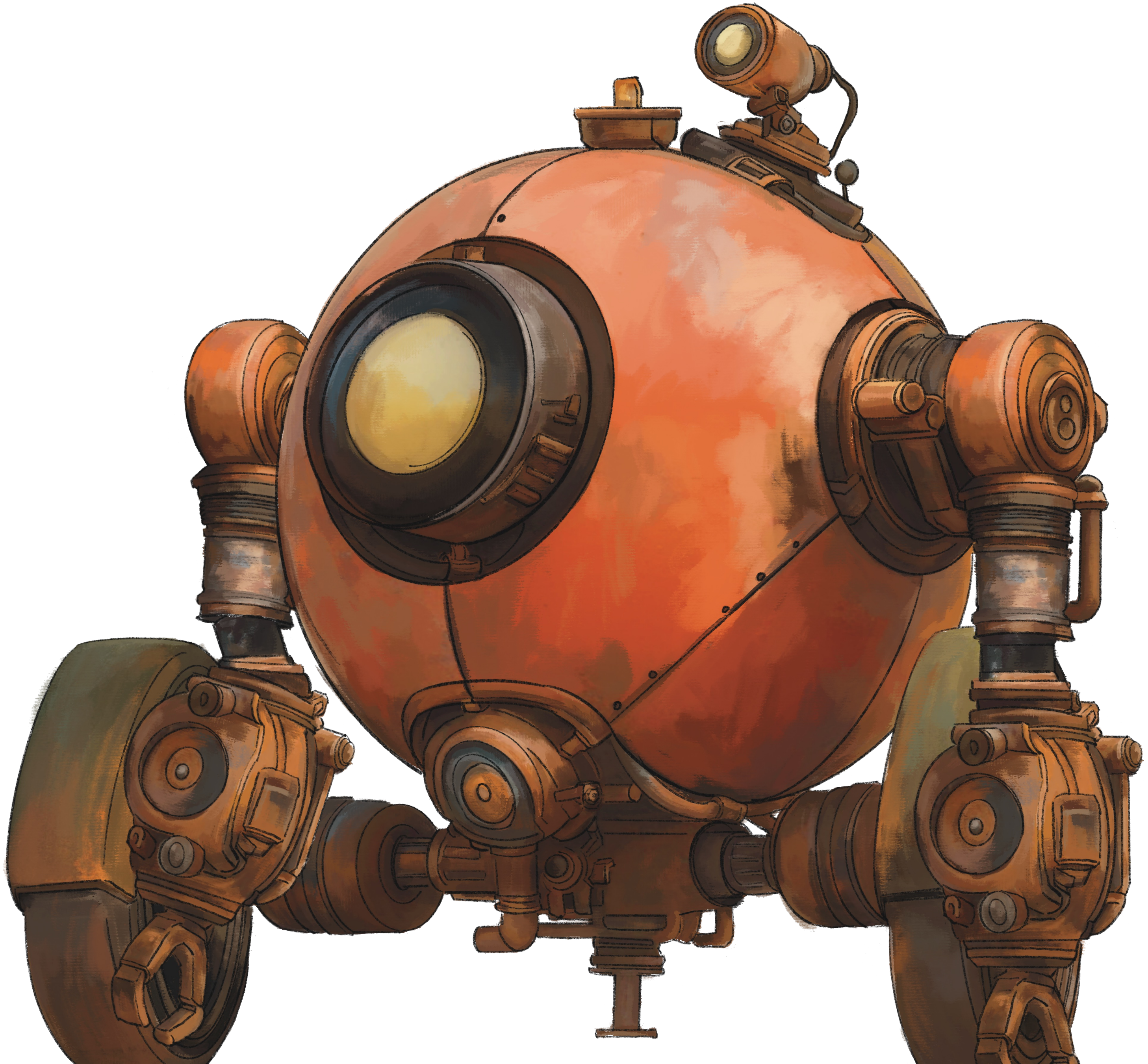
CS 3630, Fall 2025

Lecture 12:

A Logistics Robot:

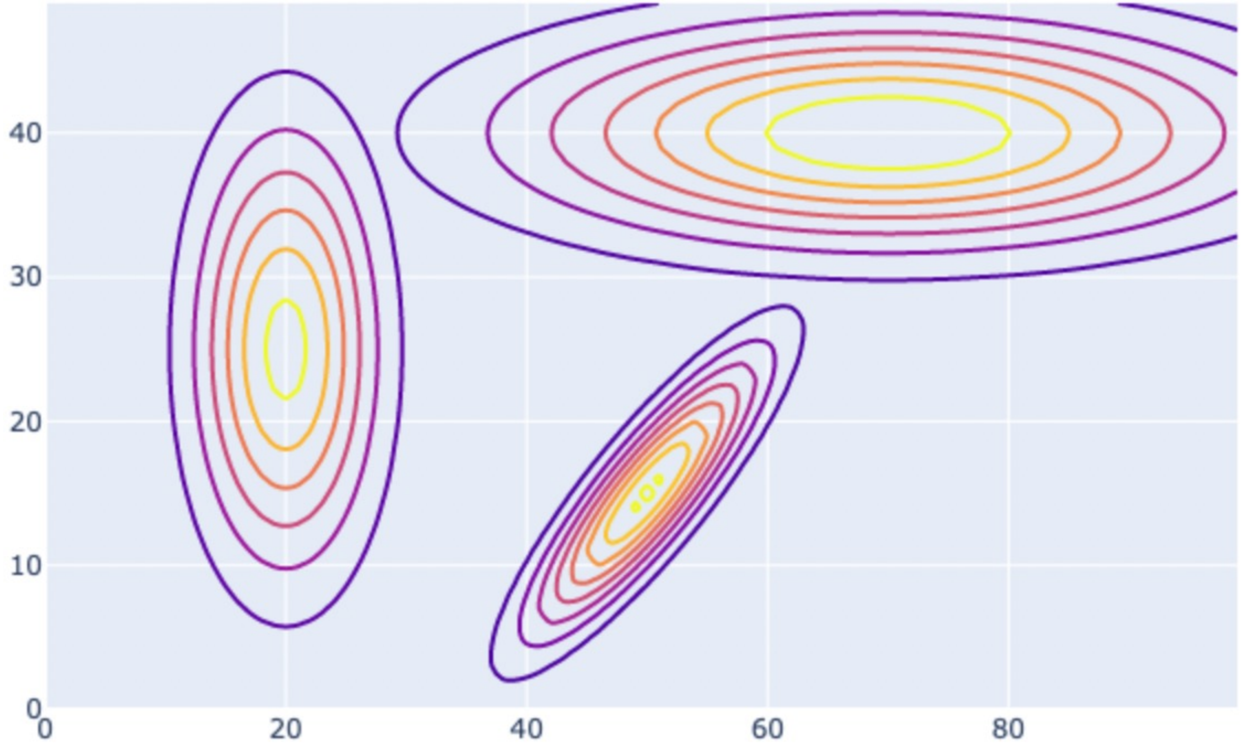
Uncertainty in Actions

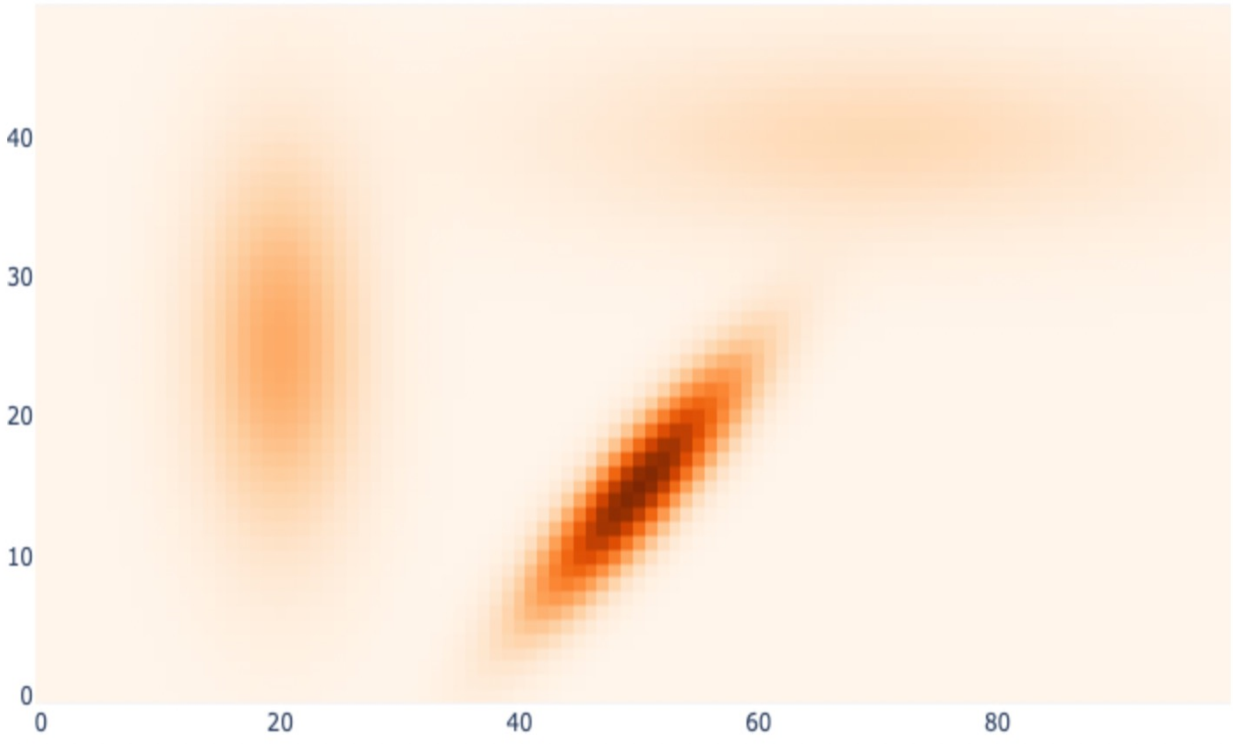


Recap

Multivariate Gaussians and Finite Elements

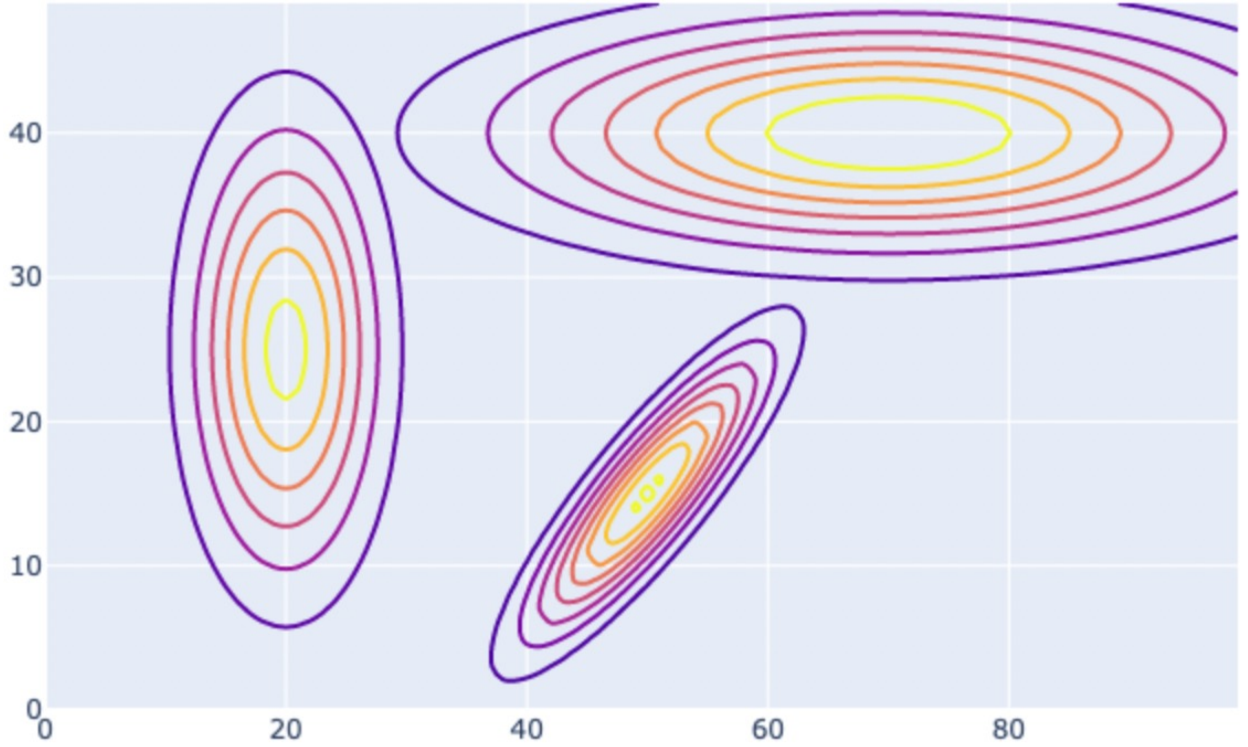
- Just chop up 2D spaces into a 2D grid of finite cells or “elements”



