2 Project Narrative – Research Effort

2.1 Introduction

In April of 2010, an explosion on the Deepwater Horizon oil drilling platform in the Gulf of Mexico initiated an underwater oil leak which continued unabated for more than three months. The underwater leak spilled more than 4.9 million barrels of crude oil into the Gulf of Mexico, making it the largest ocean-borne oil spill in history [76]. This is a tragic example of a large scale environmental process raging uncontrolled, and wreaking irrevocable environmental, human, and economic damage as a result. Sadly, this is not an isolated occurrence, but a class of harmful, large scale environmental problems that today remain beyond the scope of human control. For example, uncontrolled forest fires routinely threaten sensitive forests, residential areas, and the lives of firefighters who work to control them; industrial chemical spills and nuclear accidents endanger wildlife and human lives; and our agricultural industry is at the mercy of pest infestations, agricultural diseases, and capricious weather cycles. This project proposes an engineering solution for controlling large scale phenomena in order to alleviate the effect of such environmental problems. We propose to close a control loop around large scale environmental processes with an Environmental Sensor-Actuator Network (EnviSAN); a group of mobile robots with the capacity to sense their environment locally, to produce localized control actions on their environment, and to communicate with one another over a wireless network. The goal of this project is to automatically produce an actuation strategy that the robots can use to collectively control a given environmental phenomenon. In short, the EnviSAN project proposes to control large scale environmental processes through the intelligent intervention of a team of autonomous agents carrying out repeated, localized actions.

Figure 1: Illustration of Environmental Sensor-Actuator Networks (EnviSANs) controlling three different distributed environmental phenomena. In 1(a) autonomous boats clean up an oil spill, in 1(b) ground robots selectively water crop rows in a farm, and in 1(c) aerial spraying robots reverse the spread of a destructive fungus in a forest.

The key to enabling such environmental control lies in both identifying a dynamical model of the environmental process to be controlled, and automatically designing a decentralized feedback control strategy based on this model. This feedback strategy itself must estimate the state of the large scale environment, and use this state estimate to apply appropriate control actuation to the environment. In order to carry out such a controller, the agents in our Sensor-Actuator Network must have sensors to collect information about the environment, for example cameras, laser range finders, or chemical sensors. They must also have an actuation mechanism to effect a change in the environment. These actuators will be specific to the environmental process to be controlled; for example, oil vacuuming units for cleaning an oil spill, water spraying apparatus for watering crops, or targeted anti-fungal sprayers for combating fungal infestations, as illustrated graphically in Fig. 1.

We proposed a two-tiered control strategy for closing the loop around the environmental process. Firstly, a high-level controller specifies a feedback law for producing control actions on the environment, and generates a planned trajectory for the robots to follow in order to administer these control actions. The low-level controller then controls the robots to realize the trajectory commanded by the high-level controller. For example, the high-level controller for controlling a fungal infestation will specify an anti-fungal spraying policy and trajectories along which the robots should spray. The low level controller will then be tasked with driving the robots along the desired trajectories and delivering the desired spray actuation. The robots coordinate their efforts to carry out the control strategy by communicating over a wireless network. We propose a similar two tiered approach for estimation and for model identification, as well. One of the main challenges is then balancing the priorities between the competing incentives of control, estimation, and model learning. This two tiered distributed control architecture is shown graphically in Fig. 2.
Multi-Robot Systems

Large scale environment

Actuation

Communication

Sensing

Distributed

Computation
Nature is Distributed

Morphogenesis  Herding  Flocking

Swarming  Human group dynamics  Cities and human institutions
Prototypical Application

Deployment and coverage
Adaptive sampling
Self maintenance
Environmental estimation
Model learning/system ID
Environmental control
Other Application Areas

- Chemical spill clean up
- Agriculture
- Infestation Control
- Disaster relief
- Urban surveillance
- Transportation systems
Multi-robot Systems Lab
Multi-robot Systems Lab

Motion capture system

Flying Arena, 10m x 5m x 3m

Floor projection system

Robot fleet
Multi-robot Systems Research

- Deployment and coverage
- Coordinated agile maneuvers
- Model learning/System ID
- Multi-robot manipulation
- Active estimation with guarantees
- Multi-robot herding
- Neighbor trust adversaries
- Persistent monitoring
- Active 3D vision and control

Platform Lab Seminar
April 19, 2016
Multi-robot Systems Research

- Deployment and coverage
- Coordinated agile maneuvers
- Model learning/System ID
- Multi-robot manipulation
- Active estimation with guarantees
- Multi-robot herding
- Neighbor trust adversaries
- Persistent monitoring
- Active 3D vision and control
Deployment and Coverage
Existing Methods

Geometric:
- **L. C. A. Pimenta, V. Kumar, R. C. Mesquita, G. A. S. Pereira.** Sensing and coverage for a network of heterogeneous robots, *CDC 08*.

