# Track Memorization

EE192 Spring 2019

Lab Lecture 13: April 24, 25

Justin Yim

# Track Memorization

- Objective: use prior knowledge to run the track faster on the second lap
  - "Anticipate" turns
  - Optimize path


- Two related problems:
  - Mapping: where is the track?
  - Localization: where am I?

  - Note: we do not need *Simultaneous* Localization and Mapping (SLAM); we will do one, then the other

# Track Memorization

- Three steps:
  1. **Map** the track
  2. **Plan** a trajectory for lap two
  3. Follow the planned trajectory with **localization** and control (you already have control)

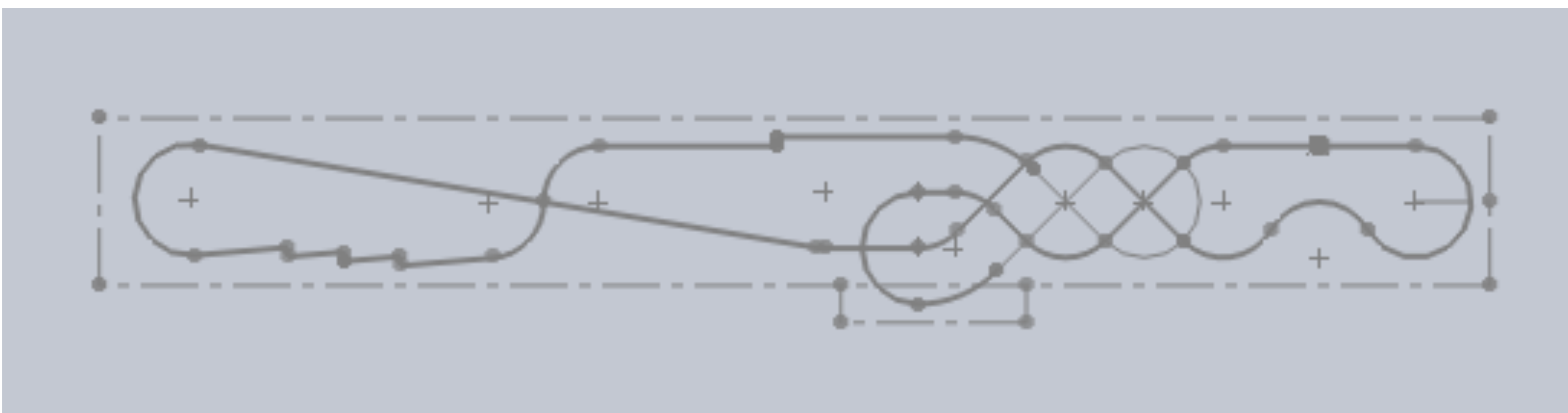
# 1. Mapping

# 1. Mapping

- Objective: record the track layout
- During mapping phase, assume the car's position estimate is perfect to record location of track features
- Use wheel odometry, integrated velocity, etc. for distance
  - Go slow so the car doesn't slip or behave oddly

# 1. Mapping

- Several possible parameterizations:
  - 2D track path (like a map)
    - con: complex loop closures

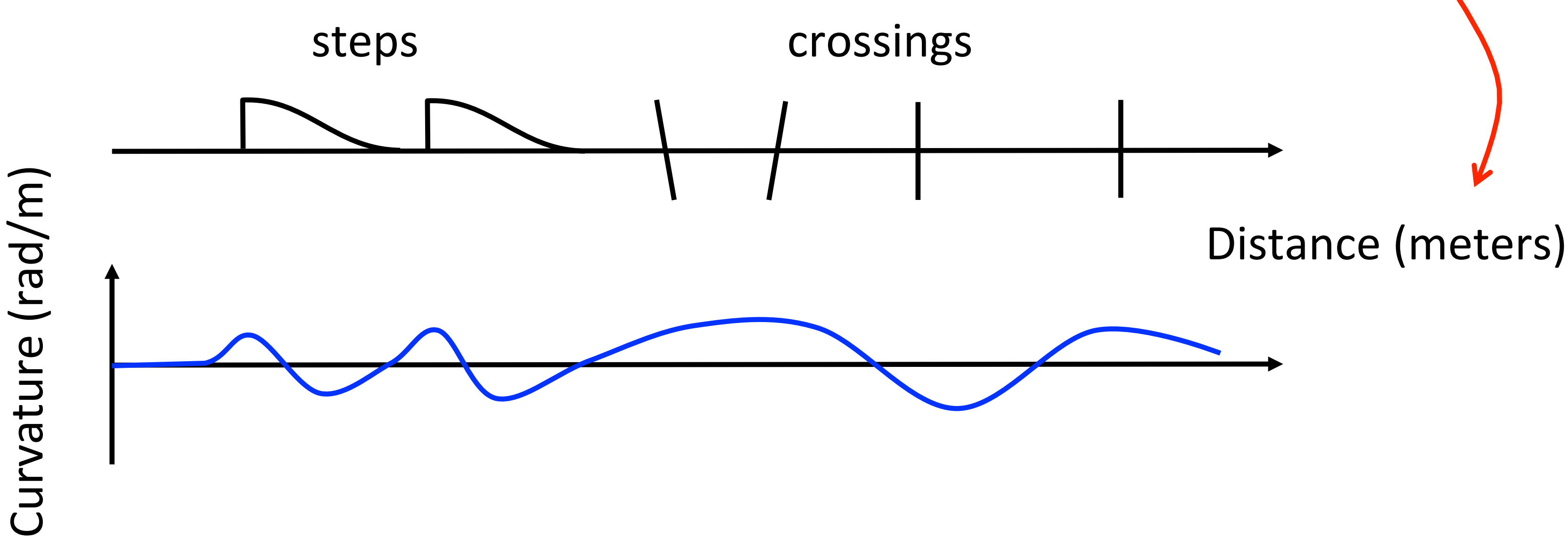


Note: NATCAR explicitly bans pre-coding the track and allows memorizing it

# 1. Mapping

- Several possible parameterizations:
  - 1D list of track features (like driving directions)
    - Steps, heading, etc.
    - Most important: curvature

Map distance, NOT time: lap 2 distance is equal, but time is faster



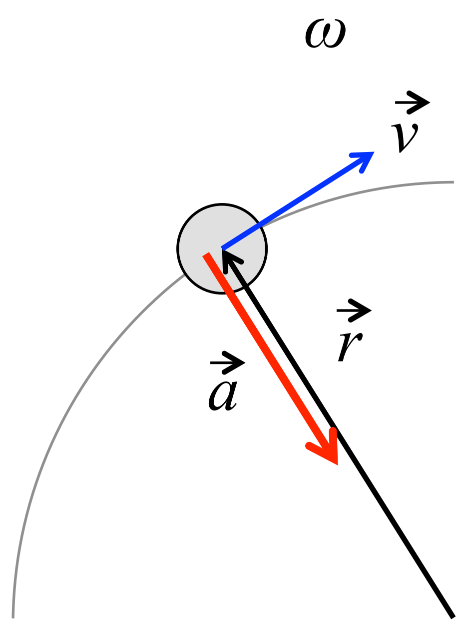
# 1. Mapping

- How to store the map?
- A 1D map of curvature could be stored simply in a very long array of ints
  - I’m using a 20,000 element array of ints recording curvature in centi-radians/meter every 10 cm

# 2. Trajectory Planning

# 2. Trajectory Planning

- Recall the friction performance limits
- The car can corner only so fast and accelerate/brake only so fast



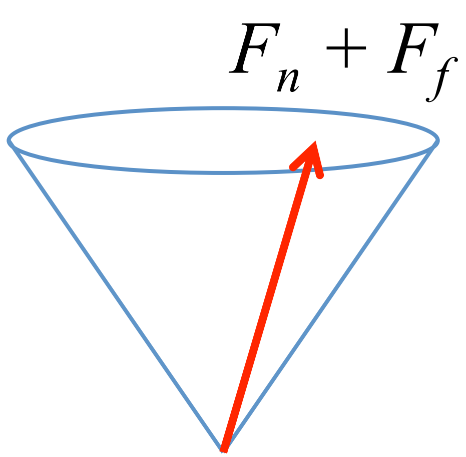

- Back of the envelope limits:
  - $a_r = v^2/r$
  - $a_\theta = dv/dt$
  - $\vec{a} = ||a_r + a_\theta||_2 \leq \mu g$

- Car dynamics caveats:
  - Lateral and longitudinal weight shift
  - More complex tire dynamics

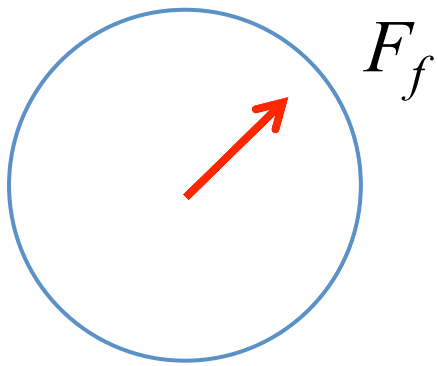
# 2. Trajectory Planning

- Optimal trajectory operates the car near the friction limit as much as possible
  - Take corners at maximum cornering speed
  - Maximum acceleration/braking on the straightaways
- How do we come up with a good velocity plan?