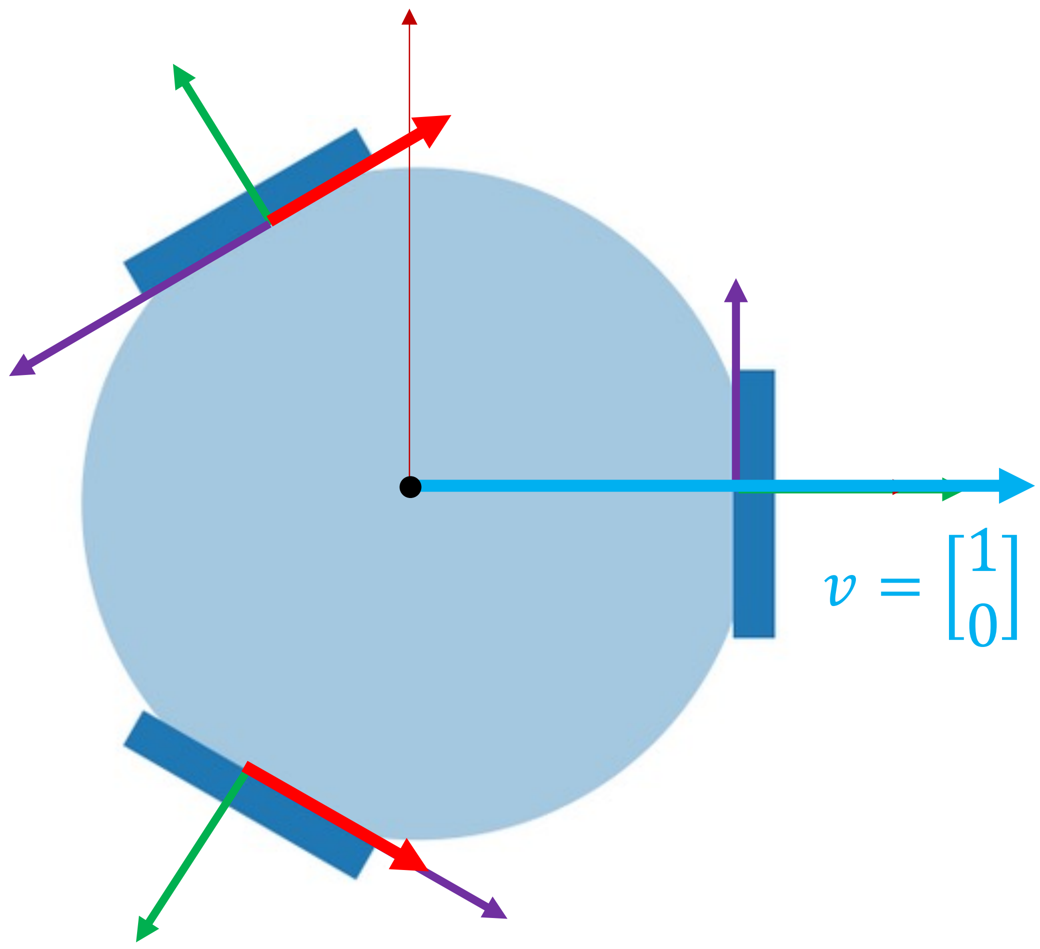


Sampling-based Representation

- Simple, efficient alternative
- Scales with “typical set”



Omniwheels: simple 2x3 Jacobians!



$$\begin{bmatrix} \omega^1 \\ \omega^2 \\ \omega^3 \end{bmatrix} = \frac{1}{r} \begin{bmatrix} 0 & 1 \\ -0.866 & -0.5 \\ 0.866 & -0.5 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \frac{1}{r} \begin{bmatrix} 0 \\ -0.866 \\ 0.866 \end{bmatrix}$$

Control Uncertainty

- Now that we have model for omni-wheel robot kinematics, we can develop a model for uncertainty in the robot’s motion.
- We’ll start with a 1-D robot, and develop the necessary probability theory to model and propagate various types of uncertainty (uniform and Gaussian noise in the motion)
- Once we understand the basics, we’ll extend the results to the 2-D case (motion in the plane).
- We’ll use multivariate Gaussian random variables to model noise/disturbances in the motion model.

Discrete Time Motion Model

- The control input for our robot is a linear velocity v.
- This is converted to angular velocities for each wheel.
- We could model the motion of the robot using a differential equation: $\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u})$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} v_x \\ v_y \end{bmatrix}$$

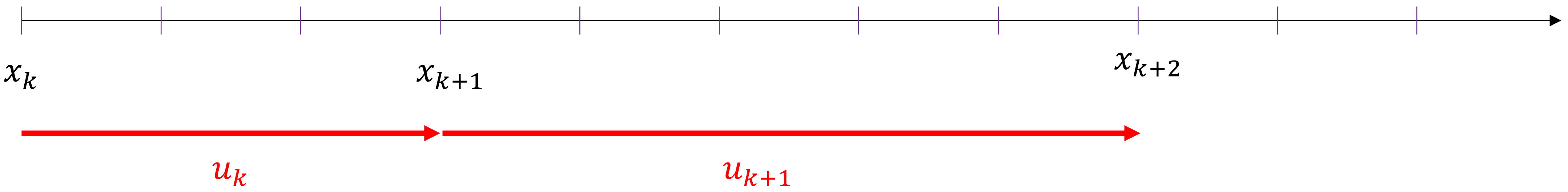
- It's much simpler to use a discrete time model for the position of the robot:

$$\begin{bmatrix} x_{t+1} \\ y_{t+1} \end{bmatrix} = \begin{bmatrix} x_t + v_x \Delta T \\ y_t + v_y \Delta T \end{bmatrix} = \begin{bmatrix} x_t + u_x \\ y_t + u_y \end{bmatrix}$$

- If the motion of the robot happened to be deterministic and error-free, this would be all we need.
- We'll assume that the motion model is stochastic and show how to model uncertainty using continuous probability density functions.

Motion Model – the 1-D Case.

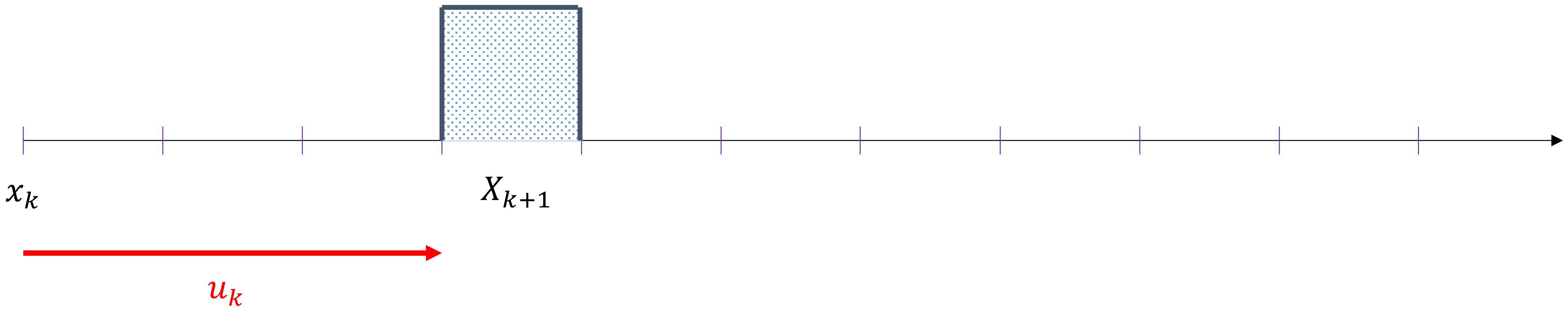
- Wheeled mobile robot that is constrained to move along a single line (e.g., a robot on a track, or a robot following a magnetic guidewire in the floor).
- We will define the control input as $u_k = v\Delta T$, i.e., we command the robot to move along the track with velocity v for an amount of time ΔT.
- In the absence of uncertainty, the state equation is simple: $x_{k+1} = x_k + u_k$
- If we execute a sequence of actions, u_k, u_{k+1} we arrive to $x_{k+2} = x_k + u_k + u_{k+1}$



➢ If there's no uncertainty in the motion model, predicting future states is pretty easy.

