Prof. Kirstin Hagelskjær Petersen
kirstin@cornell.edu

Fast Robots

Lab 10 - Highlights

Robert Whitney

Lab 10 - Highlights

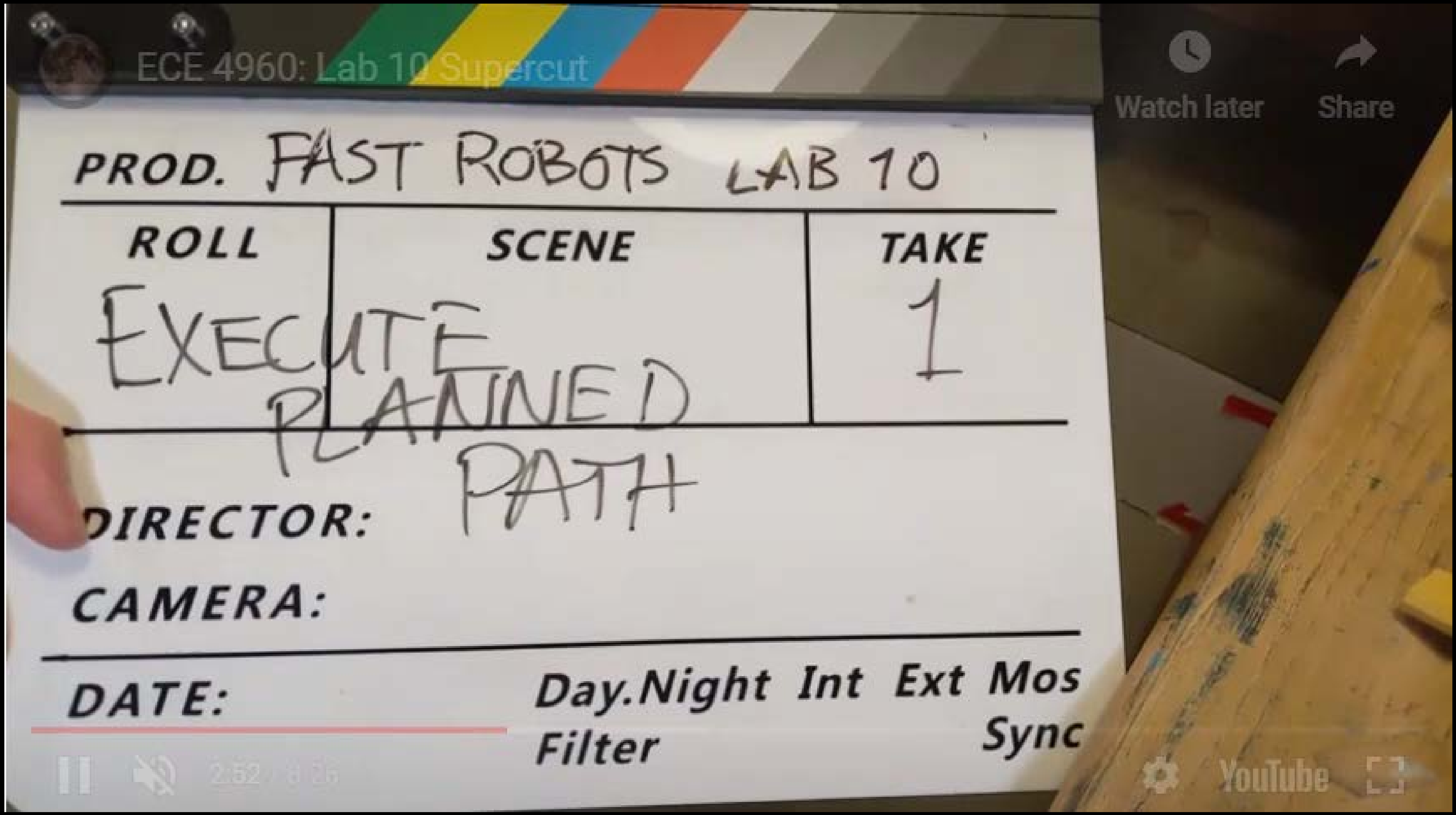
Jade Pinkenburg

Lab 10 - Highlights

Kathleen Wang

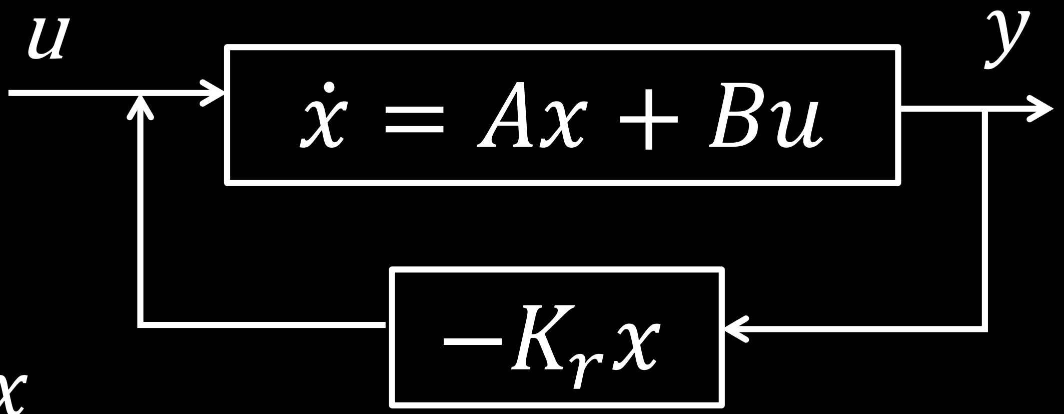
Lab 10 - Highlights

Greg Kaiser



Control Recap

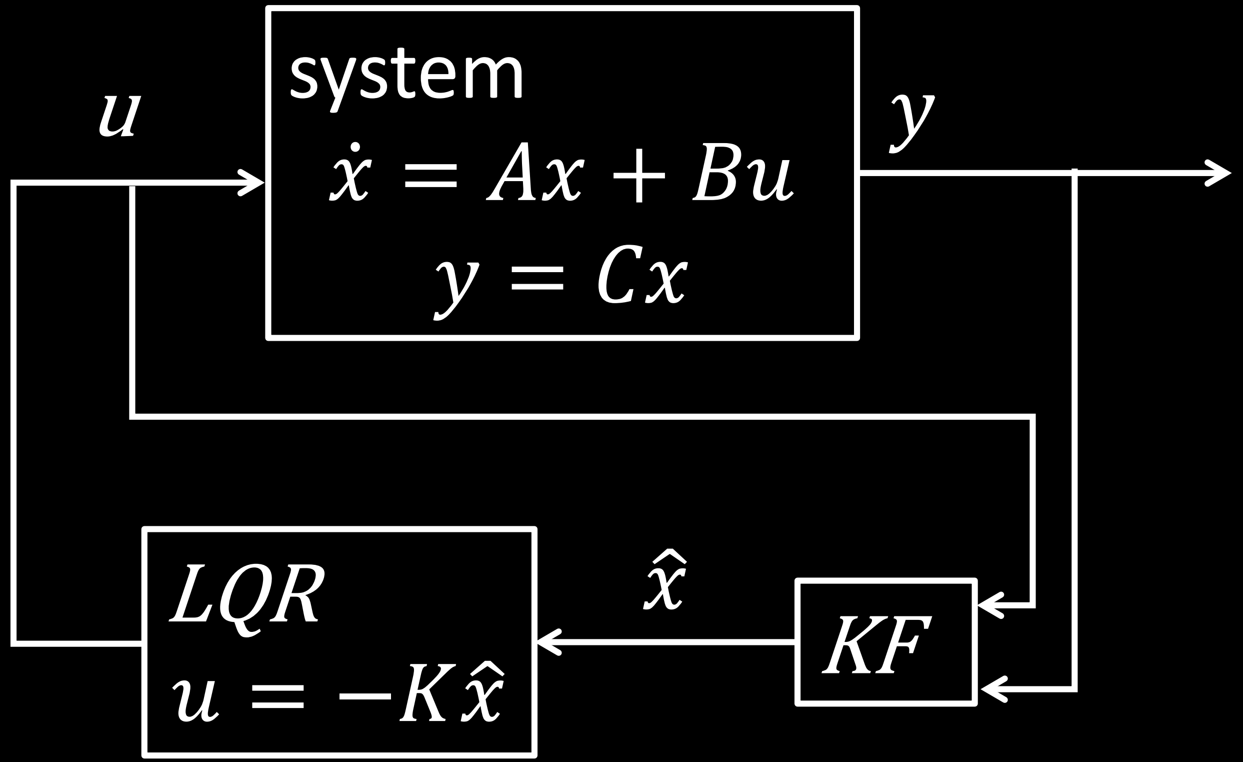
- Linear systems: $\dot{x} = Ax + Bu, \quad x \epsilon \mathbb{R}^n$
- Linearizing nonlinear systems
  - Jacobians, fixed points
- Eigenvectors/eigenvalues and stability
- Controllability
  - rank(ctrb(A,B)) = n
  - Reachability
  - Controllability Gramian
  - Pole placement: $\dot{x} = (A - BK_r)x$
    - Linear Quadratic Regulator: $u = -K_r x$
      - *Observability*



Full State Feedback

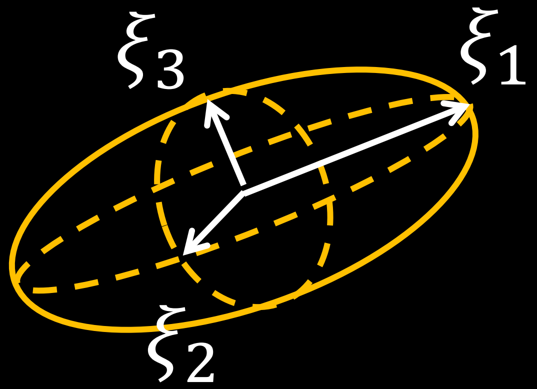
- Controllability
  - Can we steer the system anywhere given some control input u?
- Observability
  - Can we estimate any state x, from a time series of measurements y(t)?