Friction cone




Friction cone top view

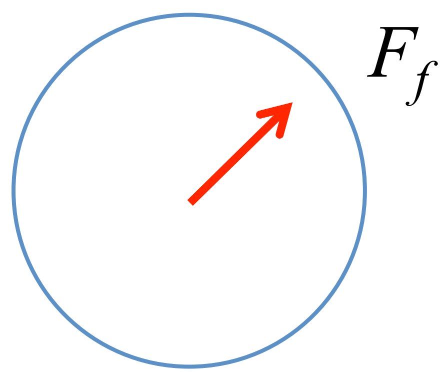


# 2. Trajectory Planning

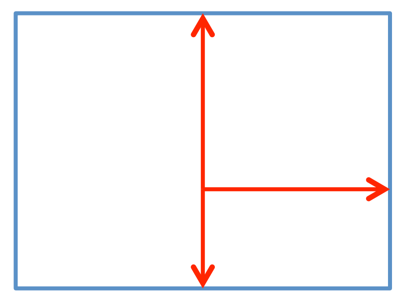
Justin's strategy: simplifying assumptions:

1. Ignore coupling between $a_r$ and $a_\theta$
2. Top speed $v_{max}$
3. Max acceleration $a_{max}$
4. Max braking $a_{min}$

Friction cone top view



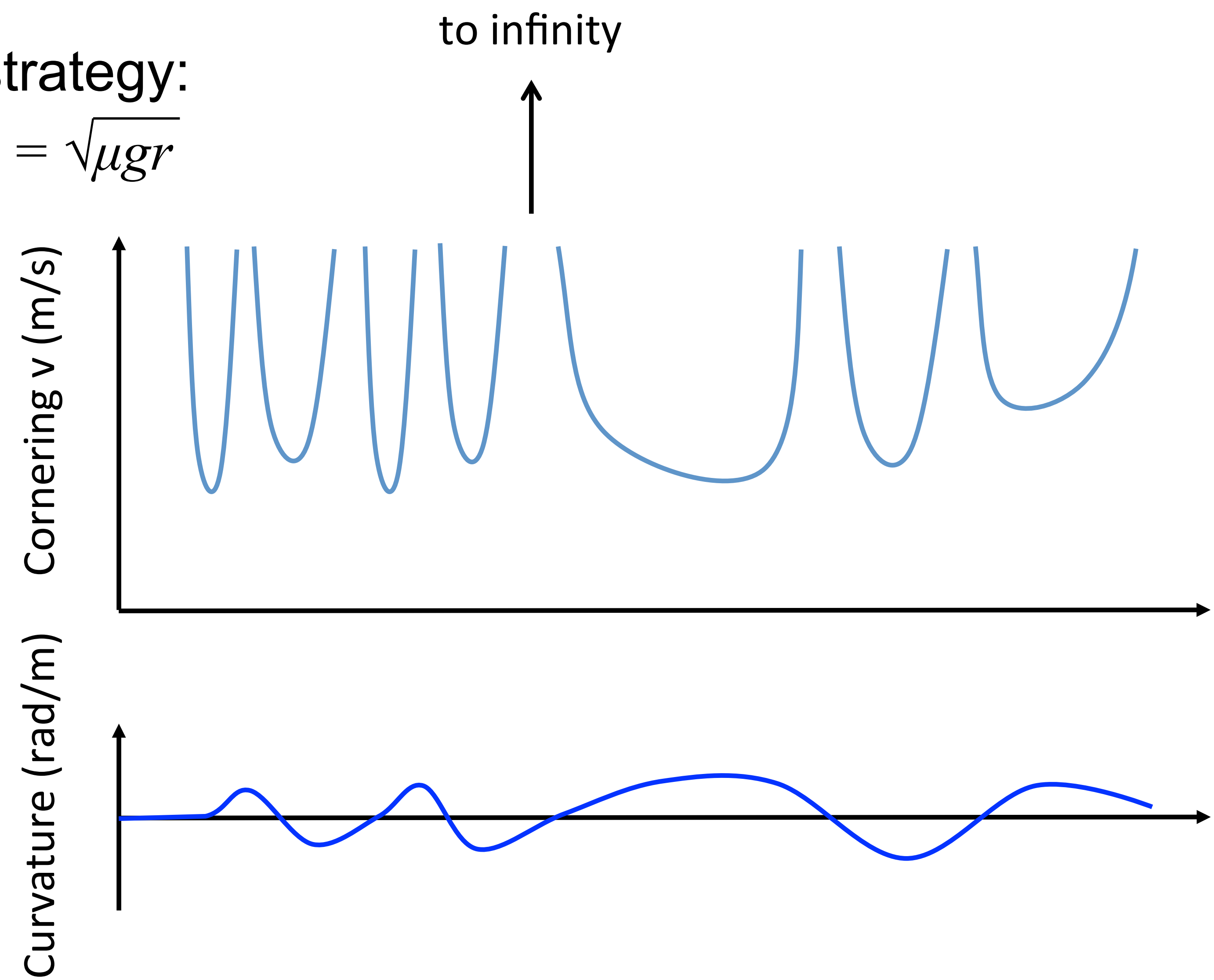

Friction "Box"?



# 2. Trajectory Planning

Justin's strategy:

– 1. $v = \sqrt{\mu g r}$

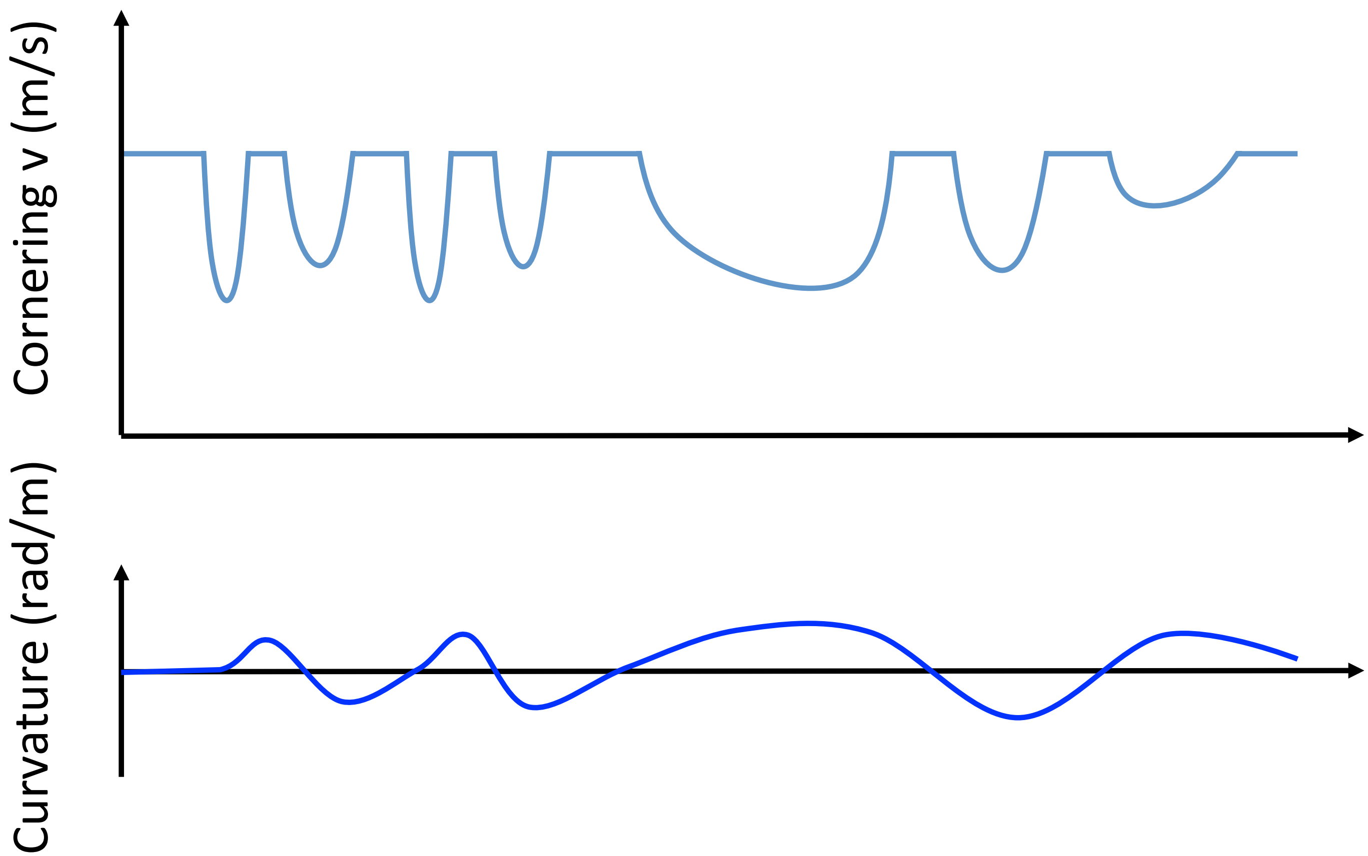



Note r = 1/curvature

# 2. Trajectory Planning

Justin's strategy:

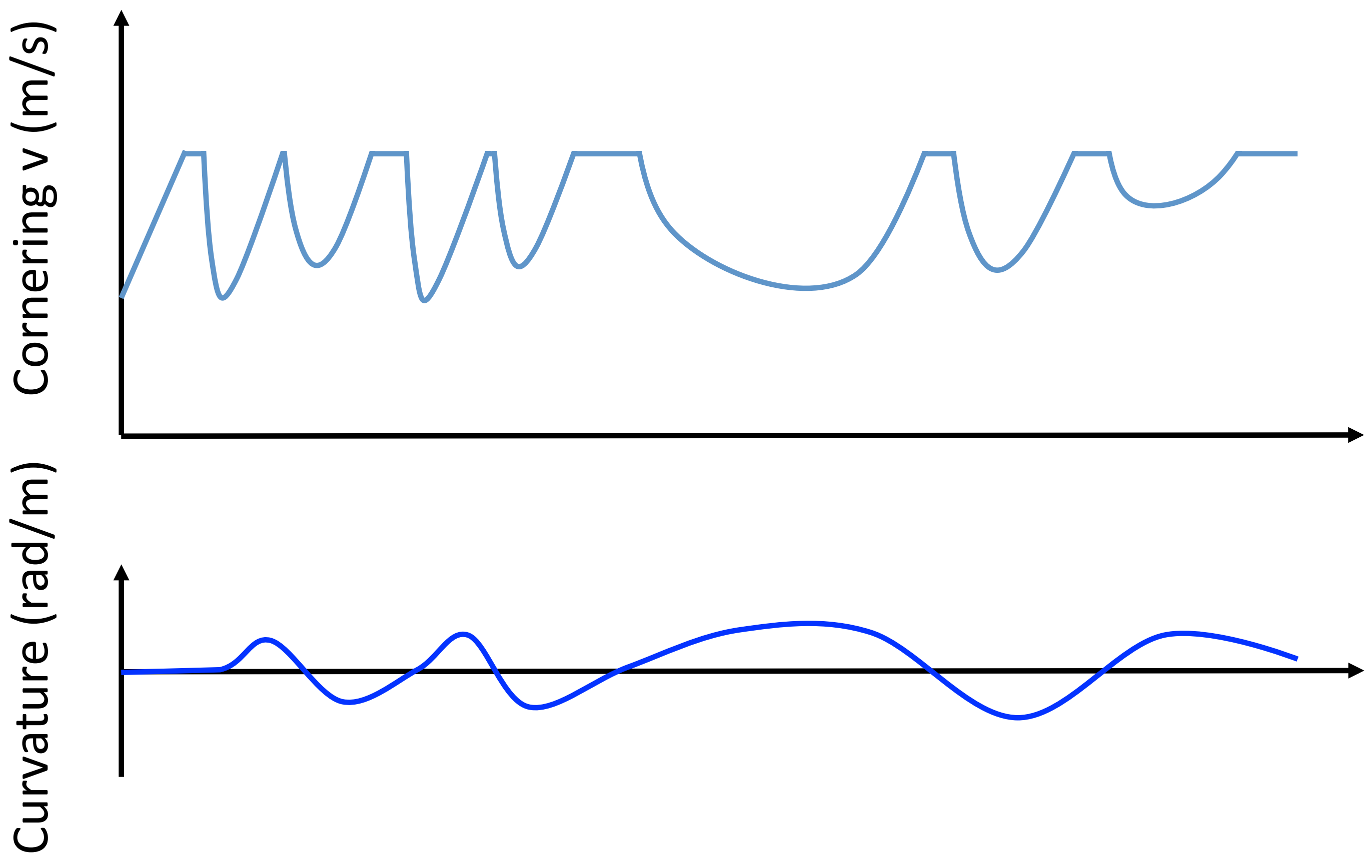
– 2. $v \leq v_{max}$



# 2. Trajectory Planning

Justin's strategy:

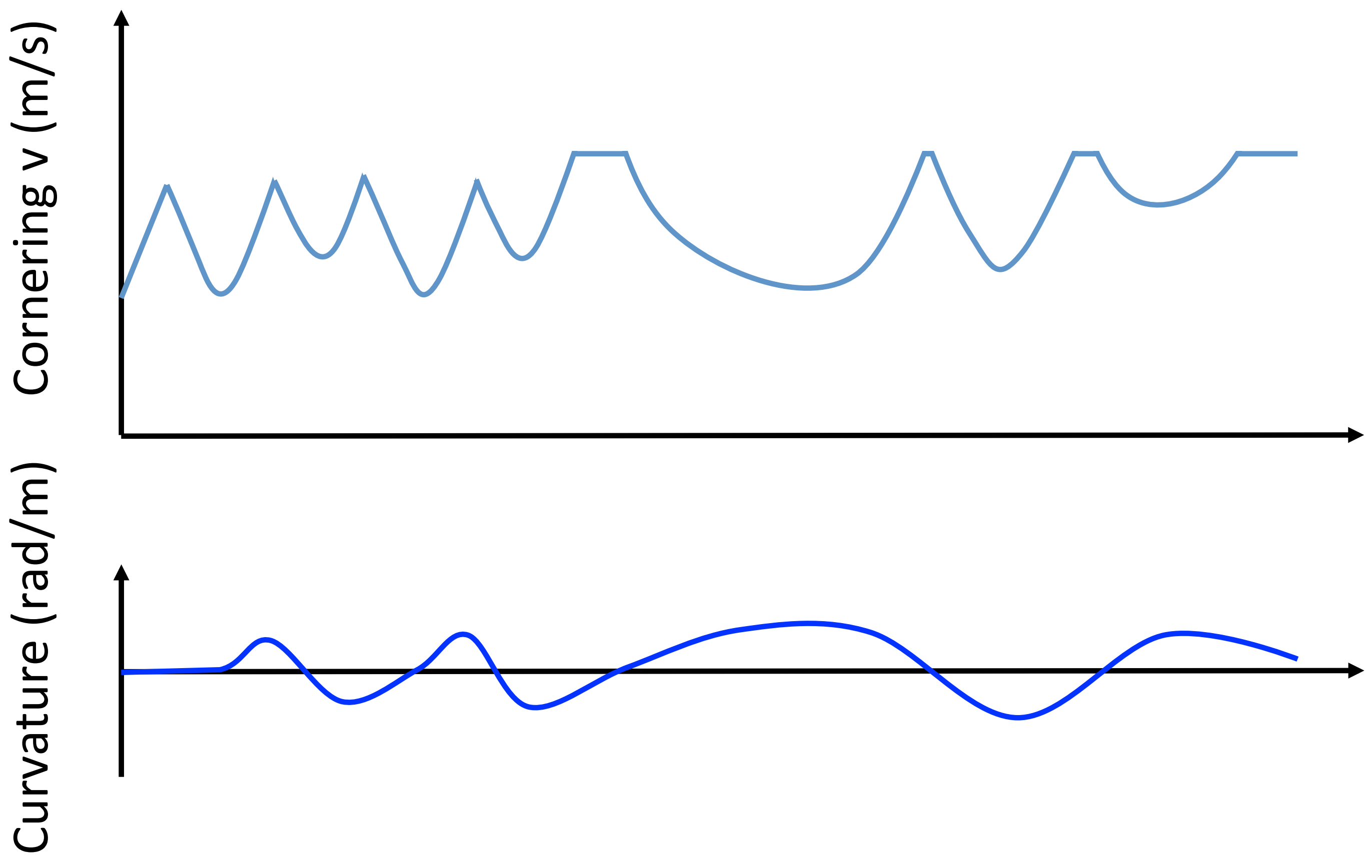
– 3. $dv/dt \leq a_{max}$



# 2. Trajectory Planning

Justin's strategy:

– 4. $dv/dt \geq a_{min}$

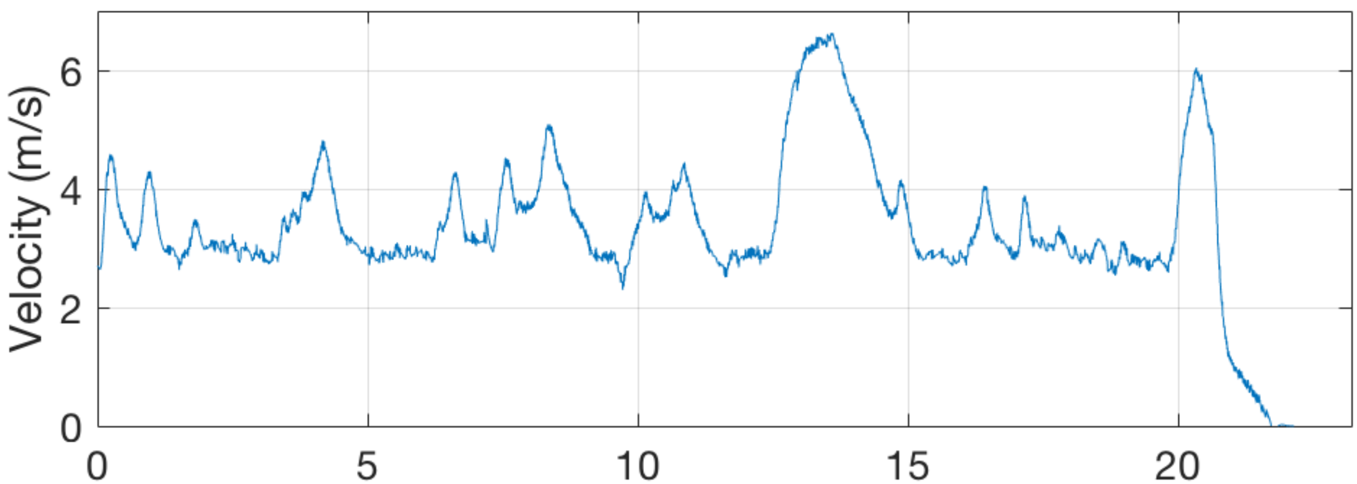


# 2. Trajectory Planning

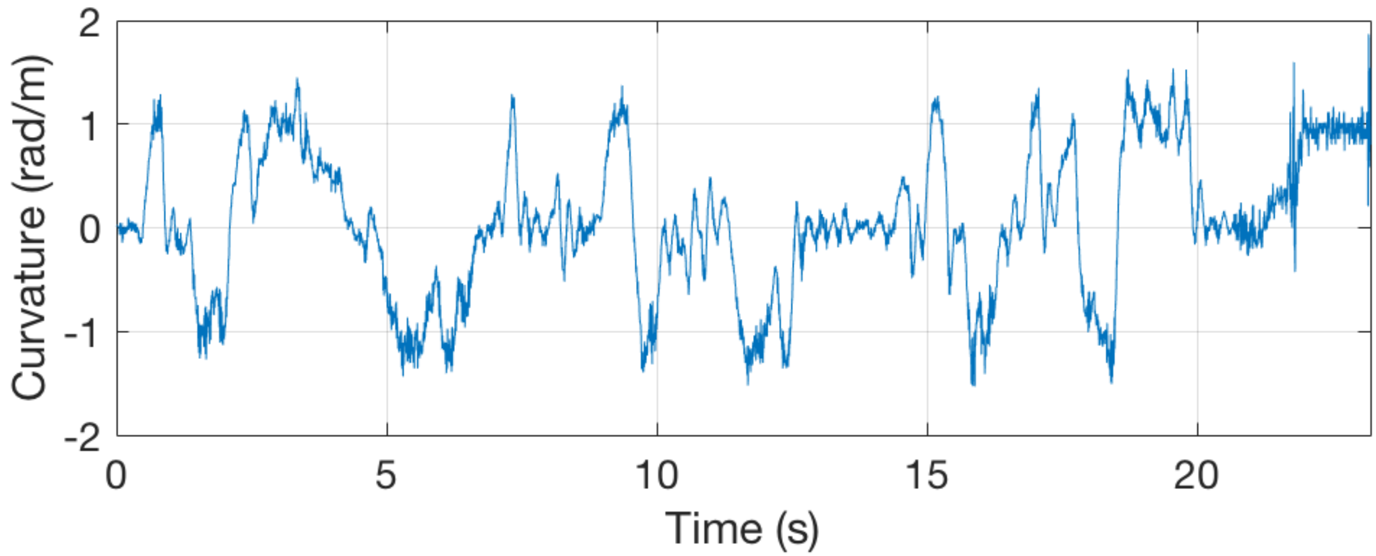
Justin's strategy can be computed in two for loops: one forwards and one backwards

- Forwards: set $v = \sqrt{\mu g r}$, $v \le v_{max}$, $dv/dt \le a_{max}$
- Backwards: set $dv/dt \ge a_{min}$

