1 Literature Review

I need to start writing a comprehensive literature review

1. The majority of efforts focused on modelling oil spills and plumes are focused on the long-term fate of the oil spill, ei, how much area will the spill cover, which wildlife populations it will impact, which harbors to focus clean-up efforts in.

2. The concern with this project was not perfectly capturing the long-term image of the plume, but generating a plume with realistic characteristics in the instantaneous snap-shot.

3. This is relevant to concerns of sensing plumes, high-dosage exposure limits of chemicals in the plume, and of course robotic tracking of plumes.

4. We cannot use the classic and well established Eulerian method of generating the plume. (Jones 1982)(Elkinton 1982)

5. Experimental analysis evinces that actual plumes can have instantaneous concentrations of up to 30 times higher than those predicted by the Eulerian method (Jones?). Also, the plume can have an intermittency of up to 85%.

6. There are the Eulerian models, and the Lagrangian models. Eulerian methods model plumes and fluid flows as smooth gradients. They focus of using finite difference methods to solve a set of PDEs over a defined grid and interpolatin results between gridpoints. The Lagrangian method models the plume as composed of many (if not infinite) molecules, all moving irrespective of the other molecules around it. This method using no grid lines so can theoretically produce results with a much finer resolution.

7. There are many aspects to modelling plumes. advection, spreading, evaporation, dispersion, eumulsification, interaction with shorelines (Reed et al,1999). Most preminent and relevant of these 6 are advection and dispersion.

8. our method is as modelling a passive scalar in a fluid flow. To do this we use a Lagrangian method of modelling the plume as structure composite of many individual particles

9. The diffusion portion of the plume is modelling with a random walk. The advection portion is modelled as eat molecule moving in the direction of the fluid flow (this is kind of hard to explain how simple the advection is while still sounding sophisticated.)

2 Notes

2.1 Webster et al 2001

"On the usefulness of bilateral comparison to tracking turbulent chemical odor plumes" D.R. Webster, S. Rahman, and L.P. Dasi. *Limnol. Oceanogr.* 46(5), 2001, 1048-1053

Most impressive about this article are all of the pretty looking graphs.

2.2 Reed, 1999

"Oil Spill Modeling towards the Close of the 20th Century: Overview of the State of the Art" by Mark Reed et al (1999)

Aspects of oil spill modeling - advection, spreading, evaporation, dispersion, emulsification, interaction with ice and shorelines. The 2 that are most prominent and that for this study we need to worry about are advection and dispersion

2.3 ASCE

"State-of-the-Art Review of Modeling Transport and Fate of of Oil Spills" by ASCE Task Committee on Modeling of Oil Spills of the Water Resources Engineering Division Journal of Hydraulic Engineering Vol 122 No 11 November 1996 pp 594-609

The dominating forces of oil slick movement at the surface are wind, wave, and current (ASCE Review). Wind moves and current move the oil. current moves with near 100%of its knottage, and wind will move oil with something like 10% of its knottage. Waves mostly break the oil apart

2.4 Nagheeby, 2010

Numerical modeling of two-phase fluid flow and oil slick transport in estuarine water. M Nagheeby, M Kolahdoozan. Department of Civil and Environmental Engineering, 2010

This talks a lot about using the Lagrangian model for the oil transport through an Eulerian calculated fluid flow. This corresponds directly to our navier-stokes fluid flow and the random walk

2.5 Scase, Hewitt

'Unsteady turbulent plume models' by M.M. Scase and R.E. Hewitt.

This article talks about "four existing integral models of unsteady turbulent plumes" and then presents another model. These models are good. this is a great survey of plume models.

2.6 Elkinton et al, 1984

"Evaluation of Time-Average Dispersion Models for Estimating Pheromone Concentration in a Deciduous Forest" by J.S. Elkinton, Carde, Mason an experiment. predictions made using a Gaussian-type model. from abstract " Thus the models estimate pheromone concentrations for time intervals appreciably longer than required for behavioral response"

There have been objections of using the time-averaged model of plume to model behavioral reactions that occur over short time intervals - paraphrased. look at Mason, Aylor et al, Miksad and Kittredge, Murlis and Jones

2.7 C.D. Jones, 1984

Jones, C.D.: 1983, On the structure of instantaneous plumes in the atmosphere. Journal of Hazardous Material., 7, 87-112

2.8 Developed models

the following website talks about some models that use the puff model that are already out there. Most notable of these is the CalPuff software. http://elte.prompt.hu/sites/default/files/tananyag

3 Intro

The goal of this plume simulation is to generate a model that will suffice in testing and development of plume tracking algorithms. Standard plume modelling methods are potentially unsuitable for this algorithm development (Jones 1982, Elkinton 1983). A satisfactory plume simulation has to resemble a plume both instantaneously and over a time-averaged view. To accomplish this, the plume simulation presented here uses a random walk model (Kinzelbach 1990, Kitandis 1994). In the limit, the random walk simulates a Gaussian advection-diffusion plume, but when tuned down, with the number of molecules being far from the limit, shows high intermittency and erraticness seen in (Jones 1982 and Elkinton 1983).