$$\dot{x} = Ax + Bu, x \epsilon \mathbb{R}^n$$
$$u = -Kx$$
$$\dot{x} = (A - BK)x$$



Observability

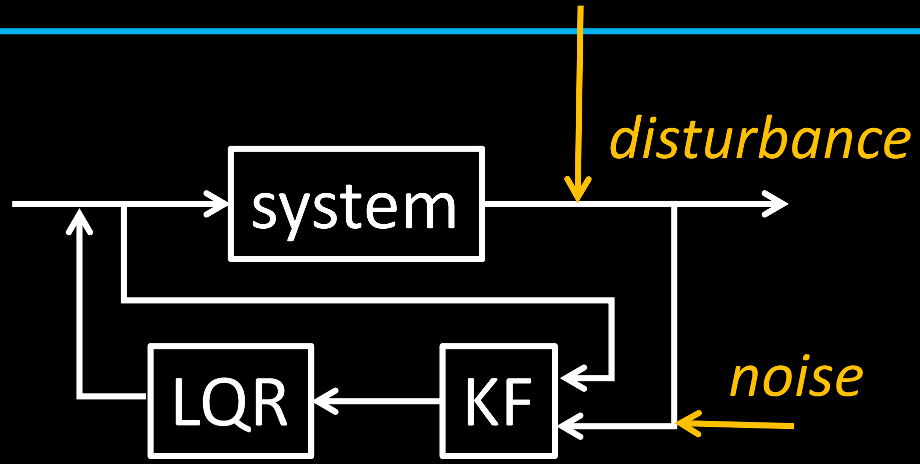
- $\mathcal{O} = \begin{bmatrix} C \\ CA \\ CA^2 \\ \dots \\ CA^{n-1} \end{bmatrix}$



1. Observable iff rank($\mathcal{O}$) = n
   - `>>rank(obsv(A,C))`
2. Iff a system Is observable, we can estimate x from y
   - Observability Gramian
     - `>>[U,Σ, V]=svd(`$\mathcal{O}$`)`

$$\dot{x} = Ax + Bu + d \qquad x\epsilon\mathbb{R}^n$$
$$y = Cx + n \qquad u\epsilon\mathbb{R}^q$$
$$y\epsilon\mathbb{R}^p$$

- Controllability
- $\mathbb{C} = [B \quad AB \quad A^2B \quad \dots \quad A^{n-1}B]$
- >>ctrb(A,B)
- Reachability



Bayes Filter – Kalman Filter

- Incorporate uncertainty to get better estimates based on inputs and observations
- Kalman filter is based on the same idea
  - Assume that posterior and prior belief are Gaussian variables

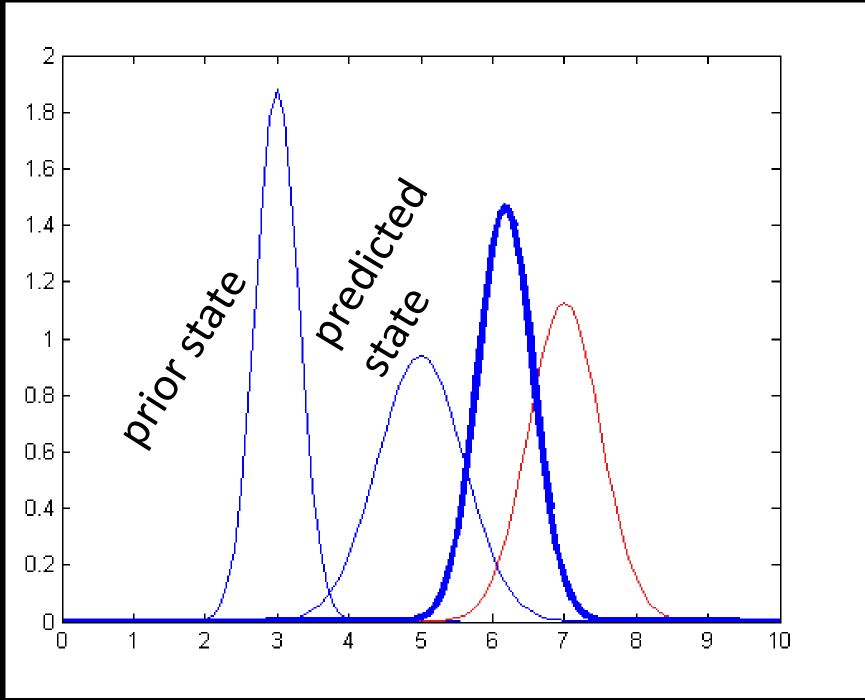
Bayes Filter($bel(x_{t-1})$, u_t, z_t)