# 2. Trajectory Planning




I didn't save the plan, so here is the TA car telemetry from following it

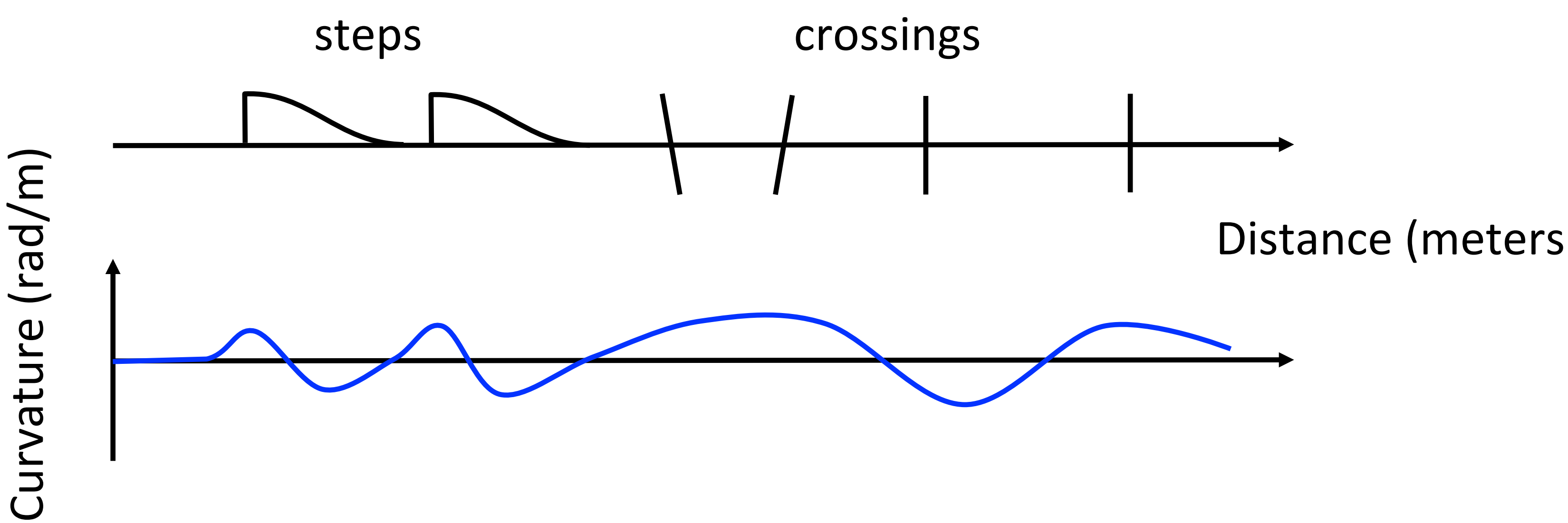


# 3. Localization

# 3. Localization

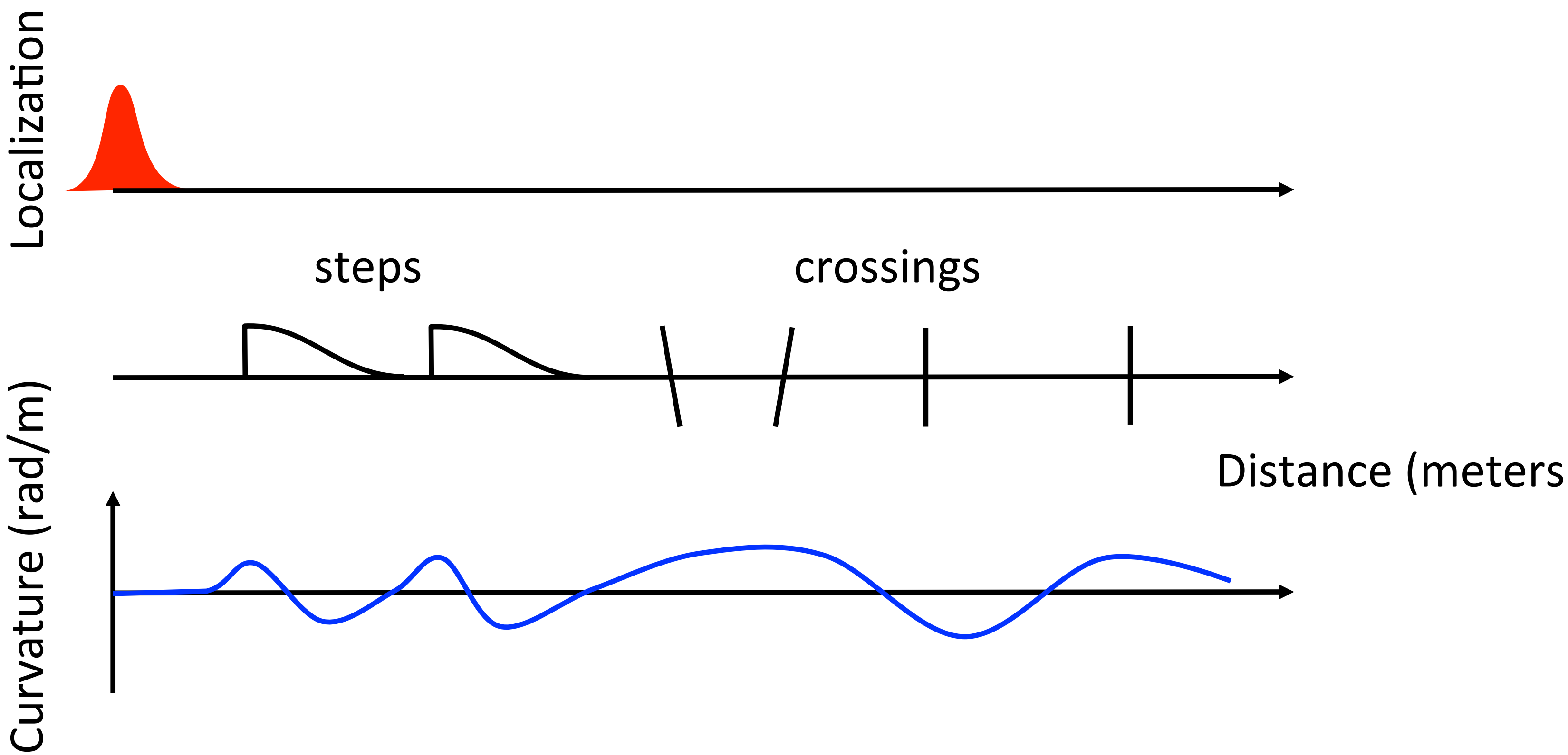
- Assume the map is perfect and correlate observed sensor readings with the map to estimate location

- This can be formulated as Bayesian estimation



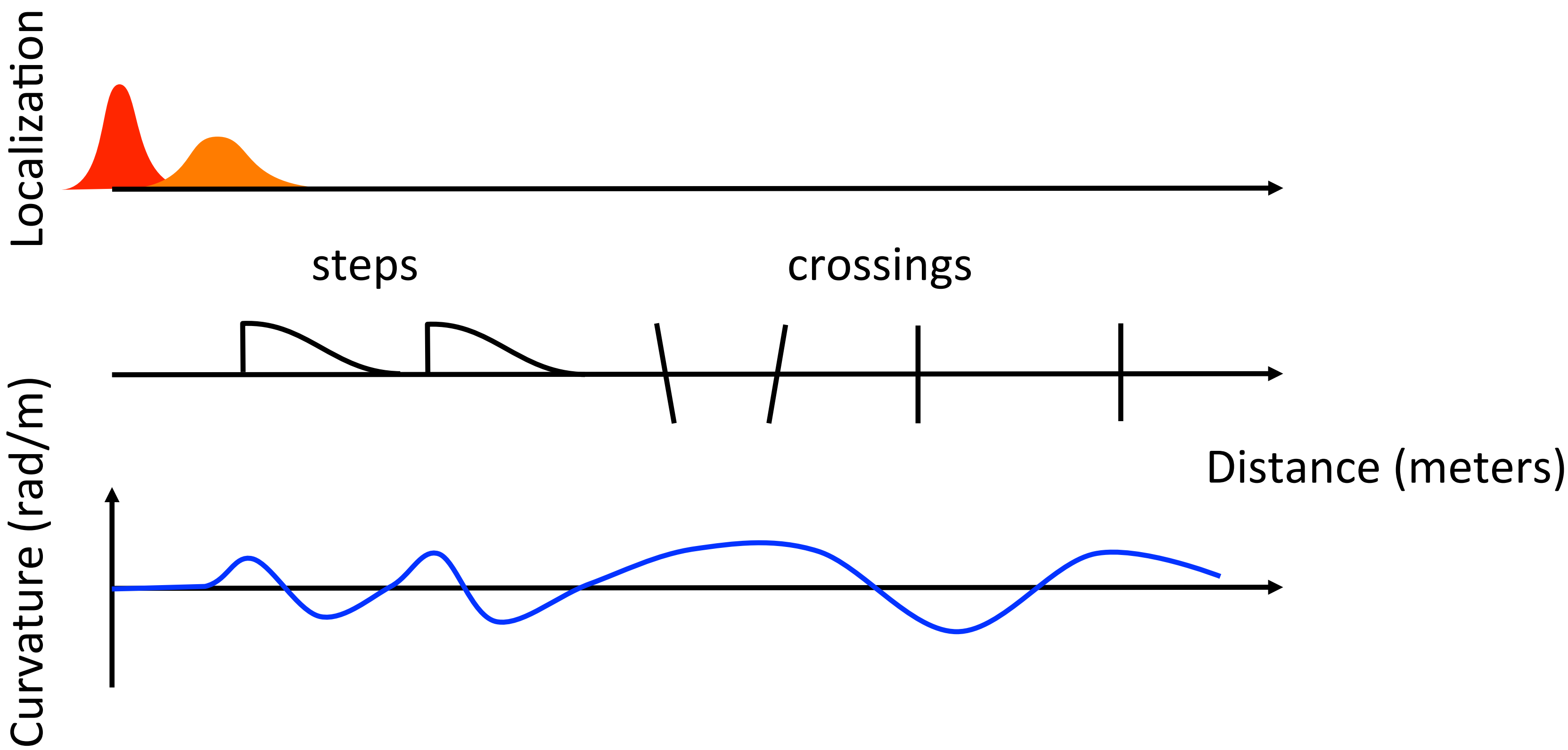
# 3. Localization

- This can be formulated as Bayesian estimation:
  - Probability distribution for distance along track
  - 1. Initial location (a “tracking” problem: we know start)



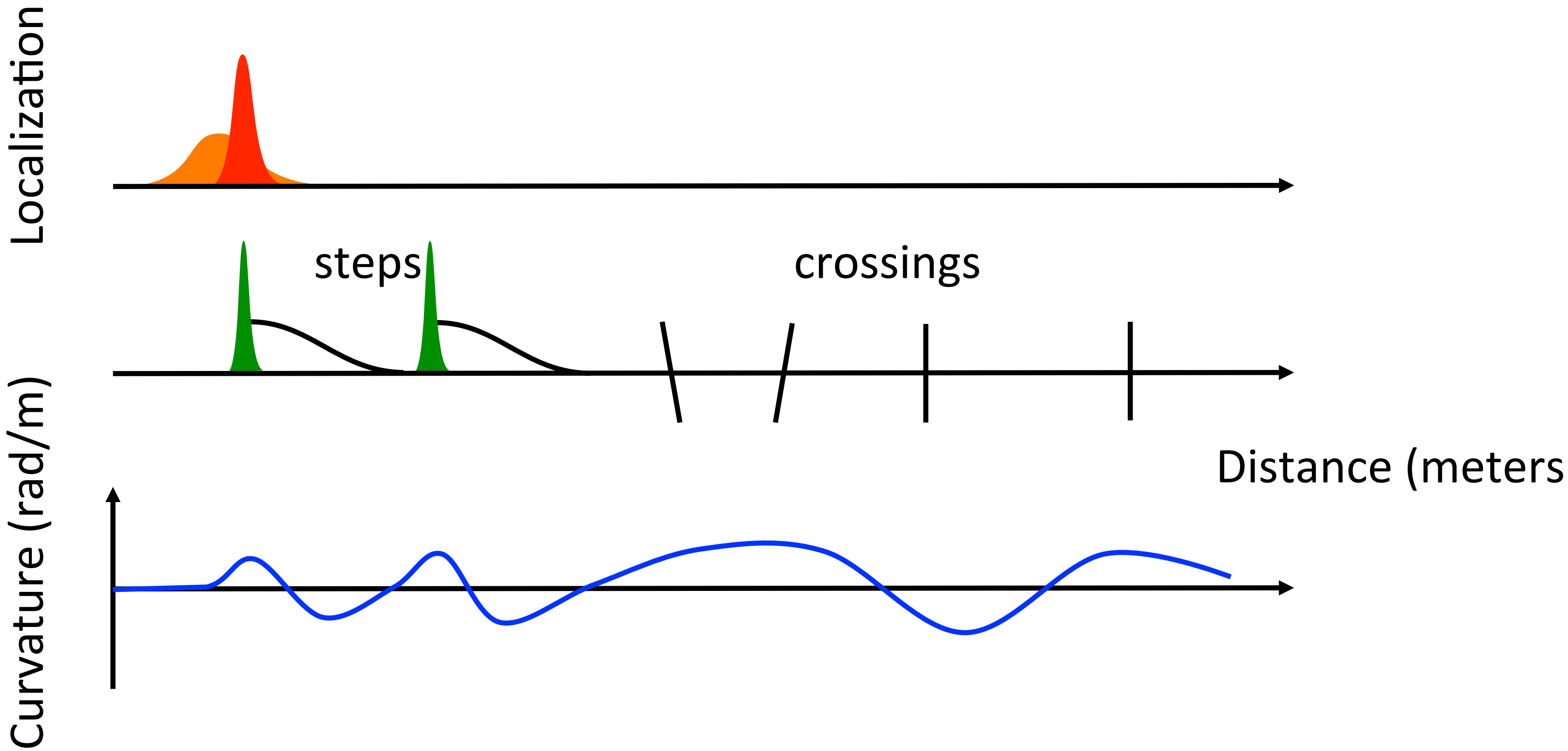
# 3. Localization

- This can be formulated as Bayesian estimation:
  - 2. Use dynamics to predict our position ($x \mathrel{+}= v\,\Delta t$)
  - Move forward and uncertainty increases



# 3. Localization

- This can be formulated as Bayesian estimation:
  - 3. Compare sensor readings to environment
  - Uncertainty should decrease (e.g., we see a step)

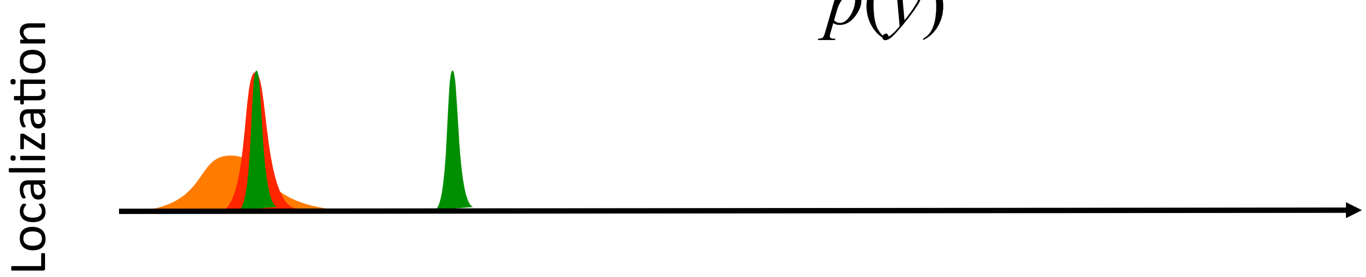


# 3. Localization

- This can be formulated as Bayesian estimation:
  - 2. Use dynamics to predict our position $x$
  - 3. Compare sensor readings $y$ to map

Bayes rule

$$p(x|y) = \frac{p(y|x)\, p(x)}{p(y)}$$



# 3. Localization

- This can be formulated as Bayesian estimation:
  - 2. Use dynamics to predict our position $x$
  - 3. Compare sensor readings $y$ to map