3.1 Requirements:

The modelled needed for our testing does not have to accurately show long term movement of the plume perfectly, this is not a concern. The concern is with the short-term view of the plume on a scale of meters, not miles. For this, waves will play a large role in the movement

1. Instantaneously, the plume will have a high intermittency. For the majority of time (upwards of 80%) there is a material concentration of zero at any given point. (Jones1983).

2. The long-term, time-averaged plume, will resemble a plume model such as the model used in the original simulation.

3. The simulation will be able to retrieve concentration, gradient, divergence of gradient, and flow vector at all locations (x, y) and all time t.

4 Governing Equations

4.1 Random Walk

The random walk equation (1) has been derived by Kinzelbach (1990) from the transport equation of an ideal tracer. In the limit, as both $\Delta t \to 0$ and number of molecules $\to \infty$, the concentration of equation (1) will be equivalent to the advection-diffusion equations solved in the original environment model.

$$x_{t+\Delta t} = x_t + u\Delta t + Z(2\alpha_L u \Delta t)^{1/2} \tag{1}$$

Here u represents the fluid velocity vector, Z is a normally distributed random variable centered at 0 and with a variance of 1, α_L is the longitudinal dispersivity, or the diffusion coefficient and t is time.

4.2 Concentration

The concentration at any point x is then the total number of molecules m within a small radius $r$of the point x, divided by a normalizing factor based on the area of observation and the water and molecule properties.

$$c(x,t) = card(\{m : ||m - x|| < r\})$$

5 LCM Interface

The environment thread consists of 2 main functionalities, environmental update and data retrieval. The lcm channels to cue these are respectively named *envUpdate* and *envRetrieve.*

envUpdate will push the plume forward 1 time-step, *dt*, and return a message on channel *envUpdateConfirm* when it is finished. It does not matter what is in the message that is sent in either direction, they are only used as cues. *envRetrieve* takes a point in Cartesian coordinates and returns the estimated concentration, gradient of concentration, the normal vector to the gradient, divergence of the gradient, and the fluid flow velocity.

Input	Hidden	Output	
envRetrieve			*dataReturn*
	C(x,y)	Concentration	U0
	C(x+dx, y)	Gradient	DU
(x,y)	C(x-dx, y)	Divergence	D2U0
	C(x, y+dy)	Normal to gradient	DU_p
	C(x, y-dy)	Fluid flow vector	V0

The current implementations of calculations for the gradient and the divergence require five points of concentration to be surveyed. Currently, we use points up, down, left, and right of the robot.

6 Operation

First, download the code from the github repository:

`https://github.com/sauln/plumeSimulation`

6.1 Simulation with lcm

To run the plume in the loop with the robot simulation, you must run 3 files:

1. *plumeSim/plumeSimWlcm.py*,
2. *roboSim/controlThread.py* and
3. *roboSim/simulationThread.py.*

plumeSim/plumeSimWlcm.py must be run within *ipython –pylab* for the visualizations to work properly. The simulation is currently setup to use the single integrator robot model as this was the last stable version I have on my machine. Plugging in the new model should be as simple as changing the lcm channel names.

6.2 Helpful functions

To run the plume simulation by itself, you can run *plumeSim/plumeSim.py.* This file is currently setup to run the basic plume model, but there are a number of functions in *plumeSime/plumeExperiment.py* that explore different random walk equations and calculate various statistics and interesting plots about the plumes. Most important are the functions that generates statistics about the intermittency of the plume and the time-averaged probability distribution function at various points down the plume.

6.3 Fluid flows

There are 2 different fluid flows that you can use. There is also a lot of room to add new fluid flows. When the simulation initiates the plume, it defines which flow to use for the calculations. The two that are built in now are named 'simple' and 'mit'. The 'mit' is the same fluid flow that was used in the original plume simulation. The 'simple' flow is a constant downstream flow.

7 Initial Observations

In my experiments, it is very clear that something is wrong. I found a few bugs with how the gradient and divergence of gradient were calculated, but now there are a few test functions that show the calculations are correct now.

The main observation is it seems that when the robot looses touch with the plume (all sensors read zero), the robot continues in the previous direction it was moving. Intuitively, I would expect it to turn around at that point in an attempt to re-establish contact.

8 Todo

1. Calibrate the plume. Currently the calibration is only estimated. This includes the concentration normalizing factor, how many molecules to emit, and the relative time frame.