1. for all x(t) do
2. $\overline{bel}(x(t)) = \Sigma\, x(t-1)\; p(x(t) \mid u(t), x(t-1))\; bel(x(t-1))$
3. $bel(x(t)) = \alpha\; p(z(t) \mid x(t))\; \overline{bel}(x(t))$
4. end for
5. return $bel(x_t)$

Bayes Filter – Kalman Filter

State estimate: $\mu(t)$
State uncertainty: $\Sigma(t)$
Process noise: Σ_u

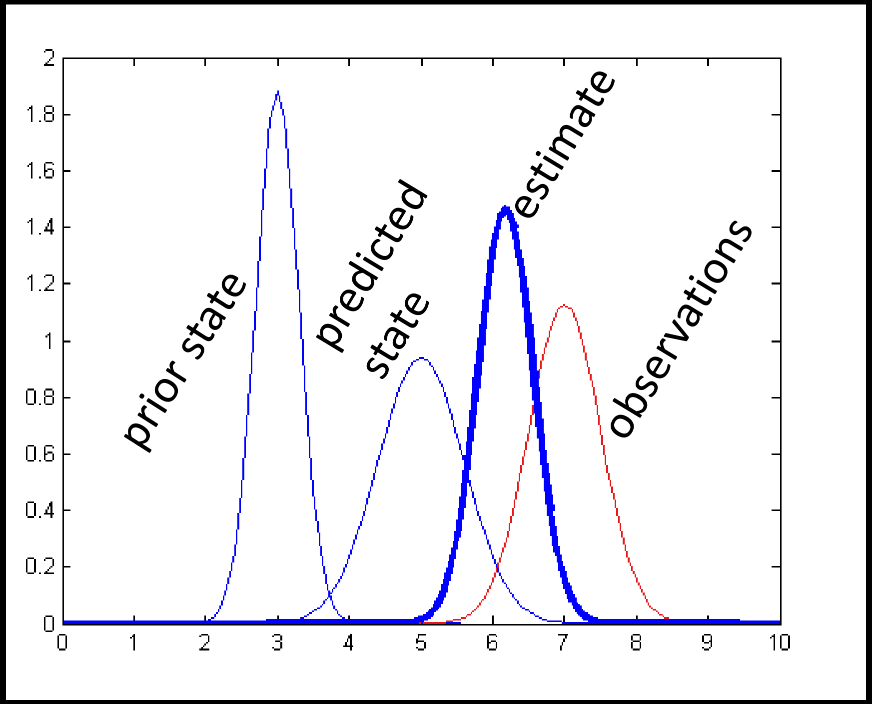
- Incorporate uncertainty to get better estimates based on inputs and observations
- Kalman filter is based on the same idea
  - Assume that posterior and prior belief are Gaussian variables
  - Prediction step
    - $x(t) = A\,x(t-1) + Bu(t) + n$, where…
      - $\mu_p(t) = A\,\mu(t-1) + B\,u(t)$
      - $\Sigma_p(t) = A\,\Sigma(t-1)\,A^T + \Sigma_u$



Bayes Filter – Kalman Filter

- Incorporate uncertainty to get better estimates based on inputs and observations
- Kalman filter is based on the same idea
  - Assume that posterior and prior belief are Gaussian variables
  - Prediction step
    - $x(t) = A\, x(t-1) + Bu(t) + n$, where…
      - $\mu_p(t) = A\, \mu(t-1) + B\, u(t)$
      - $\Sigma_p(t) = A\, \Sigma(t-1)\, A^T + \Sigma_u$
  - Update step
    - $K_{KF} = \Sigma_p(t)\, C^T\, (\, C\, \Sigma_p(t)\, C^T + \Sigma_z)^{-1}$
    - $\mu(t) = \mu_p(t) + K_{KF}\, (\, z(t) - C\, \mu_p(t)\,)$
    - $\Sigma(t) = (\, \mathbf{I} - K_{KF}\, C)\, \Sigma_p(t)$

State estimate: $\mu(t)$
State uncertainty: $\Sigma(t)$
Process noise: Σ_u
Kalman filter gain: K_{KF}
Measurement noise: Σ_z
Observations: $z(t)$



Kalman Filter Implementation

State estimate: $\mu(t)$
State uncertainty: $\Sigma(t)$
Process noise: Σ_u
Kalman filter gain: K_{KF}
Measurement noise: Σ_z
Observations: $z(t)$

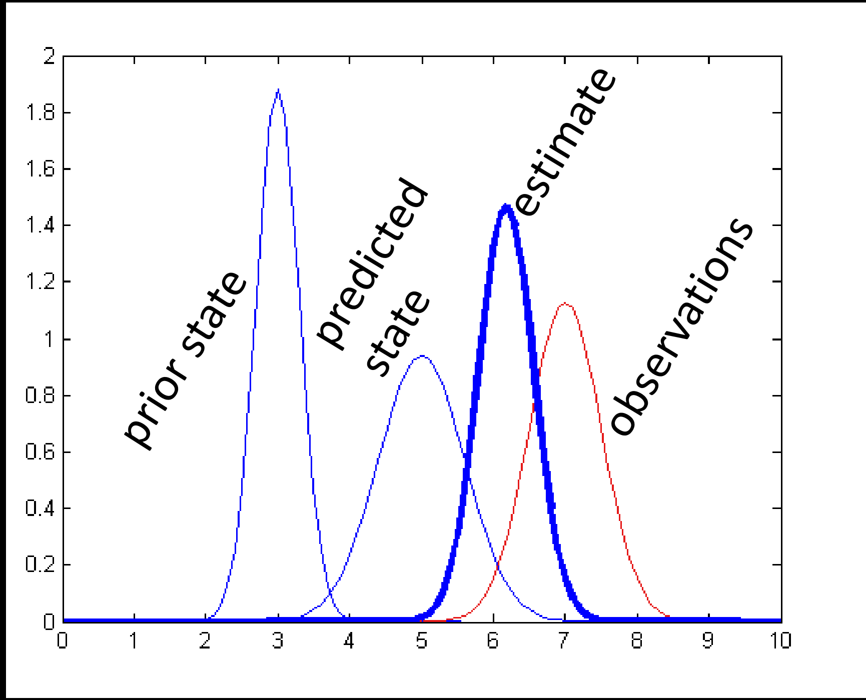
Kalman Filter ($\mu(t-1)$, $\Sigma(t-1)$, u(t), z(t))

1. $\mu_p(t) = A\,\mu(t-1) + B\,u(t)$
2. $\Sigma_p(t) = A\,\Sigma(t-1)\,A^T + \Sigma_u$
3. $K_{KF} = \Sigma_p(t)\,C^T\,(C\,\Sigma_p(t)\,C^T + \Sigma_z)^{-1}$
4. $\mu(t) = \mu_p(t) + K_{KF}\,(z(t) - C\,\mu_p(t))$
5. $\Sigma(t) = (\mathbf{I} - K_{KF}\,C)\,\Sigma_p(t)$
6. Return $\mu(t)$ and $\Sigma(t)$

Steps 1–2: prediction

Steps 3–5: update

$$\Sigma_u = \begin{bmatrix} \sigma_1{}^2 & 0 & 0 \\ 0 & \sigma_2{}^2 & 0 \\ 0 & 0 & \sigma_3{}^2 \end{bmatrix}, \Sigma_z = \begin{bmatrix} \sigma_4{}^2 & 0 \\ 0 & \sigma_5{}^2 \end{bmatrix}$$



Lab 11-12a

- Controllers and Observers
- Real robot
  - Full speed wall-following/turns
  - First order principles and fitting

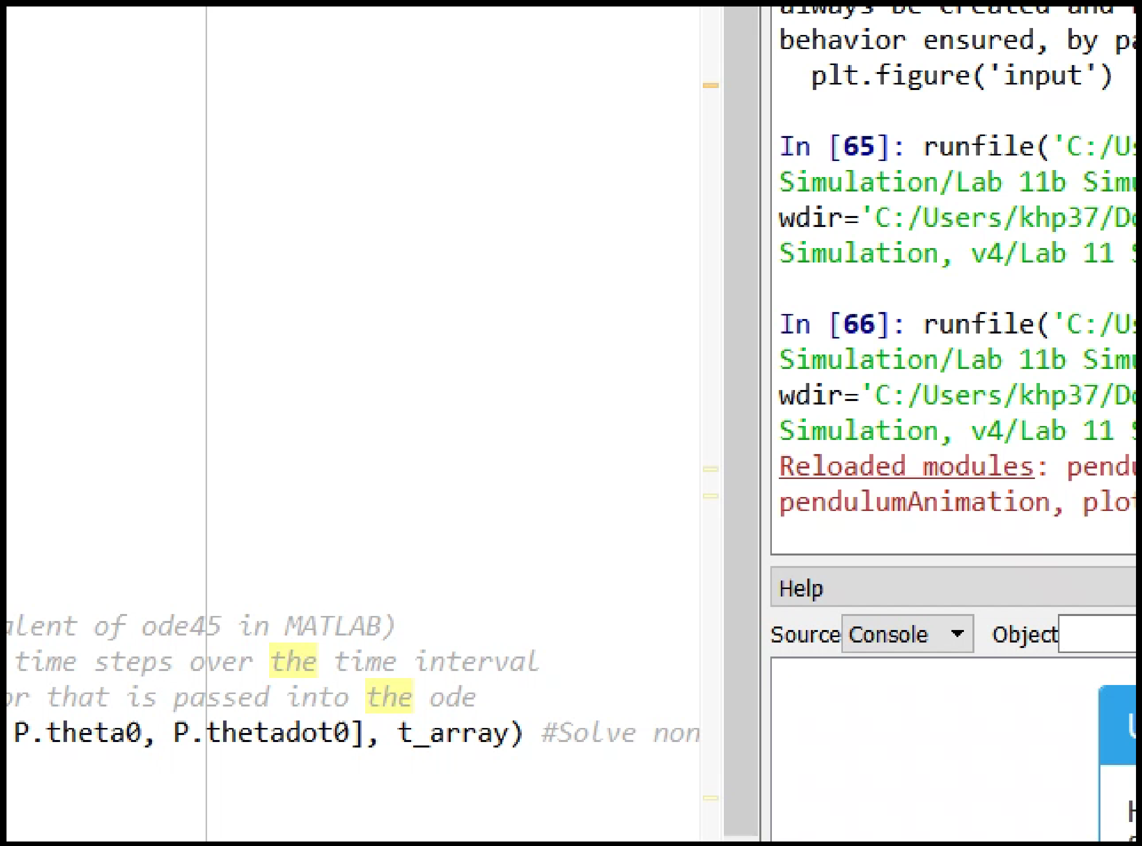
Lab 11-12b

- Controllers and Observers
- Simulation
  - Inverted pendulum on a cart
  - First order principles

Pick one or the other (for both labs)

Lab 11b (and 12b): Inverted pendulum on a cart - *simulation*

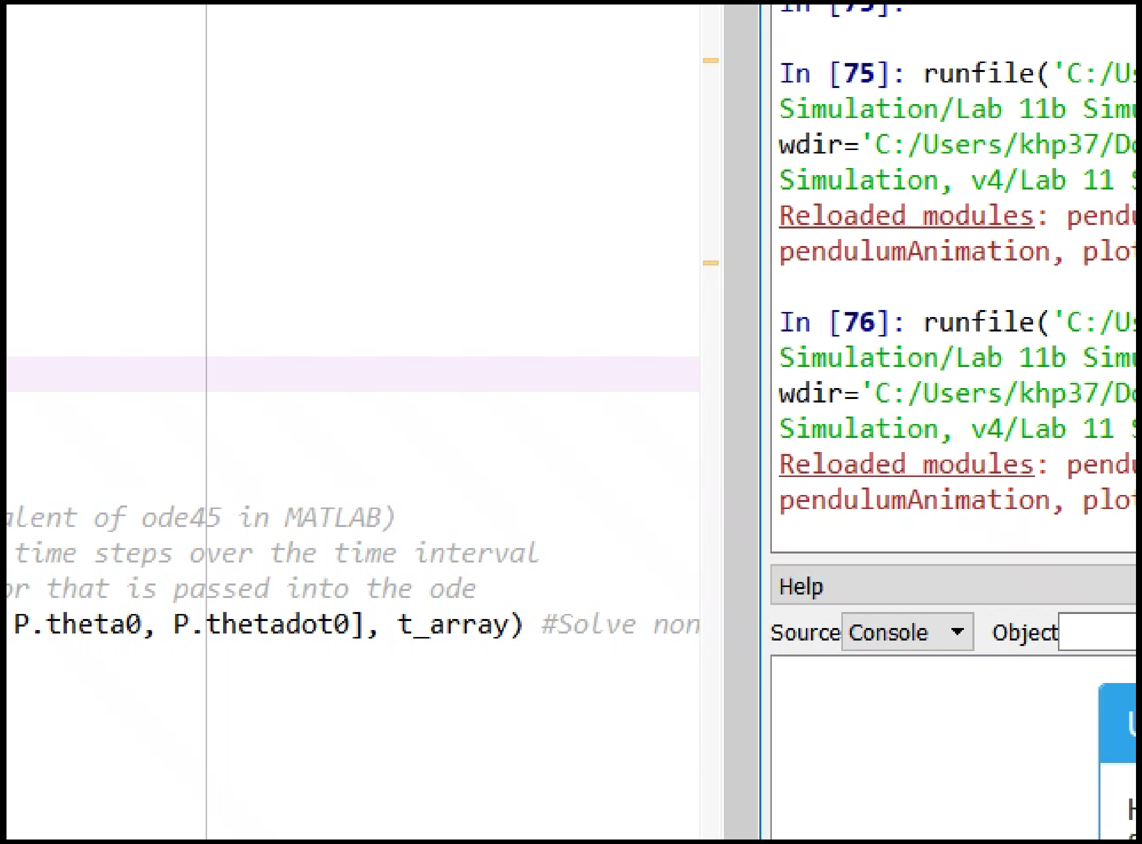
- Objectives
  - Implement a controller and an observer
  - How to best use simulations?
    - Quick and safe testing



