}{3} \left( \frac{t}{100} \right)^{1.5} + \frac{1}{2} \left( 1 - \cos \frac{t}{3} \right) \]

\[ u_d = \sqrt{\frac{t}{120} + \frac{1}{6} \sin \frac{t}{3} \text{ m}} \]

Roll maneuver:

\[ \phi_d = \frac{1}{2} \left( 1 - \cos \frac{t}{4} \right) \text{ rad} \]

\[ p_d = \frac{1}{8} \sin \frac{t}{4} \text{ rad} \]
SAS Results – Roll Only Disturbance

![Graph showing roll, pitch, and yaw with SAS results and Kφ values]
SAS Results – Pitch Only Disturbance

![Graph showing roll, pitch, and yaw responses with SAS and different gain settings.]

- Roll (deg) vs. time (s)
- Pitch (deg) vs. time (s)
- Yaw (deg) vs. time (s)

- Blue line: no SAS
- Red line: $K_\theta=5$, $K_q=1$
- Green line: $K_\theta=8$, $K_q=1$

CS-417 Introduction to Robotics and Intelligent Systems