2. Move the gradient and divergence calculations to the *estimator* side of the simulation.

Plume Simulation

Nathaniel Saul

July 21, 2014

8.1 Outline for Lab

1. **Intro**

   In the long-term will resemble a time-averaged plume model such as the model used in the original simulation, with definitively Gaussian distribution.

   Instantaneously, the plume will have an intermittency above a certain threshold of near 80% (Jones1983).

   The goal of this environment simulation is to generate a plume that, in the long term, But also in the short term will

2. **Governing Equations**

   The governing equations are described in Kinzelbach (1990).

   $$x_{t+\Delta t} = x_t + u\Delta t + Z(2\alpha_L u \Delta t)^{1/2}$$

   u is the velocity vector. Z is a normally distributed random variable centered at 0 and variance 1. α_L is the longitudinal dispersivity. t is time. In the limit as both $\Delta t \to 0$ and number of molecules $\to \infty$, this equation becomes the

   The concentration at any point is then the sum of all molecules within a small perimeter of the point in question, divided by a normalizing factor based on the water and molecule properties.

   $$c(x,t) = \frac{\Delta M}{A n_e} n(x - 0.5\Delta x, x + 0.5\Delta x, t)$$

   A is the cross-sectional area, n_e is the porosity effective.

   The vectors x and u are both 2 dimensional.

3. **Simulation Interface** The environment thread consists of 2 main functionality, environment update and data retrieve. The lcm channels are respectively named *envUpdate* and *envRetrieve*.

   envUpdate will push the plume forward 1 timestep and return a message on channel *envUpdateConfirm* when it is finished. *envRetrieve* takes a point in Cartesian coordinates and returns the estimated concentration, gradient of concentration, the normal vector to the gradient, divergence of the gradient, and the fluid flow velocity.

Input	Hidden	Output	
envRetrieve			*dataReturn*
(x,y)	C(x,y)	Concentration	U0
	C(x+dx, y)	Gradient	DU
	C(x-dx, y)	Divergence	D2U0
	C(x, y+dy)	Normal to gradient	DU_p
	C(x, y-dy)	Fluid flow vector	V0

The current implementations of calculations for the gradient and the divergence require five points of concentration to be surveyed. Currently, we use points up, down, left, and right of the robot. All a unit of dx away.

4. **Operate** To run the simulation with the plume, download the code from the github repository. There are 3 main functions. plumeSim/plumeSimWlcm.py runs the environment and plume simulation. In the roboSim directory there is roboSim/controlThread.py and roboSim/simulationThread.py. All 3 must be run at the same time and the simulationThread.py must be started last.

5. **Observations** When the robot looses touch with the plume (all sensors come up zero), the robot continues in the previous direction, instead, it needs to turn around. (not exactly around, but continue downstream and back towards the plume.

6. **Todo**

   (a) Calibrate the plume. Currently the calibration is only estimated.

   (b) Move the gradient and divergence calculations to the *estimator* side.

8.2 Outline

1. **Problem Statement:** Current models are advection diffusion doesn't capture the salient dynamics of a plume. need to develop new models that can capture the dynamics of the plume accurately and fast and that can be used in the testing and development of the robust control algorithms. Will robotic plume tracking algorithms developed using advection diffusion models hold up to more complicated plume models?

2. **Survey of plume models:**

   (a) advection and diffusion:

   (b) difference between instantaneous plume and time-averaged plume

   (c) turbulent plumes - stochastic plume model

   (d) time-averaged and gaussian plumes

   (e) puff model

   (f) random walk

3. **Model Requirements:**

   (a) capture the salient dynamics

      i. intermittent (spontaneous)

      ii. gaussian distribution

   (b) be computationally fast - thus we really don't need to do a perfect simulation...

   (c) fit into our robot simulator

4. **Our plume model: a random walk**

   (a) explain random walk simply - The most simple random walk, in 1 dimension, every tick, the molecule can either move left, or move right. This random walk is in 2 dimensions, and moves in any direction.

   (b) a modified random walk that is the Gaussian plume model in the limit (of both molecules and grid-size) (there must be a source that proves this)

   (c) when the random walk is far from the limit, the puffs are sporadic

   (d) diffusion and advection

   (e) downfalls: not 'clumpy' like the turbulent models depict

   (f) awesomeness: implementation is EASY and fast, I think it is good enough.

9 Intro/Problem Statement:

Points to talk about:

1. many control algorithms are very robust, and an accurate model is not necessary, other algorithms are very brittle and rely on the model being precise

2. the algorithm developed by [**?**] was designed using an advection-diffusion plume model and was tested with only an advection-diffusion model.

3. There is a possibility that the algorithm will break when tested a more complex plume model.

4. The goal of this paper is to develop another plume model for the testing of the control algorithm.

5. I remember reading that using an advection plume model could be bad when developing algorithms (cite this article as some motivation for this

Current models are advection diffusion doesn't capture the salient dynamics of a plume. need to develop new models that can capture the dynamics of the plume accurately and fast and that can be used in the testing and development of the robust control algorithms. Will robotic plume tracking algorithms developed using advection diffusion models hold up to more complicated plume models?