Motion Model – the 1-D Case.

- Consider the motion model $x_{k+1} = x_k + u_k + \eta_k$, and let $\eta_k \sim U(0,1)$
- Suppose x_k is known.
- What can we say about x_{k+1}?



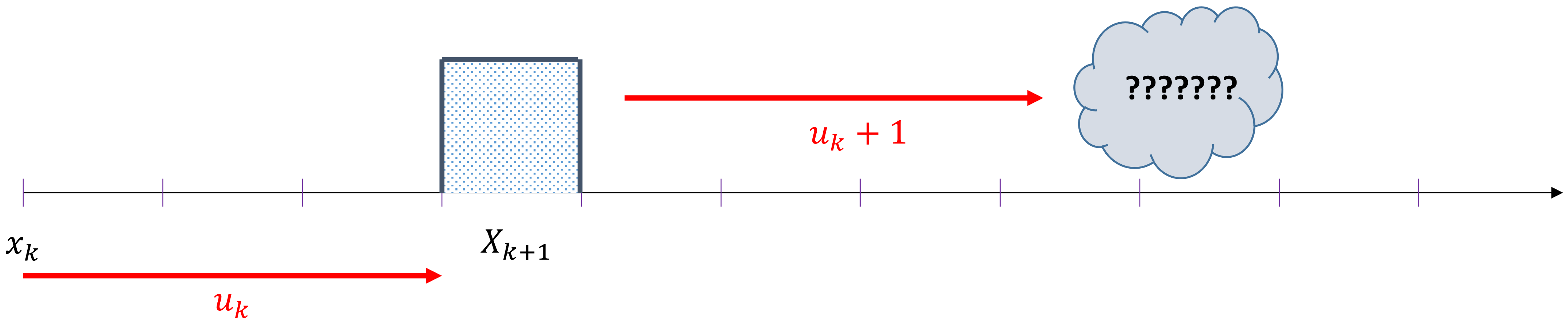
The next state is a random variable with uniform distribution

$$X_{k+1} \sim U(x_k + u_k, x_k + u_k + 1)$$

Motion Model – the 1-D Case.

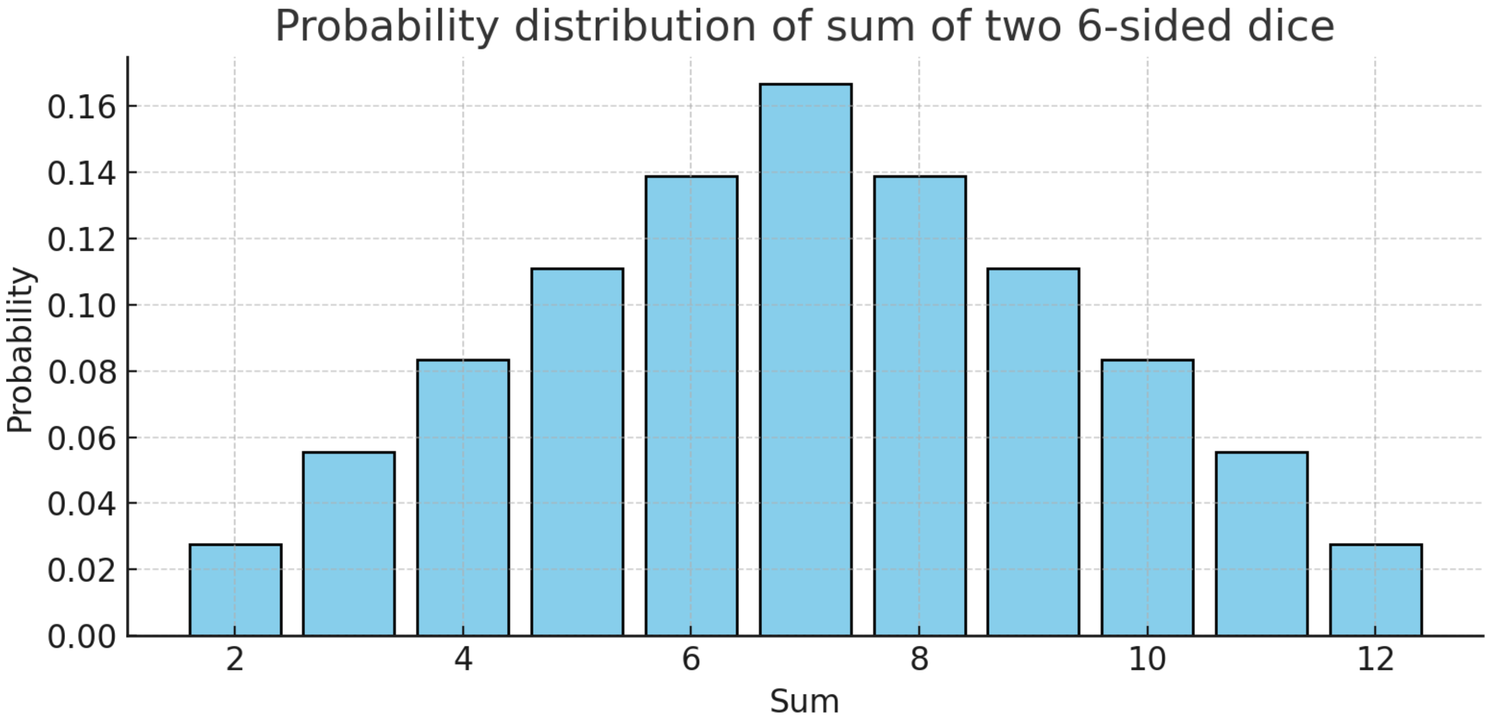
- That was so simple!!
- What happens after two time steps?

$$x_{k+2} = x_k + u_k + \eta_k + u_{k+1} + \eta_{k+1} = (x_k + u_k + u_{k+1}) + (\eta_k + \eta_{k+1})$$

