

Lab 3 – Motion Model II

A report must be submitted for this lab. It is due on Oct. 22nd. The report should include 1) a one paragraph abstract that gives an overview of the lab, 2) a brief description of the experiments carried out, 3) results including plots with all axis labeled and units provided, and 4) a discussion section that explains how the results validate theory.

1 – Create a belief update function

In Matlab, create function `[Mu, Sigma] = Algorithm_Belief_Motion_Model(Mu, U, Sigma, ts)`. The belief state is defined by (μ, Σ) .

```
[Mu, Sigma] = Algorithm_Belief_Motion_Model(Mu, U, Sigma, ts);
Mu = g(U, Mu);
Sigma = G*Sigma*G' + V*R*V';
```

Where $g()$ is obtained with your `Propagate_State` function. You should create another new function `G = GetG(X,ts)` that calculates

$$G_t = \frac{dg(u_t, \mu_{t-1})}{dx_{t-1}}$$

Also create another new function `V = GetV(ts)`

$$V_t = \frac{dg(u_t, \mu_{t-1})}{du_t}$$

Note that the covariance matrix R can take on the form $\text{diag}[(\alpha_x |f_x|)^2 (\alpha_z |f_z|)^2 (\alpha_N |f_N|)^2]$, in which the standard deviations are some linear functions of magnitude of the force $U = [f_x \ f_z \ f_N]$. The coefficients α_x , α_z , and α_N are experimentally determined, (for this lab use some arbitrary value).

2 – Create Motion Model Sampling function

In Matlab, create function `[X] = Algorithm_Sample_Motion_Model(X, U, R, ts)`

```
X = Algorithm_Sample_Motion_Model(X, U, R, ts);
sigmaU = [sqrt(R(1,1)) sqrt(R(2,2)) sqrt(R(3,3))];
U = U + normalDistribution(0,sigmaU);
X = g(X', U);
```

Note that the standard deviation of control inputs can take on the form $\text{sigmaU} = [\alpha_x |f_x| \ \alpha_z |f_z| \ \alpha_N |f_N|]$, in which the standard deviations are some linear functions of magnitude of the force $U = [f_x \ f_z \ f_N]$. The coefficients α_x , α_z , and α_N are experimentally determined, (for this lab use some arbitrary value). You must determine how to get a normally distributed random number. (See table 5.4 of text or use `randn`).

3- Match sample function output with the covariance

Compare the two functions above. For some nominal control sequence (i.e. that drives the ROV straight with some constant force held for some set amount of time), generate a large number of sample motions and resulting position states (using the function `Algorithm_Sample_Motion_Model`). Then, calculate the mean position (x,y) and variance in the x and y directions of the resulting position states. Compare with the values obtained from the `Algorithm_Belief_Motion_Model` function.

Finally, plot the initial position state, the resulting position states, and the mean position state. If you want to plot the error ellipse, you can find the function online by googling “Matlab error_ellipse”. For example, an odometry motion model might produce the following results:

