1 Ideas

1.1 Training

The considerations for training are:

- What/when I'm training
- what the goal of that training is
- How to get training samples
- LSTM fixed/variable length
- Inputs to LSTM(physical specifications, previous outputs, etc.)
-
- Train transition model with LQR, hope that the model learns how to linearize given any particular observation

2 Assumptions

We have the linear dynamical system:

$$\begin{array}{rcll} l_{t+1} &=& A_t l_t + B_t u_t + g_t \epsilon_t & \epsilon_t \sim \mathcal{N}(0,1) \\ Z_t &=& C_t l_t + D_t u_t + \sigma_t \varepsilon_t & \varepsilon_t \sim \mathcal{N}(0,1) \end{array}$$

where $A \in \mathbb{R}^{m\times n}, B \in \mathbb{R}^{m\times r}, C \in \mathbb{R}^{z\times m}, D \in \mathbb{R}^{z\times r}, l \in \mathbb{R}^m, u \in \mathbb{R}^r, Z \in \mathbb{R}^z, \epsilon \in \mathbb{R}^m, \varepsilon \in \mathbb{R}^z$ However, it simplifies if, as in our situation, C = I, and D = 0.

3 Potential mistakes

In computing the likelihood: $\mathcal{N}(z_t|\mu_t, \Sigma_t)$ Compare the paper's calculation:

$$\begin{array}{rcl} \mu_1 &=& a_1^T \mu_0 \\ \Sigma_1 &=& a_1^T \Sigma_0 a_1 + \sigma_1^2 \\ \mu_t &=& a_t F_t f_{t-1} \\ \Sigma_t &=& a_t^T (F_t S_t F_t^T + g_t g_t^T) a_t + \sigma_t^2 \end{array}$$

with my control version:

```
mu_1 = tf.matmul(trans(self.C)[0], self.mu_0)
mu = tf.matmul(C, tf.add(tf.matmul(A,l_filtered), tf.matmul(B,u)))

Sigma_1 = tf.matmul(tf.matmul(trans(self.C)[0],
    tf.linalg.diag(tf.squeeze(self.Sigma_0))),trans(self.C)[0],
    transpose_b=True)+tf.square(trans(self.sigma)[0])
```

```
temp = tf.matmul(tf.matmul(A, P_filtered), A, transpose_b=True) +
    tf.matmul(g, g, transpose_b=True)
Sigma = tf.matmul(tf.matmul(C, temp), C, transpose_b=True) +
    tf.square(sigma)
```