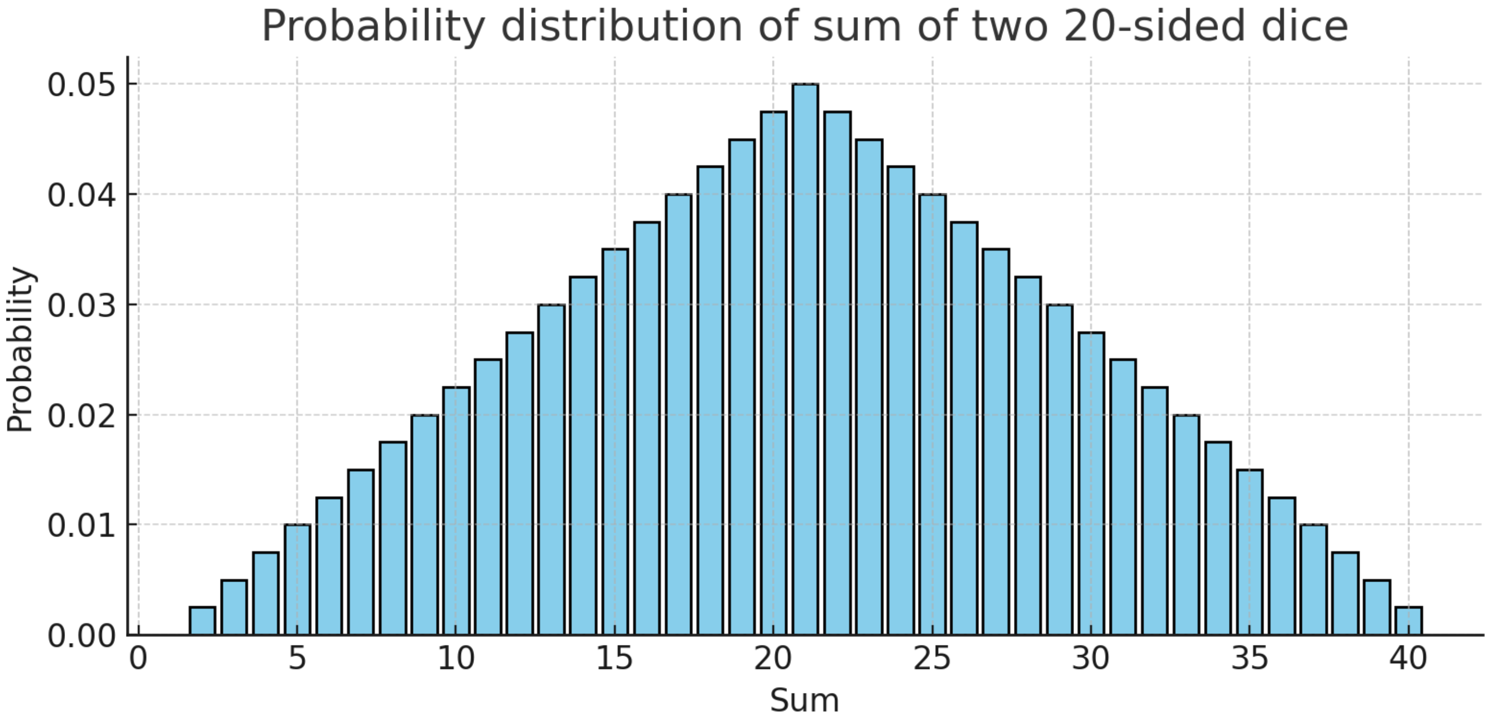


- The term $x_k + u_k + u_{k+1}$ is completely deterministic (and easy to compute).
- The term $\eta_k + \eta_{k+1}$ is completely stochastic, and somewhat mysterious.
- We need to determine the probability distribution of a sum of random variables.

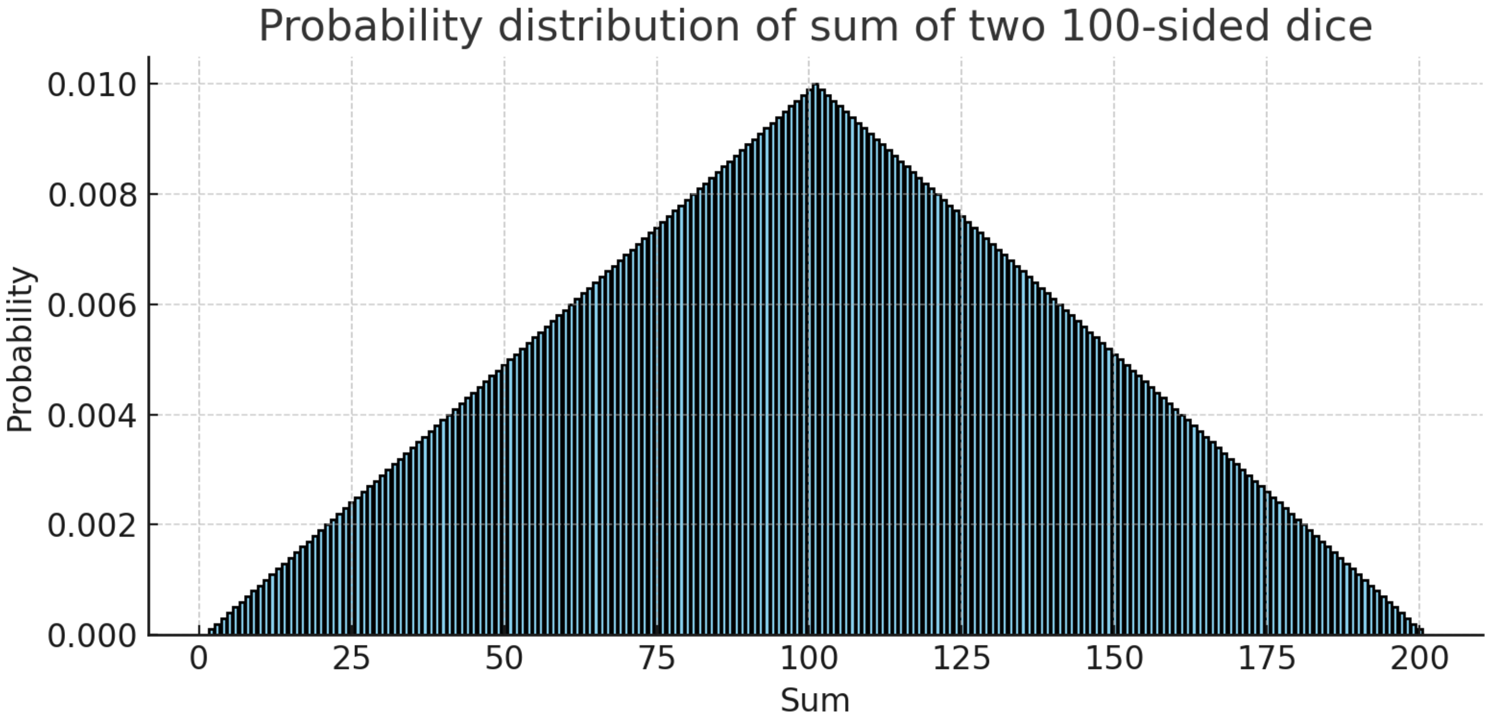
Might help to think about sum of two dice



Might help to think about two n-sided dice



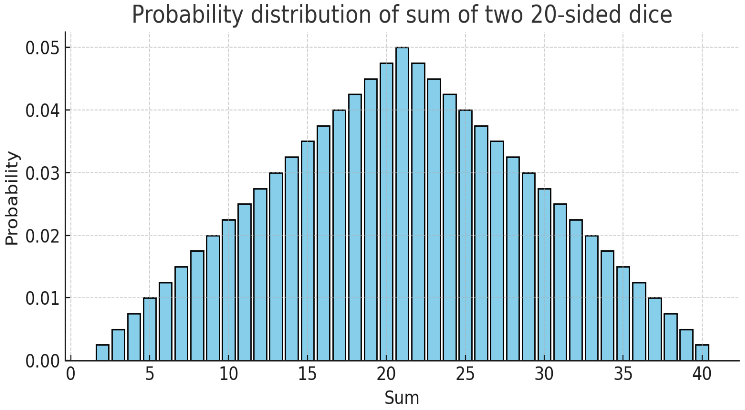
Might help to think about two n-sided dice



Sum of Two Random Variables

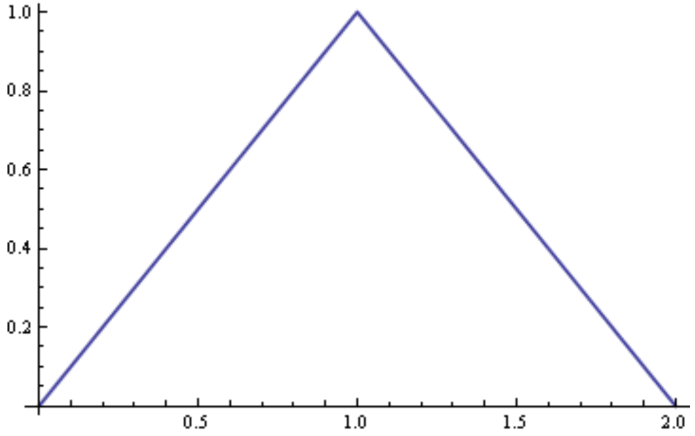
For $s = d_1 + d_2$ if $d_k \sim U(0,1)$, the probability density function for η_{S2} is:

$$P(s \mid n) = \begin{cases} \frac{s-1}{n^2}, & 2 \leq s \leq n+1, \\ \frac{2n+1-s}{n^2}, & n+1 < s \leq 2n, \\ 0, & \text{otherwise.} \end{cases}$$



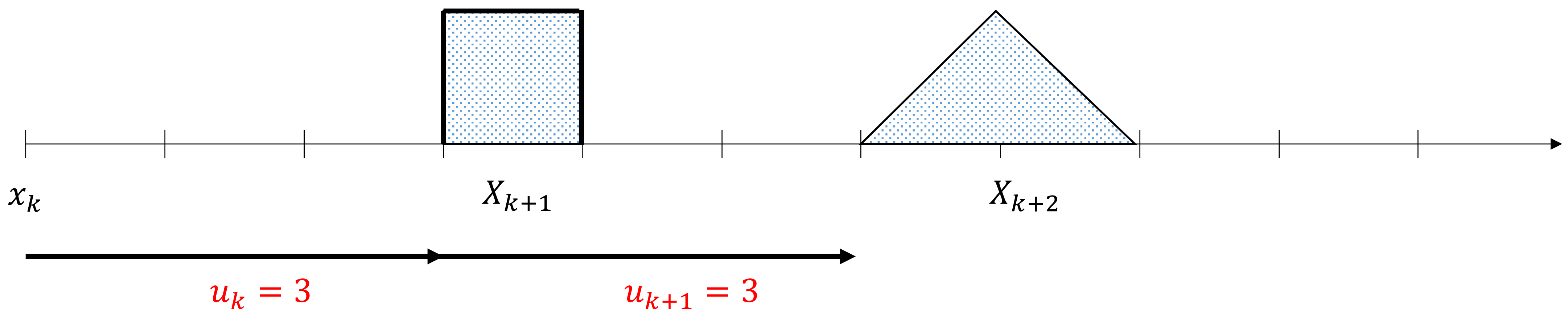
For $\eta_{S2} = \eta_1 + \eta_2$ if $\eta_k \sim U(0,1)$, the probability density function for η_{S2} is:

$$f_{\eta_{S2}}(\alpha) = \begin{cases} \alpha & 0 \leq \alpha \leq 1 \\ 2 - \alpha & 1 \leq \alpha \leq 2 \end{cases}$$



Motion Model – the 1-D Case.

After two time steps, $x_{k+2} = x_k + u_k + \eta_k + u_{k+1} + \eta_{k+1} = (x_k + u_k + u_{k+1}) + (\eta_k + \eta_{k+1})$



- Both of X_{k+1} and X_{k+2} are random variables.
- ***They do not have the same probability distribution!!!***

The Sum of n i.i.d. Uniform Random Variables

Let the random variable $\eta_{Sn} = \eta_1 + \ldots + \eta_n$ be the sum of n random variables.

The pdf for η_{Sn} is called the *Irwin-Hall distribution*.

The Irwin–Hall distribution is the continuous probability distribution for the sum of *n* independent and identically distributed *U*(0, 1) random variables:

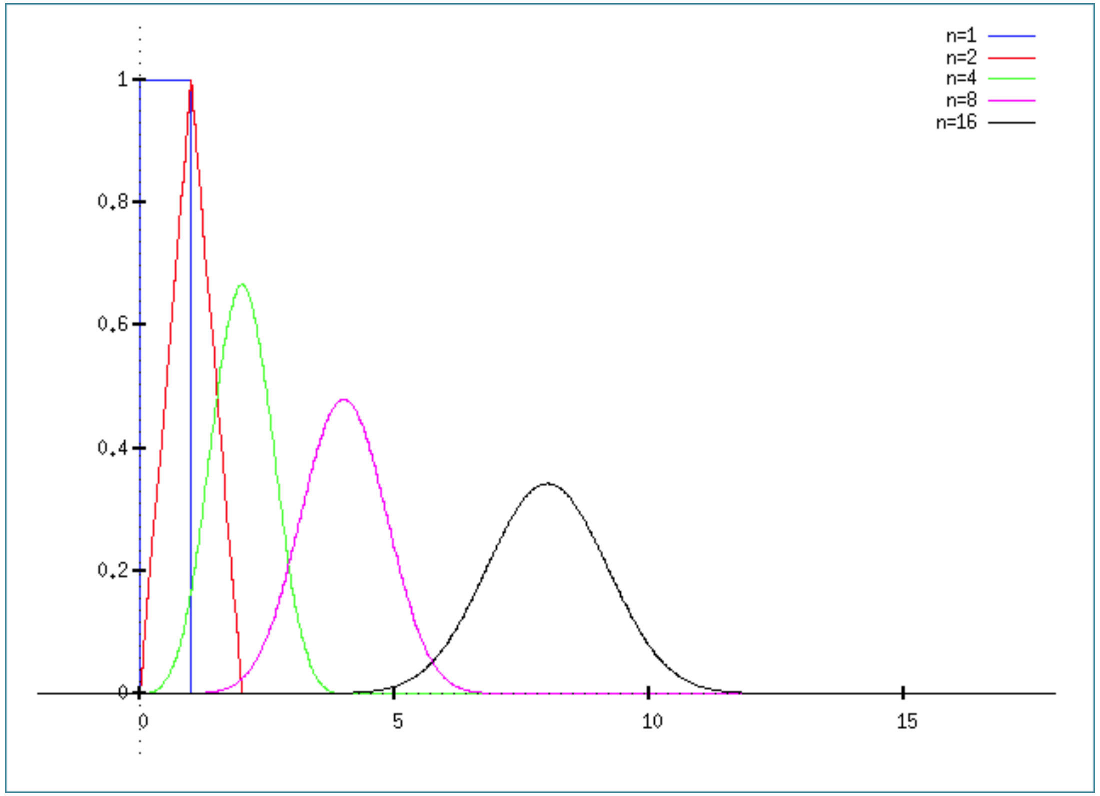
$$X = \sum_{k=1}^{n} U_k.$$

The probability density function (pdf) is given by

$$f_X(x;n) = \frac{1}{2(n-1)!} \sum_{k=0}^{n} (-1)^k \binom{n}{k} (x-k)^{n-1} \operatorname{sgn}(x-k)$$

where sgn(*x* − *k*) denotes the sign function:

$$\operatorname{sgn}(x-k) = \begin{cases} -1 & x < k \\ 0 & x = k \\ 1 & x > k. \end{cases}$$



[wikipedia]

The Sum of n i.i.d. Uniform Random Variables

This is a nice piece of trivia, but should we really care about this?

YES! As n becomes large, $f_{\eta_{Sn}}$ approaches a Gaussian distribution.

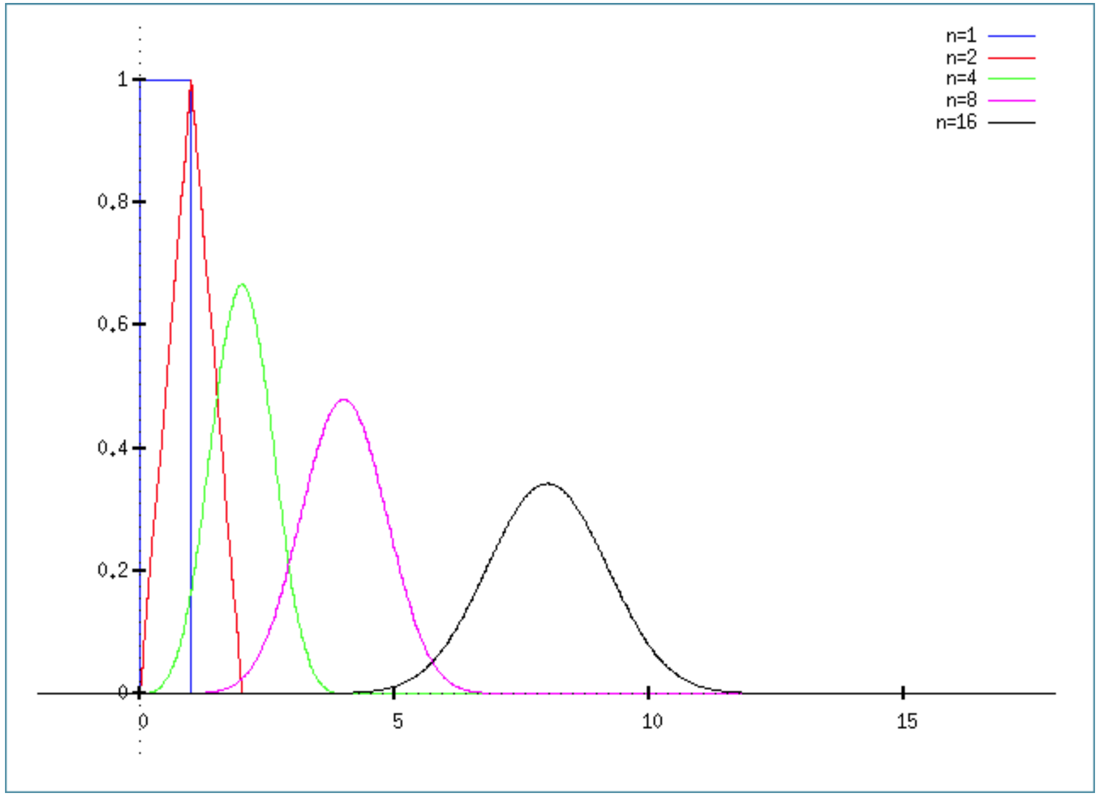
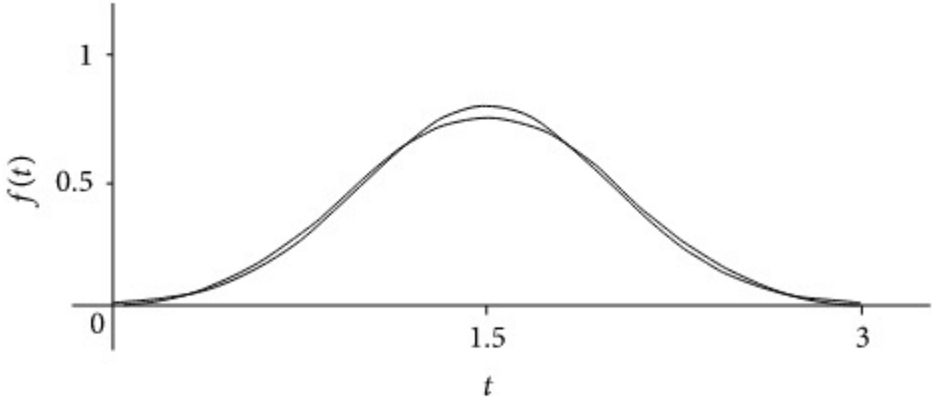


Figure 1

Irwin-Hall distribution with $n = 3$ and the matching normal distribution with mean 3/2 and variance 1/4.

