Characterizing and Compensating for Error in Horizontal Range Calculations Induced by Variations in the Sound Speed Profile

Christopher Dolan

Introduction

- 1st year JP student
- Submarine Officer
- 22.5 years of US Naval experience
- Most recently Navigator/Operations Officer on USS KEY WEST

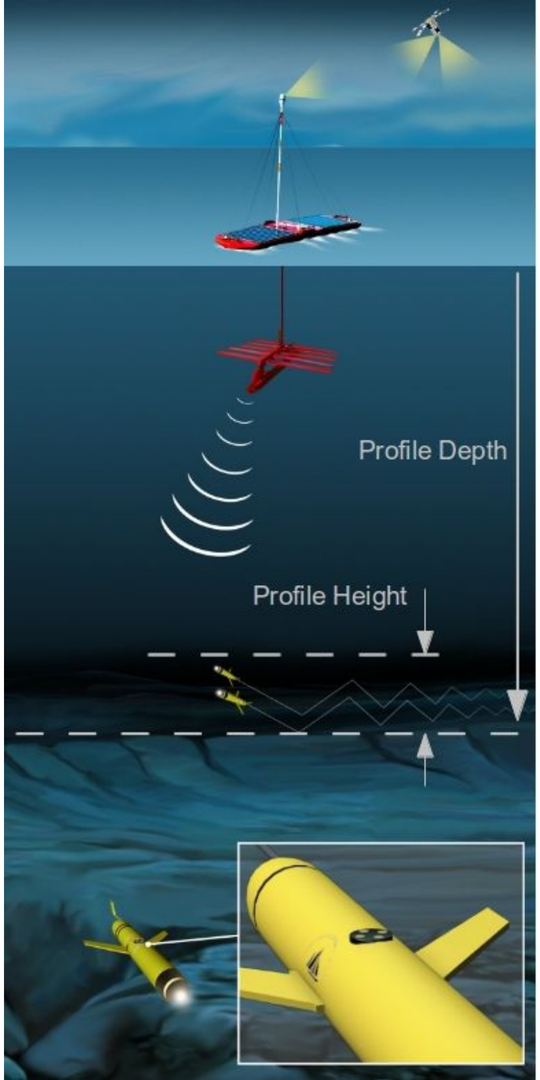
Research

Dr. Mike Jakuba and Dr. Erin Fischell

OWTT iUSBL system

Goals:

1. Accurate position estimates of measurements to be obtained in the deep ocean interior
2. Precisely navigated observations around sea floor topography
3. New observation strategies that will increase the density and cadence of observations in the ocean's interior
4. Improved water current measurements from mobile robots through better estimates of the robots' through-water velocity



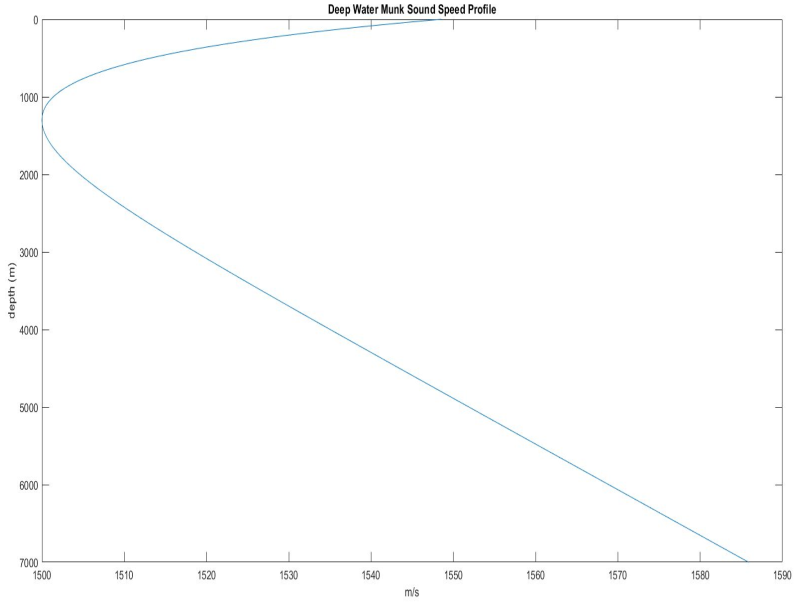
Characterizing the Problem

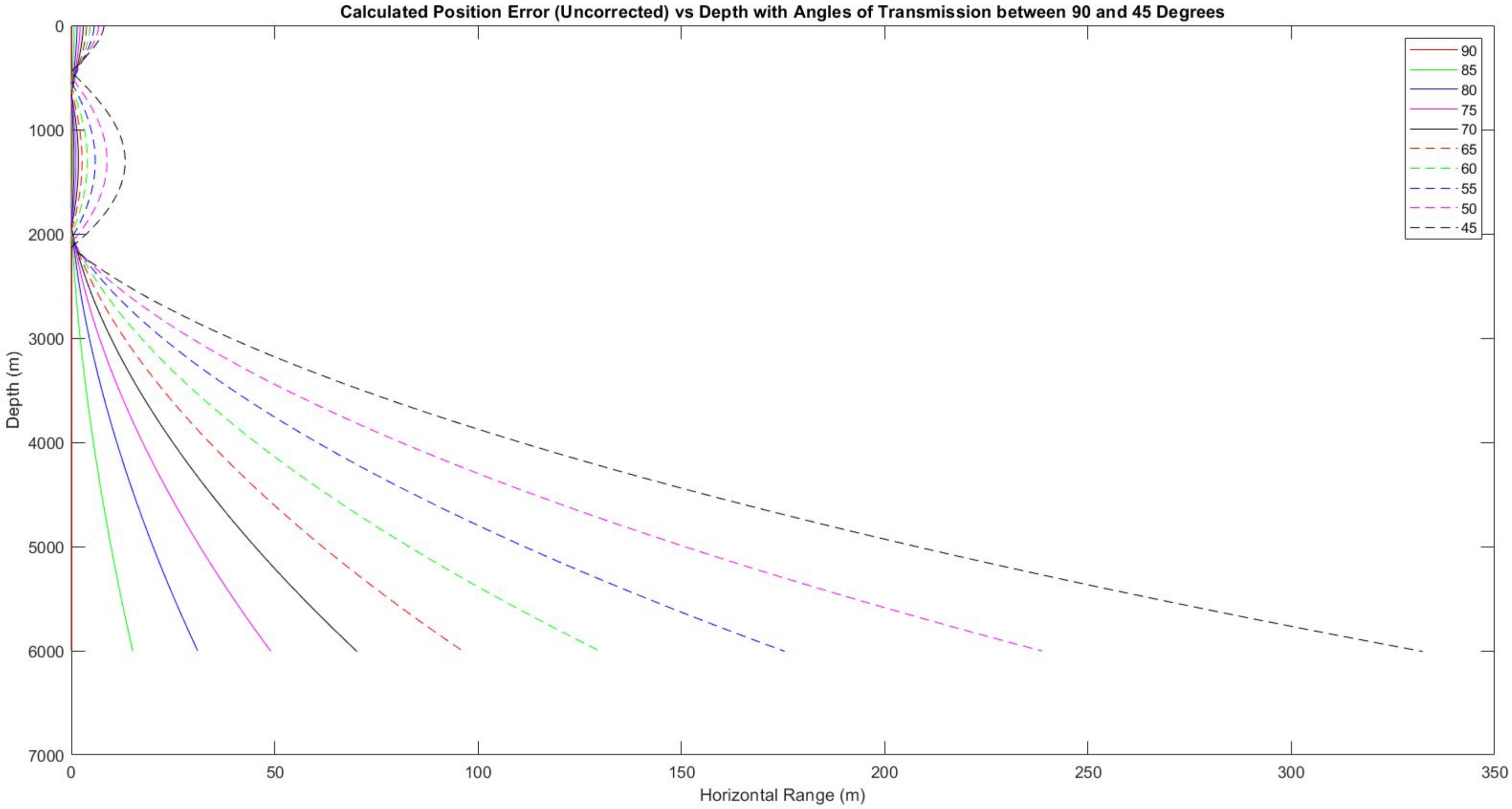
Parameters

- Sound source at 8m
- Glider rated to 6000m
- 10 kHz array
- Deep water Munk profile

Constraints

- Array geometry not optimized for application
- Angle resolution limited to 45 degrees from vertical





Calculated Position Error (Uncorrected) vs Depth with Angles of Transmission between 90 and 45 Degrees
Depth (m)
Horizontal Range (m)
0
1000
2000
3000
4000
5000
6000
7000
0
50
100
150
200
250
300
350
90
85
80
75
70
65
60
55
50
45

Compensating for the Error

Available measurements:

- Transmission travel time
  - Chip set atomic clock
- Glider depth
  - Pressure transducer
- Incident angle
  - Four element planar array

Stored data:

- Sound speed profile
  - Sound speed at discrete depths

Compensating for the Error

Calculate transmission angle

Use ray trace code to determine horizontal position

```
theta_surf =
acos((cmunkdeep(1,2)/cmunkdeep(1+round(z_glider),2))*cos(theta
_glider));
```

Compensating for the Error

Uncertainty of angular measurement

Introduce an error range

Calculate a fan of rays

```
theta_error = (3*pi/180)

theta_surf_plus =
acos((cmunkdeep(1,2)/cmunkdeep(1+round(z_glider),2))*cos(theta_glider +
theta_error));

theta_surf_minus =
acos((cmunkdeep(1,2)/cmunkdeep(1+round(z_glider),2))*cos(theta_glider -
theta_error));

thetas = theta_surf_minus:(theta_error*2)/30:theta_surf_plus;
```

Compensating for the Error

Choose appropriate array index based on transmission travel time

Determine difference between predicted and measured depth

Calculate a root-sum-square to characterize the total error

```
tau_compare1 = abs(tau_glider-tau_out(ii,jj))/tau_glider;

tau_compare2 = abs(tau_glider-tau_out(ii,jj-1))/tau_glider;

    if tau_compare1 < tau_compare2

         z_compare = abs(z_glider-zray_out(ii,jj))/z_glider;

         r_compare(ii,1) = rray_out(ii,jj);

         rss_compare(ii,1) = sqrt(tau_compare1^2 + z_compare^2);

    else

        z_compare = abs(z_glider-zray_out(ii,jj-1))/z_glider;

        r_compare(ii,1) = rray_out(ii,jj-1);

        rss_compare(ii,1) = sqrt(tau_compare2^2 + z_compare^2);

    end
```