Probabilistic:
- **W. Li, C. G. Cassandras.** Distributed cooperative coverage control of sensor networks, *CDC 05*.
- **M. Pavone, S. L. Smith, F. Bullo, E. Frazzoli.** Dynamics multi-vehicle routing with multiple classes of demands, *ACC 09*.

Potential Fields:
- **A. Howard, M. J. Mataric, G. S. Sukhatme.** Mobile sensor network deployment using potential fields, *DARS 02*.
Deployment Optimization

Theorem: Robots converge to a local minimum of the cost function $\mathcal{H}(p_1, \ldots, p_n)$.

Gradient Controller: $\dot{p}_i = \nabla \mathcal{H} \mid_{\mathcal{H}(p_1, \ldots, p_n)}$

$n$ robots in $\mathbb{R}^2$

Deployment of a Flying Camera Network


Sensor Cost from Optics

From optics and geometry:

\[ f(p, q) = \begin{cases} \infty & \text{for } q \in \mathcal{B} \\ a(b - z)^2 & \text{otherwise,} \end{cases} \]
Coverage Cost Function

Cost function:

\[ \mathcal{H} = \int_{Q} g(f(p_1, q), \ldots, f(p_n, q), q) \phi(q) \, dq \]

Combined cost for multiple cameras

\[ g(\cdots) = \left( \sum_{i \in \mathcal{N}_q} f(p_i, q)^{-1} + w^{-1} \right)^{-1} \]

Prior area/pixel
Gradient Based Controller

Controller: \[ \dot{p}_i = -k \frac{\partial H}{\partial p_i} \]

Components: \[ p = [c^T, z]^T \]

Lateral Component:

\[ \frac{\partial H}{\partial c_i} = \int_{Q \cap \partial B_i} (h_{\mathcal{N}_q} - h_{\mathcal{N}_q \setminus \{i\}}) \frac{q - c_i}{\|q - c_i\|} \phi(q) \, dq \]

- Move inside \( Q \)
- Move away from neighbors
Gradient Based Controller

Vertical Component:

\[ \frac{\partial H}{\partial z_i} = \int_{Q \cap \partial B_i} (h_{N_q} - h_{N_q \setminus \{i\}}) \phi(q) \tan \theta \, dq - \int_{Q \cap B_i} \frac{2 h_{N_q}^2}{a(b - z_i)^3} \phi(q) \, dq \]

- Move up to see more of Q
- Move down away from neighbors
- Move down to see Q better
Convergence

LaSalle’s invariance principle

• Trajectories are bounded

• $\dot{p}_i = -k \frac{\partial H}{\partial p_i}$ autonomous and locally Lipschitz

• Time derivative of $H$ non-increasing

$$\dot{H} = -\sum_{i=1}^{n} \frac{\partial H}{\partial p_i}^T \frac{\partial H}{\partial p_i} \leq 0$$

Therefore $p_i$ converges to largest invariant set where

$$\dot{H} = 0 \quad \text{which implies} \quad \frac{\partial H}{\partial p_i} \rightarrow 0$$
Harvard Forest
Simulation Results
Five Quadrotors Landing Autonomously
Forest Reconstruction
Multi-robot Systems Research

- Deployment and coverage
- Coordinated agile maneuvers
- Model learning/System ID
- Multi-robot manipulation
- Active estimation with guarantees
- Multi-robot herding
- Neighbor trust adversaries
- Persistent monitoring
- Active 3D vision and control

Multi-Robot Systems Lab Seminar
April 19, 2016
Multi-robot Systems Research

- Deployment and coverage
- Model learning/System ID
- Active estimation with guarantees
- Neighbor trust adversaries
- Active 3D vision and control
- Coordinated agile maneuvers
- Multirobot manipulation
- Multirobot herding
- Persistent monitoring
Coordinated Agile Maneuvers

Dingjiang Zhou
Dealing with Dynamics

\[ \dot{x} = f(x, u(t)) \quad \dot{p} = u(t) \]
Quadrotor Dynamics

Velocity
\[ \dot{v} = q e_3 + \frac{1}{m} R f_z e_3 \]

Angular Velocity
\[ \dot{\omega} = J^{-1} \tau - J^{-1} \Omega J \dot{\omega} \]

Position
\[ \dot{p} = v \]

Orientation
\[ \dot{R} = R \Omega \]

12 dimensional state space, 4 dimensional input space
Nonlinear dynamics
Non-Euclidean, evolve on \( SE(3) \times se(3) \)

Potentially difficult control and planning problem
Trajectory Planning

How can we ensure that $\dot{x} = f(x, u(t))$?

For what control input trajectory $u(t)$ will this be true?
Differential Flatness

A small miracle: quadrotor dynamics are differentially flat

\[ x = \beta(\sigma, \dot{\sigma}, \ldots, \sigma^{(q)}) \]

\[ u = \gamma(\sigma, \dot{\sigma}, \ldots, \sigma^{(q)}) \]

planned flat output trajectory \( \sigma(t) \)

\[ \sigma(t) = (p(t), \psi(t)) \]

State trajectory \( x(t) \)

Input trajectory \( u(t) \)
Related Work

Differential Flatness Theory
- Fliess, Levine, Martin, Rouchon, TCA 1999.