no control

Lab 11b (and 12b): Inverted pendulum on a cart - *simulation*

- Objectives
  - Implement a controller and an observer
  - How to best use simulations?
    - Quick and safe testing
    - Implications of nonlinearities
    - Implications of a poor model
    - Implications of perturbations and sensor noise
  - Feel free to try a real implementation!

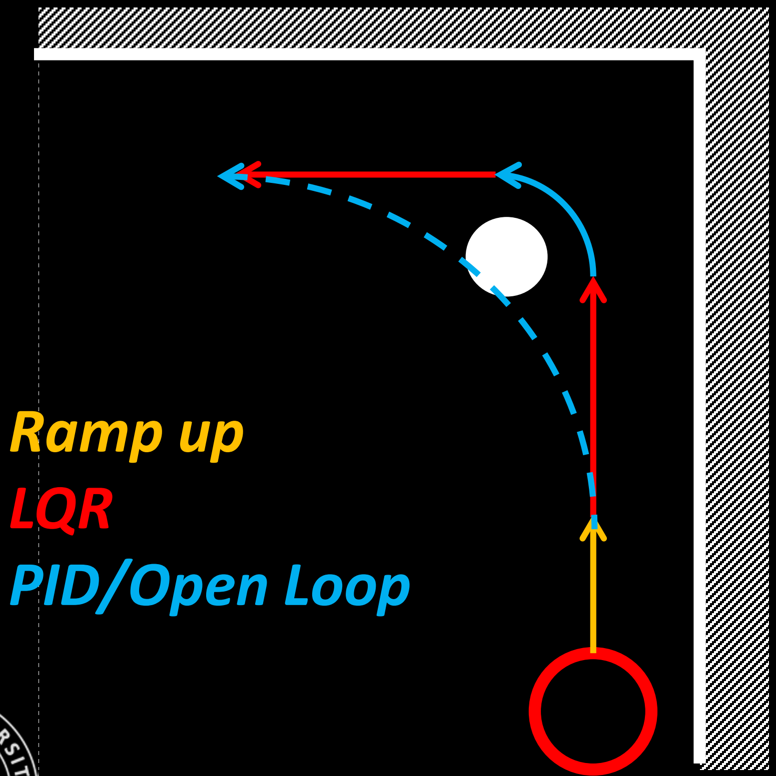


with control and changing z reference

Questions?

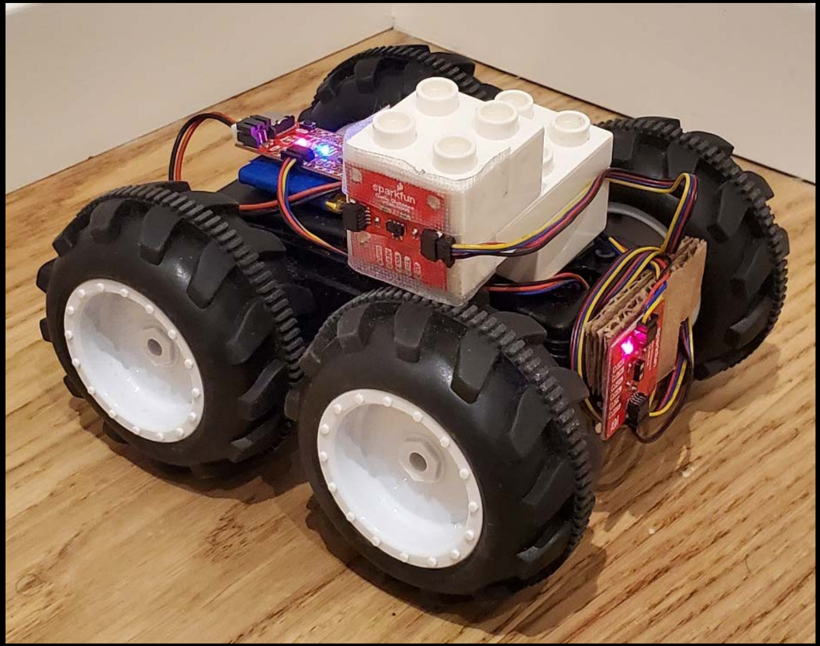
Lab 11a (and 12a): Turning a Corner – *real implementation*

- Objective
  - Implement a controller and an observer
- Lab: Full speed navigation along the inner part of a corner



Lab 11a (and 12a): Turning a Corner – *real implementation*

- Objective
  - Implement a controller and an observer
- Lab: Full speed navigation along the inner part of a corner
  - Lab 11a: LQR control for full speed wall following



Lab 11a (and 12a): Turning a Corner – *real implementation*

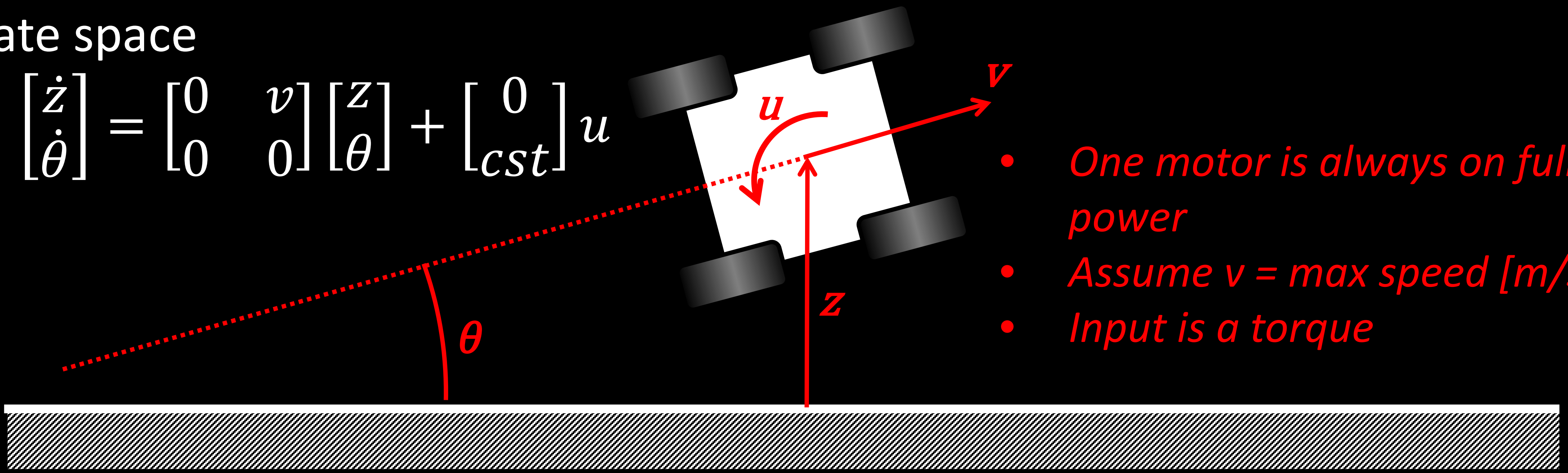
- Objective
  - Implement a controller and an observer
- Lab: Full speed navigation along the inner part of a corner
  - Lab 11a: LQR control for full speed wall following
    - State space
    - Equations of motion
    - Estimate parameters for A and B
    - Estimate and tune Q and R
    - Compute the LQR gain, K_r

Lab 11a (and 12a): Turning a Corner – *real implementation*

- State space
  - $\begin{bmatrix} z \\ \theta \end{bmatrix}$
- Small angle approximation
  - $\dot{z} = v\theta$
- State space
  - $\begin{bmatrix} \dot{z} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & v \\ 0 & 0 \end{bmatrix} \begin{bmatrix} z \\ \theta \end{bmatrix} + \begin{bmatrix} 0 \\ cst \end{bmatrix} u$

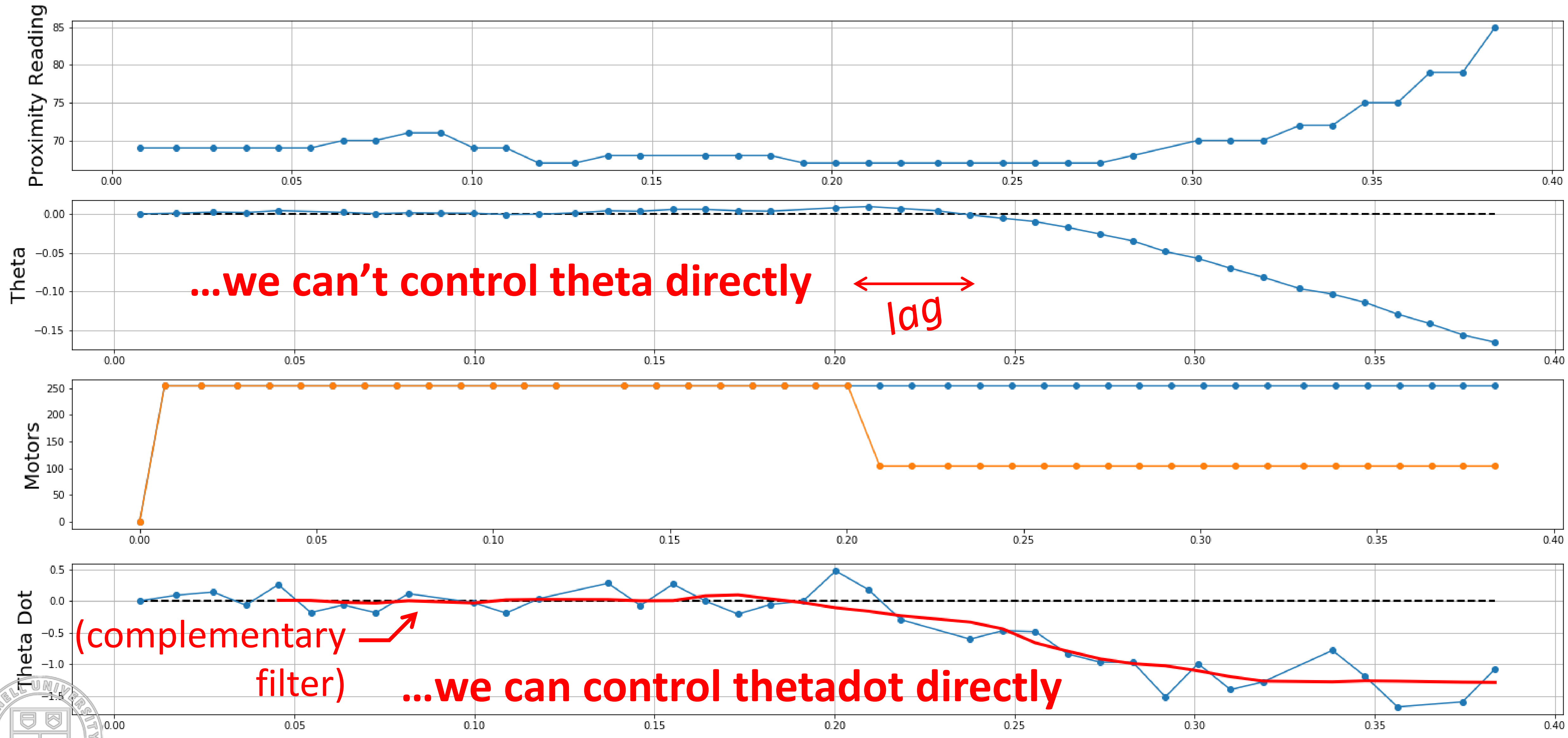
What do we do now?

- Check out the step response



- *One motor is always on full power*
- *Assume v = max speed [m/s]*
- *Input is a torque*