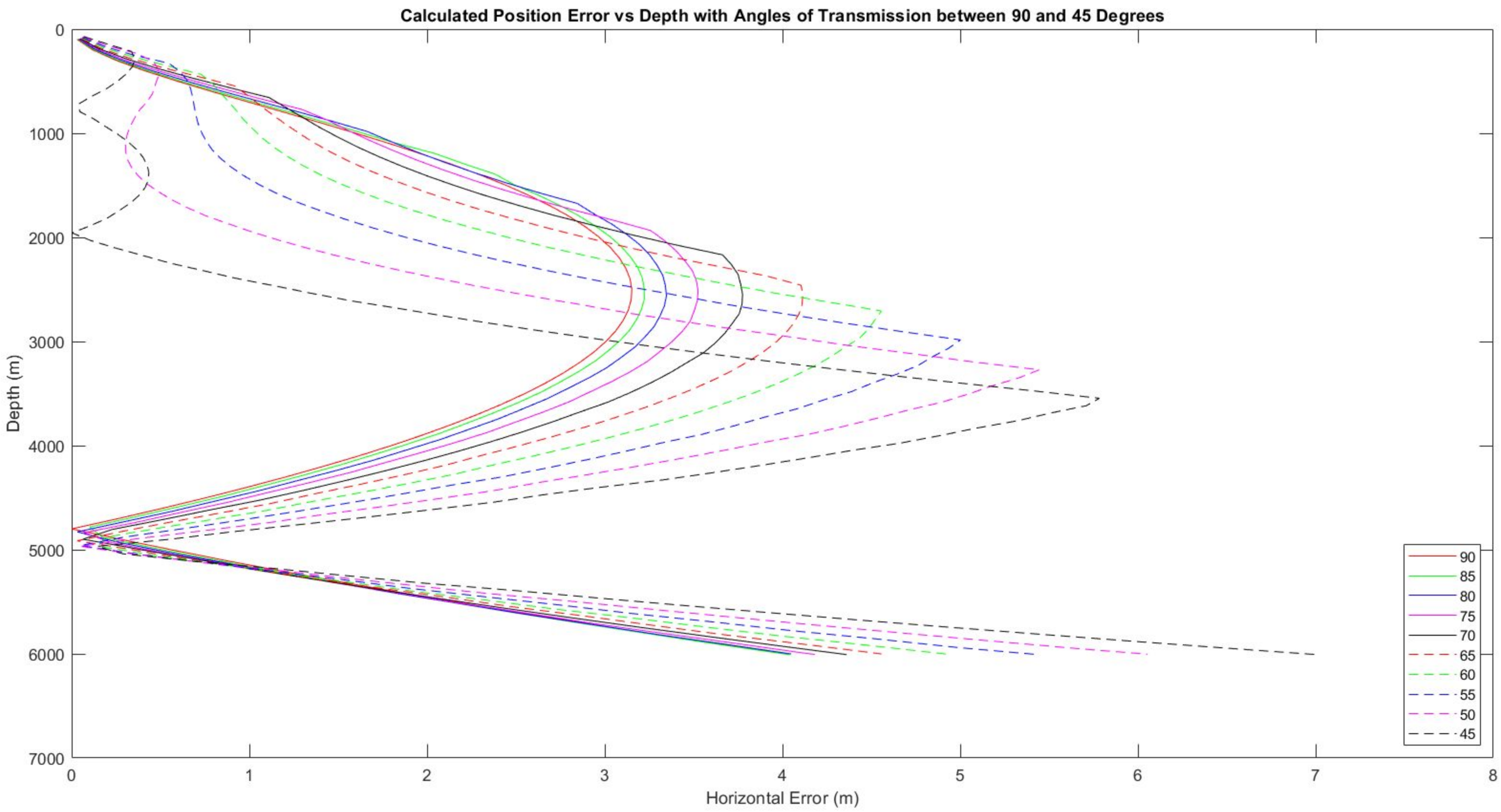
Compensating for the Error

Minimize the total error and output the horizontal range at that index

```
[~,ind] = sort(rss_compare, 'ascend');

r_compare_sort = r_compare(ind,:);

r_out = r_compare_sort(1,1);
```



Calculated Position Error vs Depth with Angles of Transmission between 90 and 45 Degrees
Depth (m)
Horizontal Error (m)
0
1000
2000
3000
4000
5000
6000
7000
0
1
2
3
4
5
6
7
8
90
85
80
75
70
65
60
55
50
45

Follow on Work

- Post processing of previously recorded data
- Determine at what angles the algorithm is useful
- Improving error band
  - model noise in angular measurement to include directionality
- Integrate with command and control
  - maintain appropriate depth/angle combination to limit error to a predetermined amount
- Calculate optimal array geometry and characterize effect of current array

Questions?