Differential Flatness for Quadrotors
- Mellinger, Kumar, ICRA 2011
- Mellinger, Michael, Kumar, IJRR 2012
- Zhou, Schwager, ICRA 2015.

Control Assuming Single Integrator Agents
- Olfati-Saber and Murray, TAC 2004.
- Jadbabaie, Lin, and Morse, TAC 2003.
- Cortes, Martinez, Karatas, and Bullo, TRO 2004.
- Dimarogonas and Johansson, ICRA 2008.
- Ayanian and Kumar, TRO 2010.
A small miracle: quadrotor dynamics are differentially flat

\[ x = \beta(\sigma, \dot{\sigma}, \ldots, \sigma^{(q)}) \]

\[ \sigma(t) = (p(t), \psi(t)) \]

\[ u = \gamma(\sigma, \dot{\sigma}, \ldots, \sigma^{(q)}) \]

Input trajectory \( u(t) \)

State trajectory \( x(t) \)

planned flat output trajectory \( \sigma(t) \)
Trajectory Following Control

Flat output trajectory

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

Differential Flatness Pipeline

\[ x_d(t), \psi(t) \]

SE(3) controller

\[ u_{ff}(t) \]

\[ u_{fb}(t) \]

\[ x(t), p(t) \]
Vector Field Following

We would like to deal with kinematic points in a vector field.

We have:

\[ \dot{x} = f(x, u) \]
\[ u = \gamma(p, \ldots, \dddot{p}) \]
\[ x_d = \beta(p, \ldots, \dddot{p}) \]
\[ u(p) = \gamma(p, \ldots, \dddot{p}(p)) \]
\[ x_d(p) = \beta(p, \ldots, \dddot{p}(p)) \]

We want:

\[ \dot{p} = v(p) \]
\[ \ddot{p} = \nabla v(p)^T \dot{p} = \nabla v(p)^T v(p) \]
\[ \dddot{p} = \vdots \]
\[ p^{(k)} = \nabla v(p)p^{(k-1)} \]

Zhou, Schwager, ICRA 2014.
Multi-Robot Control

\[ \dot{\mathbf{p}}_i = \mathbf{v}(\mathbf{p}_1, \ldots, \mathbf{p}_n) \]

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

\[ \ddot{\mathbf{m}} = \mathbf{v}(\mathbf{p}_1, \ldots, \mathbf{p}_n) \]

Back to our kinematic multi-robot system.

Zhou, Schwager, ICRA 2014.
Virtual Rigid Bodies Formation Flight
Time: 0.20(s)
Phase: II
Virtual Rigid Body (VRB)

Plan 6DoF trajectory for VRB

\[(p_{\text{VRB}}(t), R_{\text{VRB}}(t))\]

Multi-robot Formation

\[\Pi = \{r_1, \ldots, r_n\}\]

Flat output trajectories

\[p(t)_i = p_{\text{VRB}}(t) + R_{\text{VRB}}(t)r_i\]

\[\psi_i(t) \quad \text{arbitrary}\]

Zhou, Schwager, ICRA 2015.
Formations and Transformations

Formation A
\( \Pi_A \)

Transformation
\( \Phi^B_A \)

Formation B
\( \Pi_B \)

VRB trajectory
VRB for Human Swarm Interface

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

chosen dynamics

VRB

user driven trajectory

scalable swarm trajectory

user input

April 19, 2016

Platform Lab Seminar
Multi-robot Systems Research

- Deployment and coverage
- Coordinated agile maneuvers
- Model learning/System ID
- Multi-robot manipulation
- Active estimation with guarantees
- Multi-robot herding
- Neighbor trust adversaries
- Persistent monitoring
- Active 3D vision and control
My Approach

Theoretical tools

Control theory
Estimation, filtering, machine learning
Optimization

Provable distributed control algorithms

Multi-robot experiments

April 19, 2016
Platform Lab Seminar
Cloud Robotics

- Local low level control
- Centralized computation in the cloud
- Using cell phone and WiFi networks
Active 3D Forest Modeling

UAV Surveillance Mission
Intelligent Controlled Vision

3D Printed Forest Model

Structure from Motion Reconstruction

Eric Cristofalo
Summary

• Multi-robot Systems
• Coverage with cameras
• Virtual Rigid Bodies for agile coordination

Multi-robot systems will fundamentally change the way we interact with the world at large scales.
Thanks!

My Collaborators:
Prof. Daniela Rus, MIT
Prof. Vijay Kumar, UPenn
Prof. Mark Friedl, BU
Dr. Josh Gray, BU
Dr. Brian Julian, Google

My Funding Sources:
Summary

• Multi-robot Systems
• Coverage with cameras
• Virtual Rigid Bodies for agile coordination

Multi-robot systems will fundamentally change the way we interact with the world at large scales