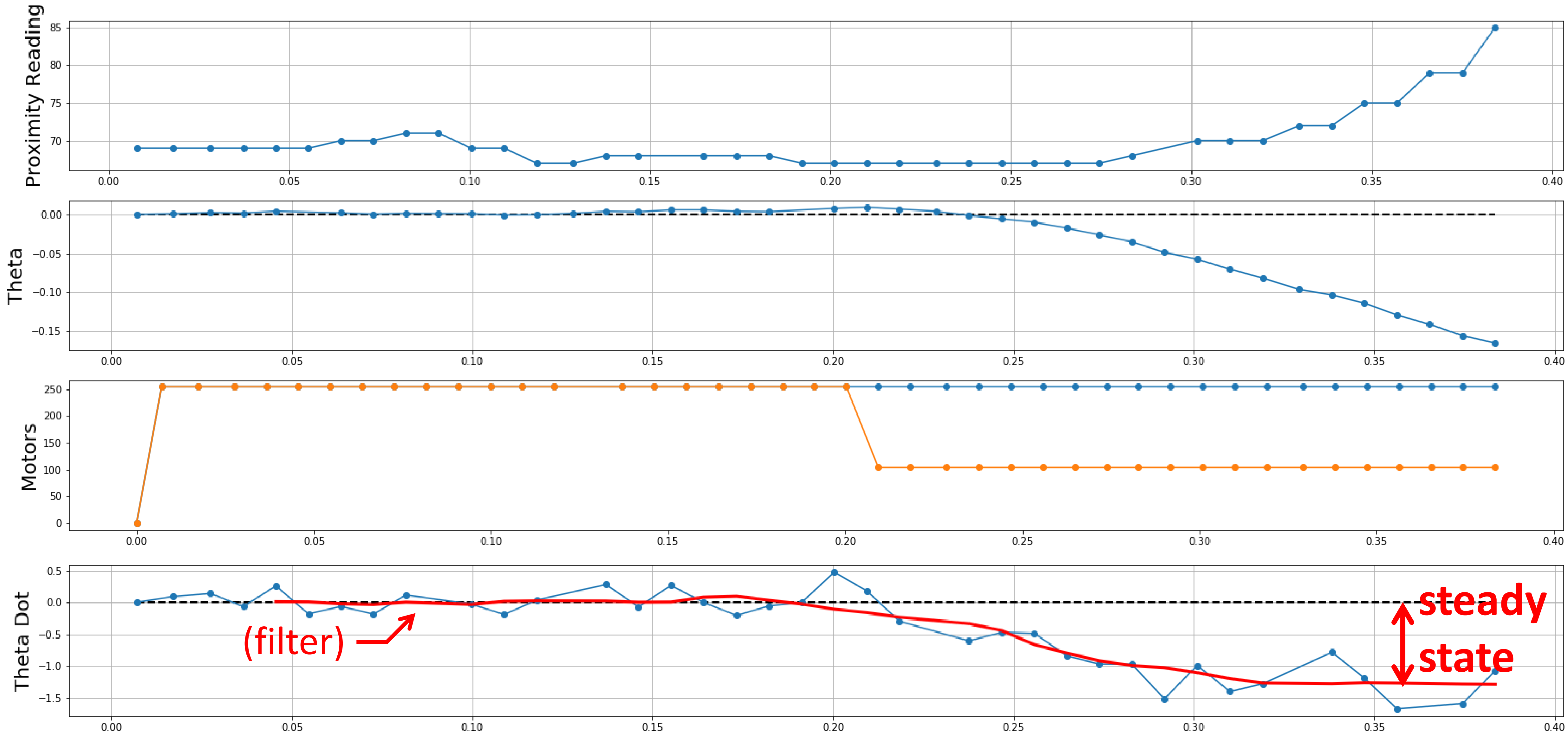
Lab 11a (and 12a): Turning a Corner – *real implementation*



Lab 11a (and 12a): Turning a Corner – *real implementation*

- Basic equation
  - $\tau = I\ddot{\theta}$
- Torque experienced by robot
  - $U - d\dot{\theta} = I\ddot{\theta}$
  - $\frac{U}{I} - \frac{d}{I}\dot{\theta} = \ddot{\theta}$ *d, I?*
- Steady state
  - $\ddot{\theta}_{SS} = 0$
  - $\frac{U_{SS}}{I} - \frac{d}{I}\dot{\theta}_{SS} = 0$
  - $d = \frac{-U_{SS}}{\dot{\theta}_{SS}}$

Lab 11a (and 12a): Turning a Corner – *real implementation*



Lab 11a (and 12a): Turning a Corner – *real implementation*

- Basic equation
  - $\tau = I\ddot{\theta}$
- Torque experienced by robot
  - $U - d\dot{\theta} = I\ddot{\theta}$
  - $\frac{U}{I} - \frac{d}{I}\dot{\theta} = \ddot{\theta}$ *d, I?*
- Steady state
  - $\ddot{\theta}_{ss} = 0$
  - $\frac{U_{ss}}{I} - \frac{d}{I}\dot{\theta}_{ss} = 0$
  - $d = \frac{-U_{ss}}{\dot{\theta}_{ss}}$

- Use the 90% rise time to determine I
  - Pretend $\dot{\theta}$=x
  - $\dot{x} = -\frac{d}{I}x + \frac{U}{I}$
  - $\dot{x} + \frac{d}{I}x = \frac{U}{I}$
  - $x = 1 - e^{-\frac{d}{I}t_{0.9}}$
  - $1 - x = e^{-\frac{d}{I}t_{0.9}}$
  - $\ln(1 - x) = -\frac{d}{I}t_{0.9}$
  - $I = \frac{-dt_{0.9}}{\ln(0.1)}$

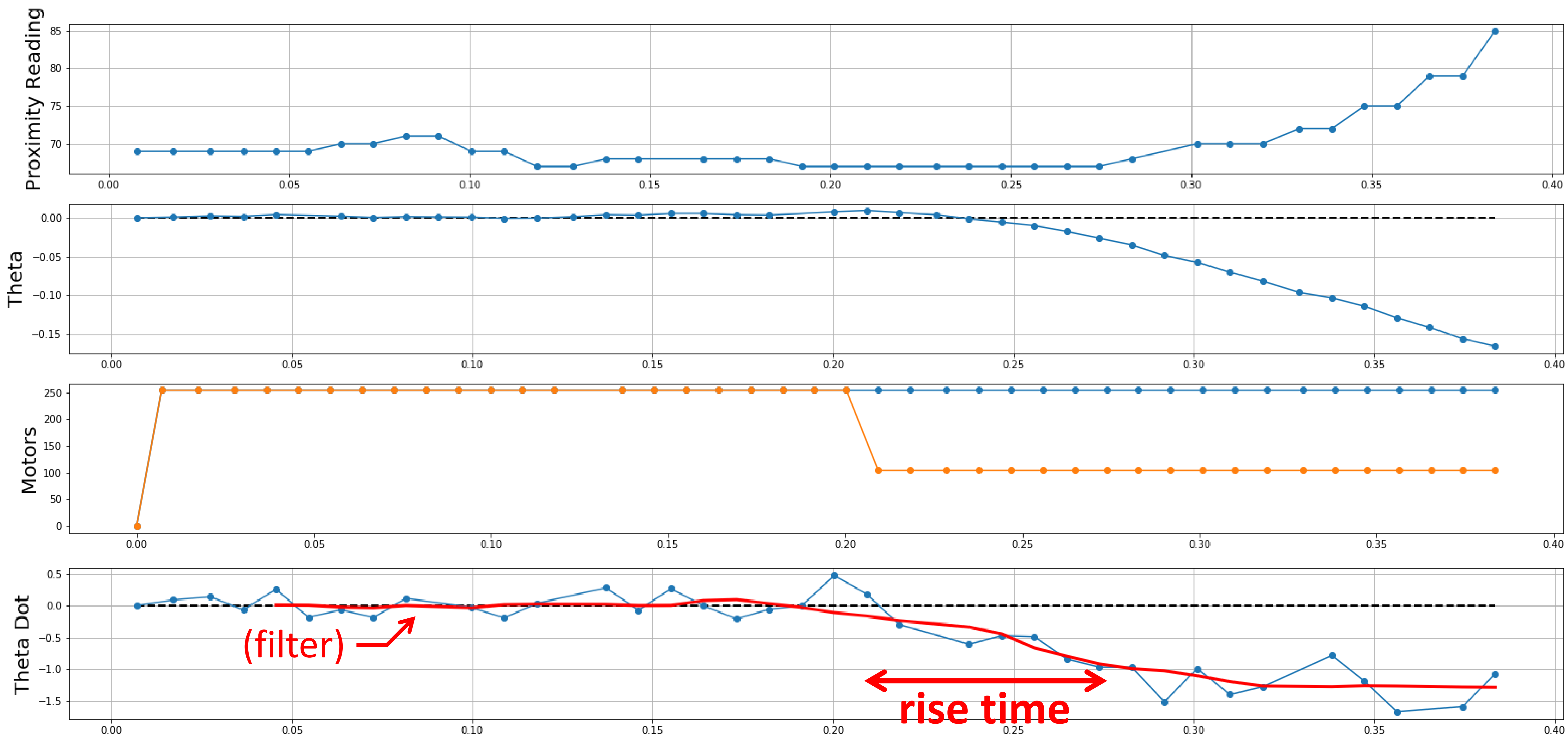
1st order system:

$$\frac{dy(t)}{dt} + \frac{1}{\tau}y(t) = x(t)$$

Step response solution:

$$y(t) = 1 - e^{-\frac{t}{\tau}}$$

Lab 11a (and 12a): Turning a Corner – *real implementation*



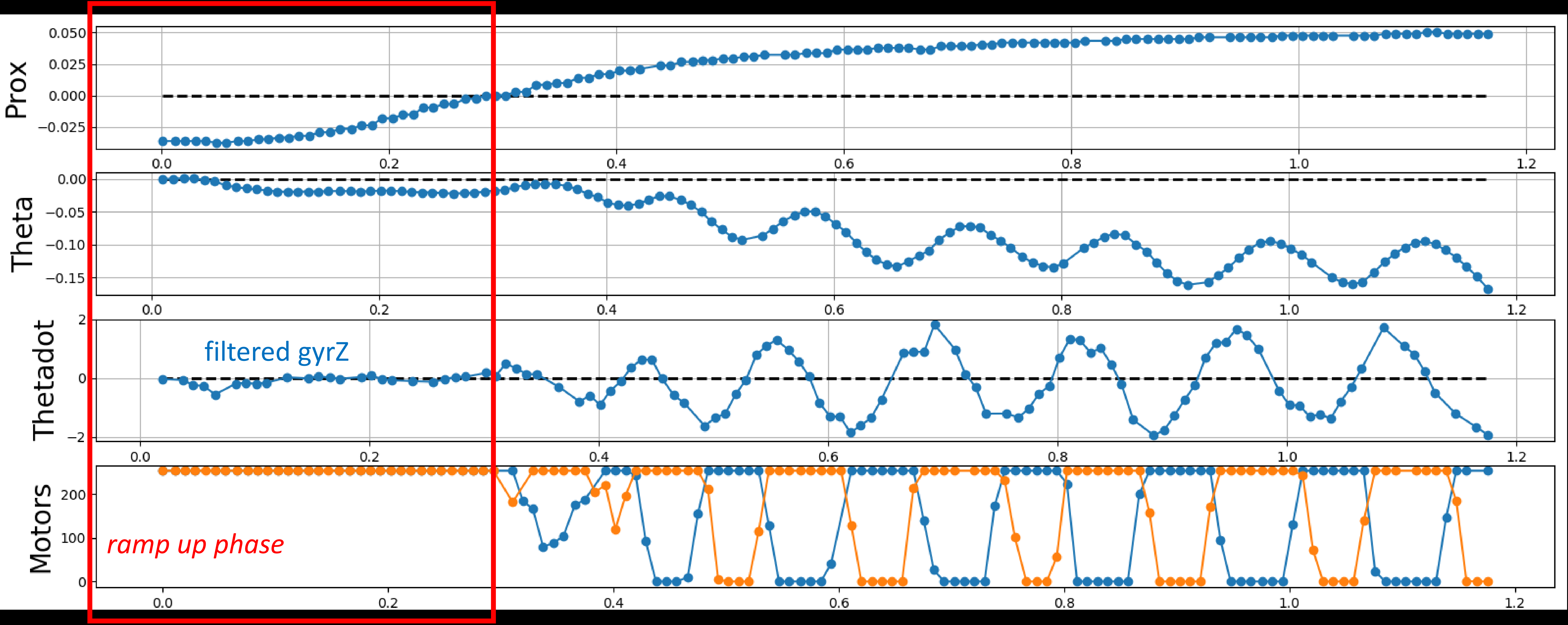
Lab 11a (and 12a): Turning a Corner – *real implementation*

- Equations of motion
  - $\dot{z} = v\theta$
  - $\ddot{\theta} = -\frac{d}{I}\dot{\theta} + \frac{U}{I}$
  - Steady state: $d = \frac{-U_{SS}}{\dot{\theta}_{SS}}$
  - Rise time: $I = \frac{-dt_{0.9}}{\ln(0.1)}$
- State space
  - $\dot{x} = Ax + Bu$
  - $\begin{bmatrix} \dot{z} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & v & 0 \\ 0 & 0 & 1 \\ 0 & 0 & \frac{-d}{I} \end{bmatrix} \begin{bmatrix} z \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{I} \end{bmatrix} U$

- Now, implement LQR feedback
  - Q, R cost functions…
    - $Q = \begin{bmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & s \end{bmatrix}$
    - $R = [s]$

Lab 11a (and 12a): Turning a Corner – *real implementation*

Lab 11a (and 12a): Turning a Corner – *real implementation*



- Adjust cost functions
- Keep units in mind
- What if we start at an angle?
- Make controller more aggressive?

Lab 11a (and 12a): Turning a Corner – *real implementation*

- Objective
  - Implement a controller and an observer
- Lab: Full speed navigation along the inner part of a corner
  - Lab 11a: LQR control for full speed wall following
    - State space
    - Equations of motion
    - Estimate A and B parameters
    - Estimate and tune Q and R
    - Compute the LQR gain, K_r
  - Lab 12a: LQG control for full speed wall following and corner turning
    - Compute and implement a Kalman Filter
    - Experiment with turns and drift

Questions?

Next up!

- December 3rd: Prof. George Konidaris, RealTime Robotics
- December 8th: Prof. Silvia Ferrari, Lab of Intelligent Systems and Control (LISC), Cornell
- December 10th: Dr. Vasumathi Raman, Zipline Robotics
- December 15th: Semester recap and evaluation/brainstorm