10 Plume Modeling

10.1 basics

1. advection
2. diffusion
3. time-averaged
4. instantaneous —

10.2 different models

1. time-averaged and gaussian plumes
2. turbulent plumes - stochastic plume model –ito-fokker-plank
3. puff model
4. random walk

There are a few governing equations named after a few different people. One that seems to pop up a bunch in literature from the 70s and 80s is Taylor Diffusion. There is also the surface tension theory. Also, turbulent diffusion theory. (see Murray- turbulent diffusion). There is the Taylor approach, and the Fikkian approach. There is the stochastic differential equation approach that seems to be centered on Ito's equation and sometimes referred to as Ito-Fikkian-Plank equation.

So we have advection-diffusion model. and there are lots of kinds of diffusion.

the oil diffusing into water equation that pops up all the time is(see : Crude oil dissolution in saline water. Salah E.M Hamam, Mohamed F Hamoda et al 1988)

$$\frac{\delta C}{\delta t} = D\frac{\delta^2 C}{\delta x^2}$$

11 Model Requirements

This is when we talk about how the time-averaged model is not good enough (there are sources I can cite) *as evidenced in papers 1, 2, and 3, the advection-diffusion models do not capture the instantaneous dynamics of plume..*

there are well studied and documented plume models that are pretty accurate (each with their own downfalls of course). For this problem though, we do not need something perfect, we just need something erratic (at least that's what I keep telling myself to justify not solving any differential equations).

The plume has to satisfy a number of conditions.

We want the time averaged plume to resemble the solution to the advection-diffusion equation

$$\frac{\delta c}{\delta t} + u\frac{\delta c}{\delta x} = D\frac{\delta^2 c}{\delta x^2}$$

We want the inputs to be -

1. the long-term behaviour of the plume must replicate the Gaussian advection-diffusion model.

2. the inputs must be easy to input, the flow field, the desired plume parameters: choose a diffusion model- looks like the turbulent diffusion theory or other fits the data the best

3. we want the instantaneous peak concentrations to be... still haven't figured out a realistic value that is backed up by data

Concentration is frequently represented as a probability density plume with

$$\bar{C}(x,y,z) = \frac{Q}{2\pi S_y S_z \bar{u}} exp\left(-\left(\frac{y^2}{2S_y^2} + \frac{z^2}{2S_z^2}\right)\right)$$

12 Our plume model: a Random Walk

12.1 Random Walk

explain the basics of a random walk - 1 dimensional a little bit of details about what a random walk is explain random walk simply - The most simple random walk, in 1 dimension, every tick, the molecule can either move left, or move right. This random walk is in 2 dimensions, and moves in any direction. a modified random walk that is the Gaussian plume model in the limit (of both molecules and grid-size) (there must be a source that proves this)

12.2 Modified random walk for plume

Explain how the random walk can be used to model advection and diffusion when the random walk is far from the limit, the puffs are sporadic. **downfalls:** not 'clumpy' like the turbulent models depict

awesomeness: implementation is EASY and fast, I think it is good enough.

In the limit, the random walk is the gaussian plume distribution.

It looks like there is a few other equations that we can use. Namely, Kitanidis

12.3 Our random walk plume

Explain the implementation of this plume Kitanidis' naive equation is:

$$x(t + \delta t) = x(t) + v(x)\delta t + (2D\delta t)^{1/2}\epsilon$$

D is the diffusion coefficient when the diffusion coefficient does not vary in space. This seems to suffice for our needs. ϵ is a normal random variable centered at 0 with a variance of 1.

Each puff will do its own random walk according to the equations 2.

The following model for a random walk fails when either v_x or v_y is zero. Though this doesn't happen in real-life, for the simulation purposes, I need a different model.

$$\begin{aligned} x_{i+1} &= x_i + (r(c, \sigma)v_x + r(0, \sigma)n_x)dt \\ y_{i+1} &= y_i + (r(c, \sigma)v_y + r(0, \sigma)n_y)dt \end{aligned} \tag{2}$$

where x_i is the current x coordinate, $r(c, \sigma)$ is a random function with center c and standard deviation σ, v_x is the fluid flow in the x direction, and dt is a scaling component.

σ represents the diffusion component of the plume and v_x represents the advection component of the plume.

12.4 How to find concentration

Explain how concentration is calculated

The concentration of the plume at a location (x, y) can be defined as the sum of each puff at that point.

$$\sum_{p_x, p_y} \delta_r, \qquad \delta_r = \begin{cases} 1 & : p_x \in [x - d, x + d] \\ 0 & : p_x \notin [x - d, x + d] \end{cases} \tag{3}$$