Bayes rule

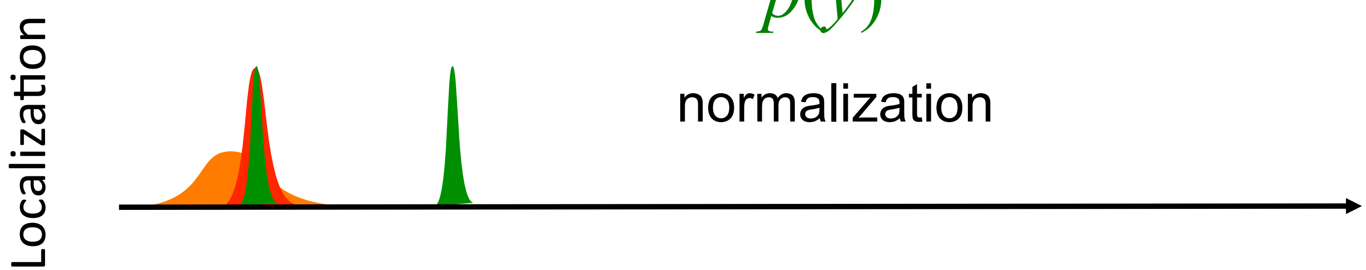
sensor match

new distribution

previous distribution

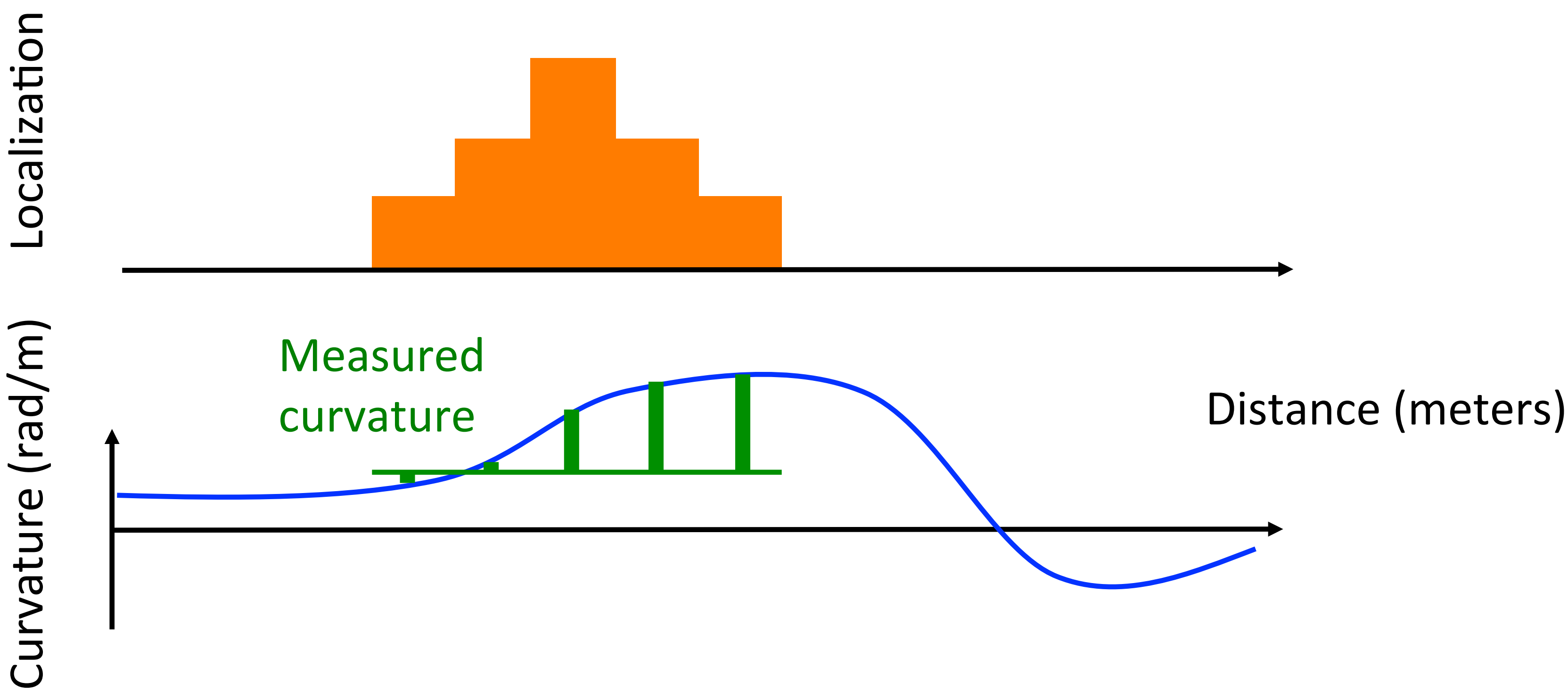
$$p(x|y) = \frac{p(y|x)}{p(y)} p(x)$$

normalization



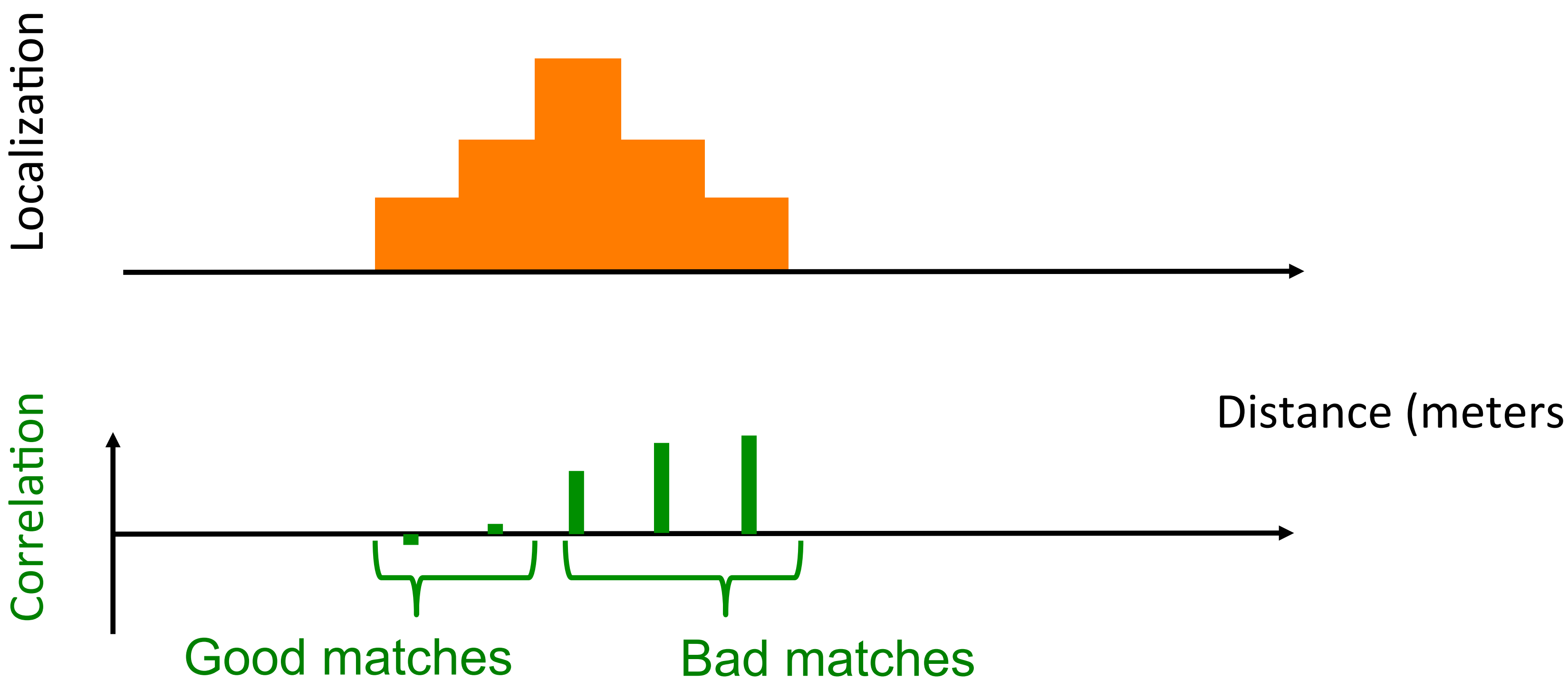
# 3. Localization

- Justin's strategy:
  - Distribution is discretized into five points
  - For each point, compare sensor reading and map:



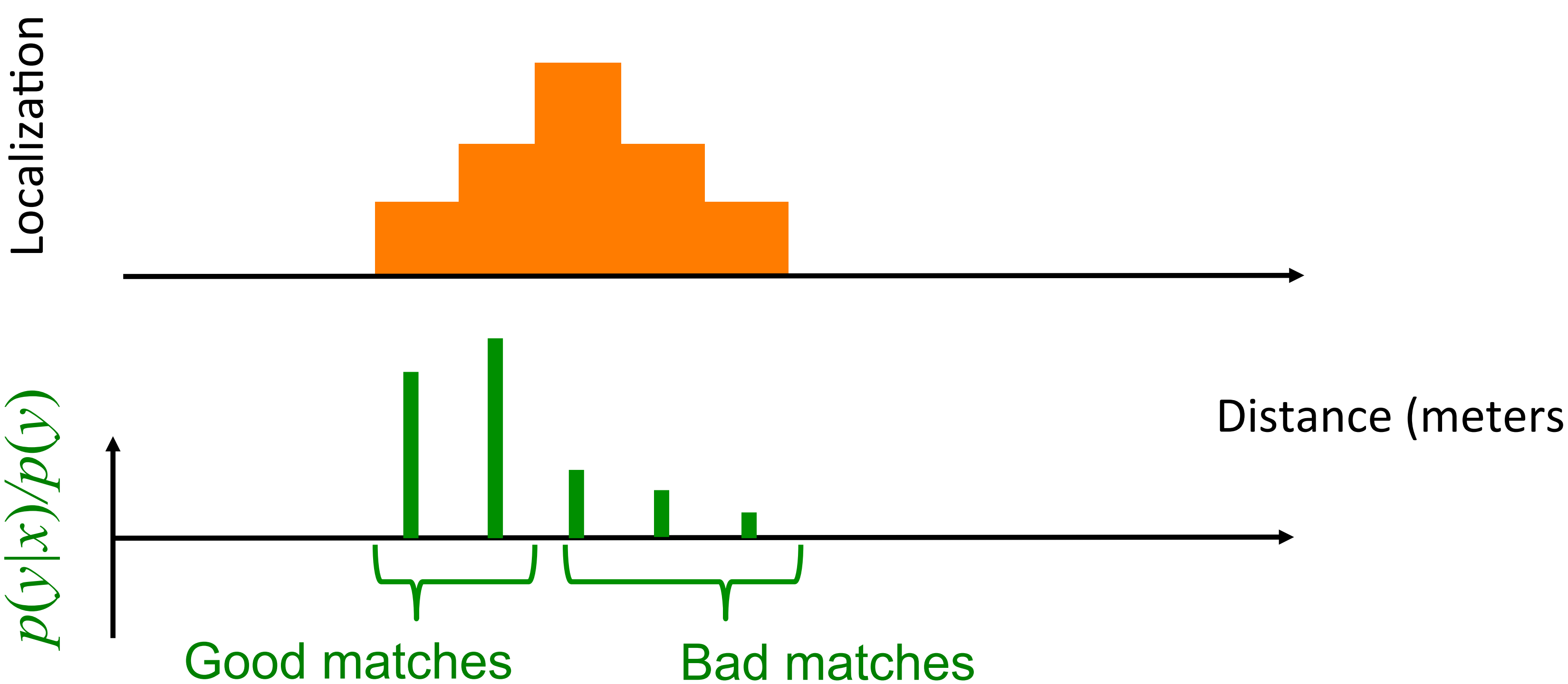
# 3. Localization

- Justin's strategy:
  - Distribution is discretized into five points
  - Heuristic likelihood of $p(y|x)$



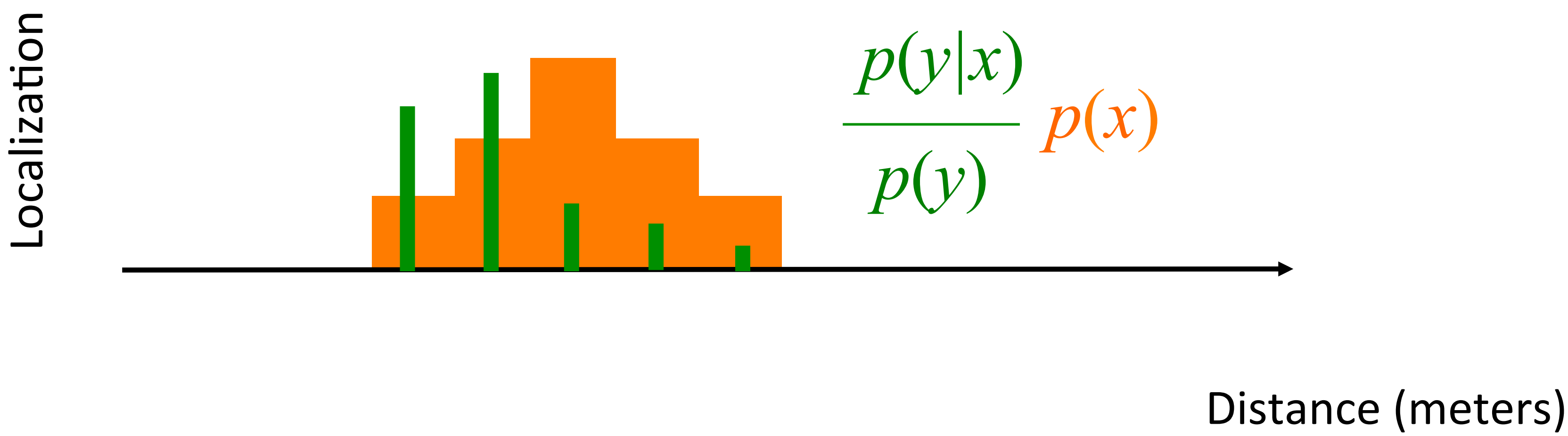
# 3. Localization

- Justin's strategy:
  - Distribution is discretized into five points
  - Heuristic likelihood of $p(y|x)$



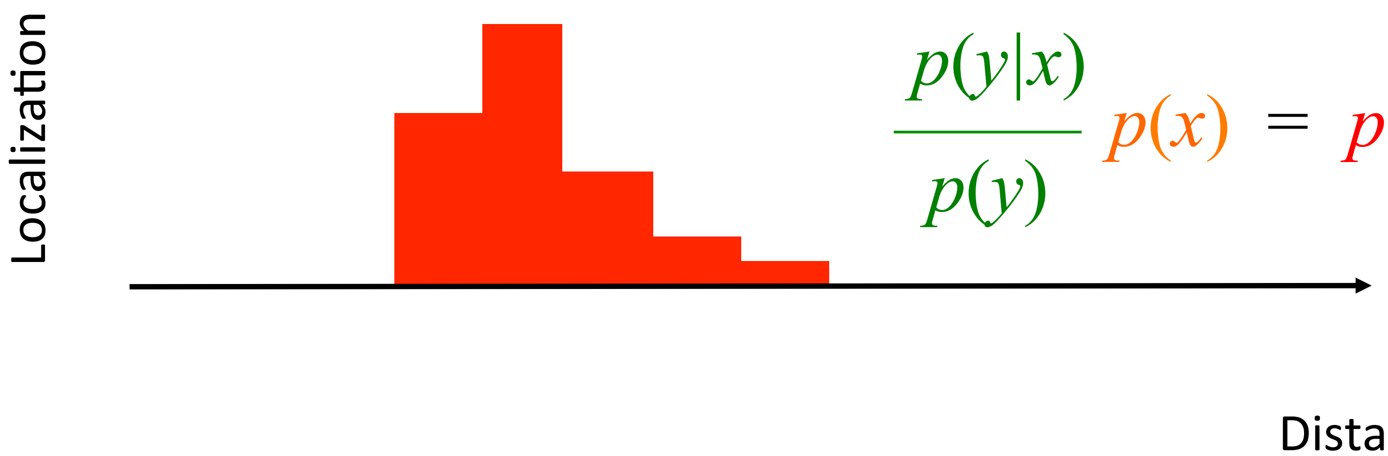
# 3. Localization

- Justin's strategy:
  - Distribution is discretized into five points
  - For each point, apply Bayes rule:



# 3. Localization

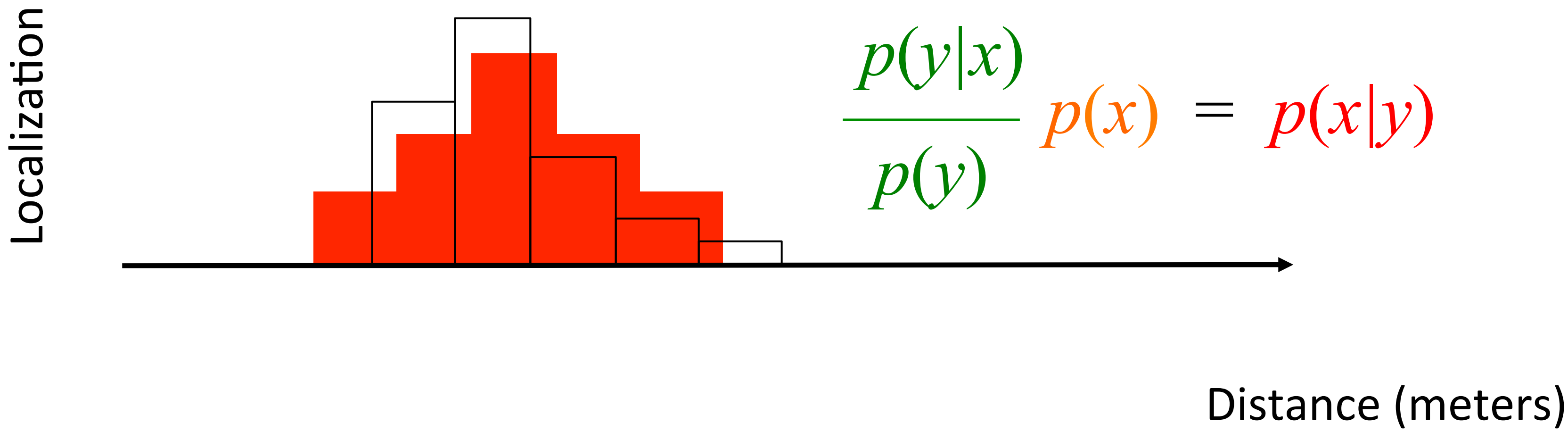
- Justin's strategy:
  - Distribution is discretized into five points
  - For each point, apply Bayes rule:




$$\frac{p(y|x)}{p(y)}\,p(x) = p(x|y)$$

# 3. Localization

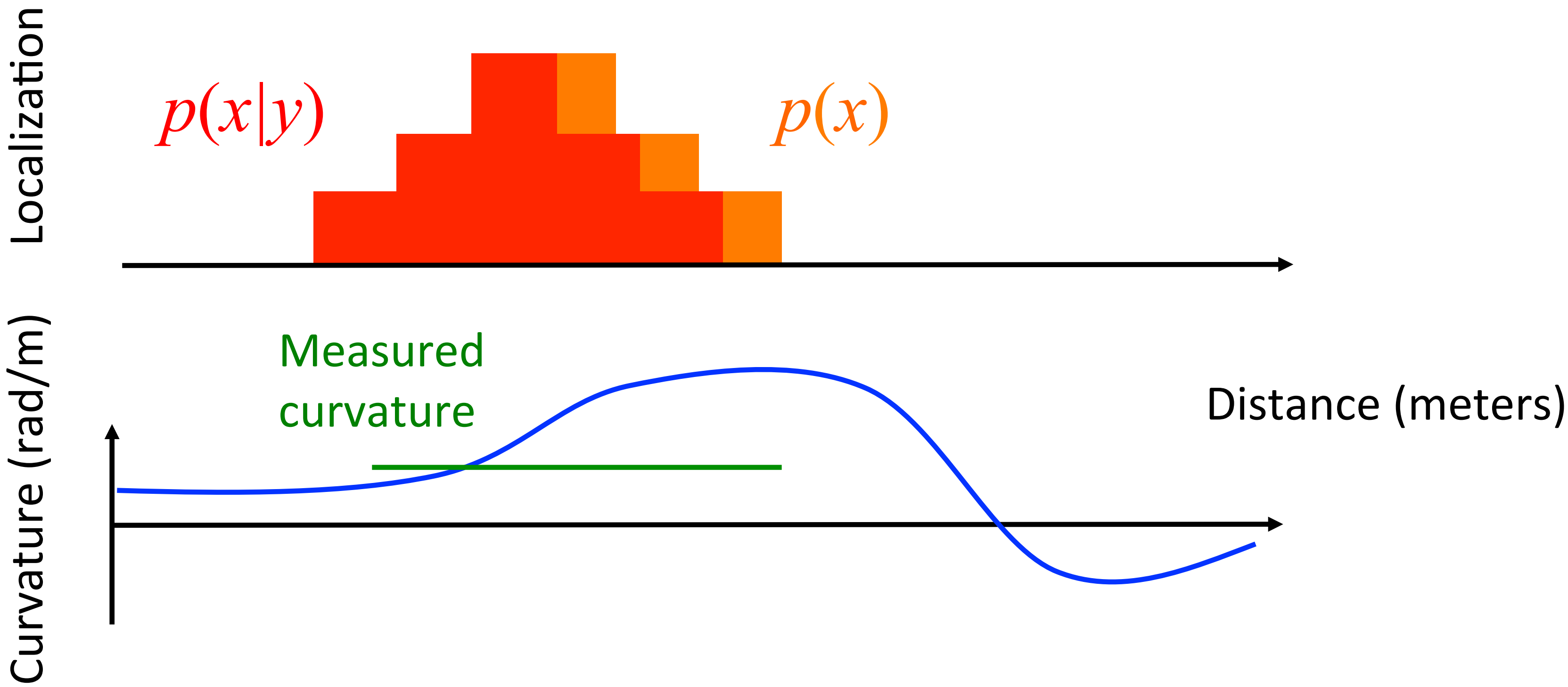
- Justin's strategy:
  - Distribution is discretized into five points
  - Update distribution (note I used only the mean and ignored the standard deviation as a hack)




$$\frac{p(y|x)}{p(y)}\,p(x) = p(x|y)$$

# 3. Localization

- Justin's strategy:
  - Distribution is discretized into five points
  - Update distribution (note I used only the mean and ignored the standard deviation as a hack)



# 3. Localization

- More principled approaches (fewer hacks):
  - Kalman filters (with adjustment for nonlinearity):
    - **Extended Kalman filter** (using linearization)
    - **Unscented Kalman filter** (principled sampling)
  - **Particle filters** (many random samples)
  - (Justin's approach uses heuristic hacky sampling)

  - All methods use the two step process:
  1. Use dynamics to predict location
  2. Update location with sensor correlation (Bayes rule)

# 3. Localization

- For more information, see presentations like: https://www.cs.cmu.edu/~motionplanning/lecture/Chap9-Bayesian-Mapping_howie.pdf