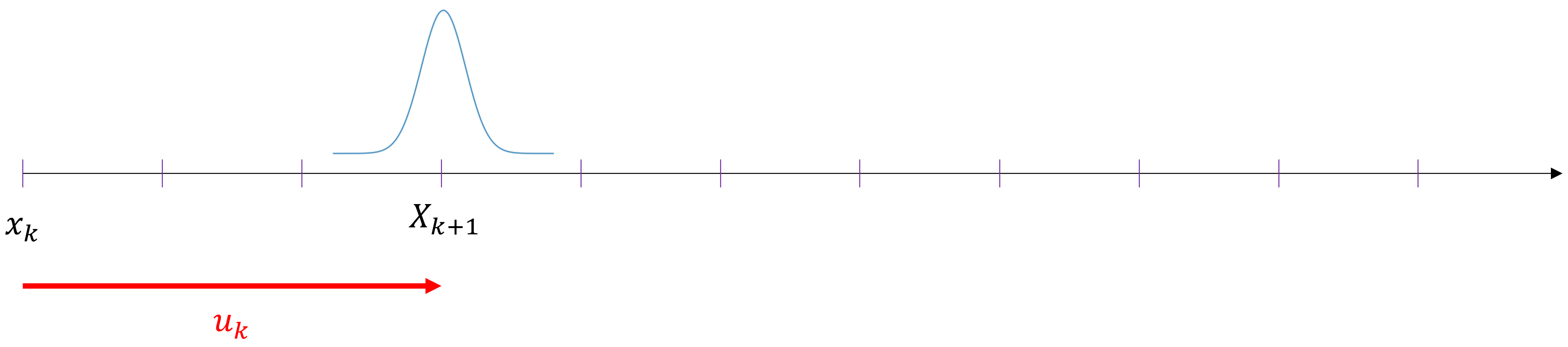


Even for $n = 3$ we can start to see the similarity.

<u>In general, when we add together a bunch of i.i.d. random variables, things start to look Gaussian before long.</u>

1D Motion Model with Gaussian Noise

- Consider again the motion model $x_{k+1} = x_k + u_k + \eta_k$, but now let $\eta_k \sim N(0, \sigma^2)$, with all η_k independent.
- Suppose x_k is known.
- What can we say about x_{k+1}?

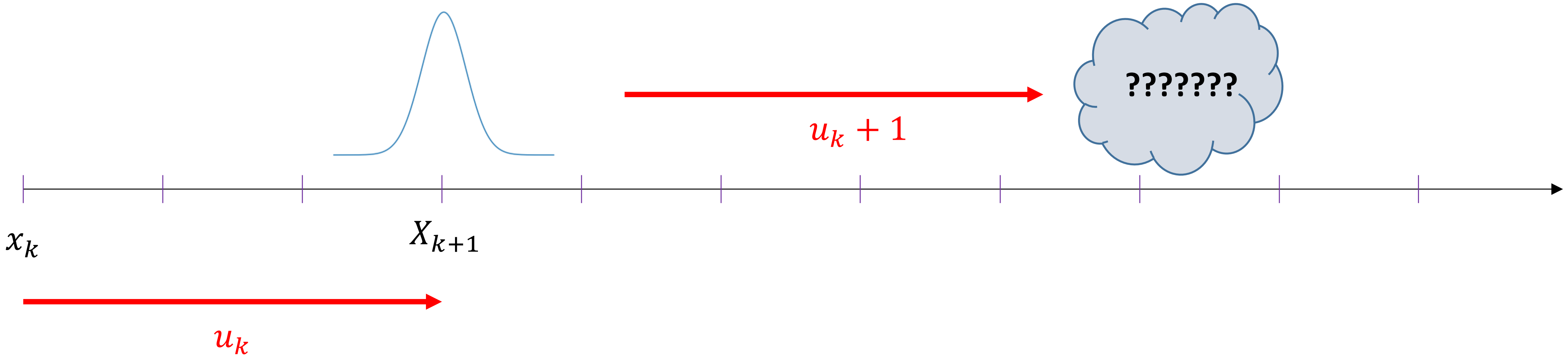


- The next state is a random variable with Gaussian distribution $X_{k+1} \sim N(x_k + u_k, \sigma^2)$.
- $E[X_{k+1}] = x_k + u_k$
- The variance of X_{k+1} is exactly the variance in the noise.

1D Motion Model with Gaussian Noise

- Not too difficult…
- What happens after two time steps?

$$x_{k+2} = x_k + u_k + \eta_k + u_{k+1} + \eta_{k+1} = (x_k + u_k + u_{k+1}) + (\eta_k + \eta_{k+1})$$

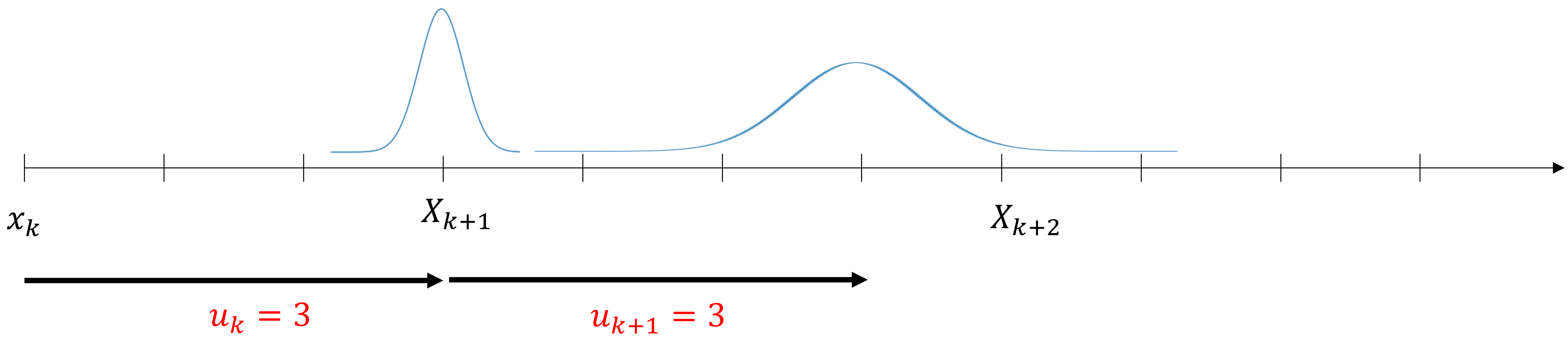


- The term $x_k + u_k + u_{k+1}$ is completely deterministic (and easy to compute).
- The term $\eta_k + \eta_{k+1}$ is completely stochastic, and somewhat mysterious.
- We need to determine the probability distribution of a sum of Gaussian random variables.

1D Motion Model with Gaussian Noise

After two time steps, $x_{k+2} = x_k + u_k + \eta_k + u_{k+1} + \eta_{k+1} = (x_k + u_k + u_{k+1}) + (\eta_k + \eta_{k+1})$

- Both of X_{k+1} and X_{k+2} are Gaussian random variables.
- ***The do not have the same variance!!!***



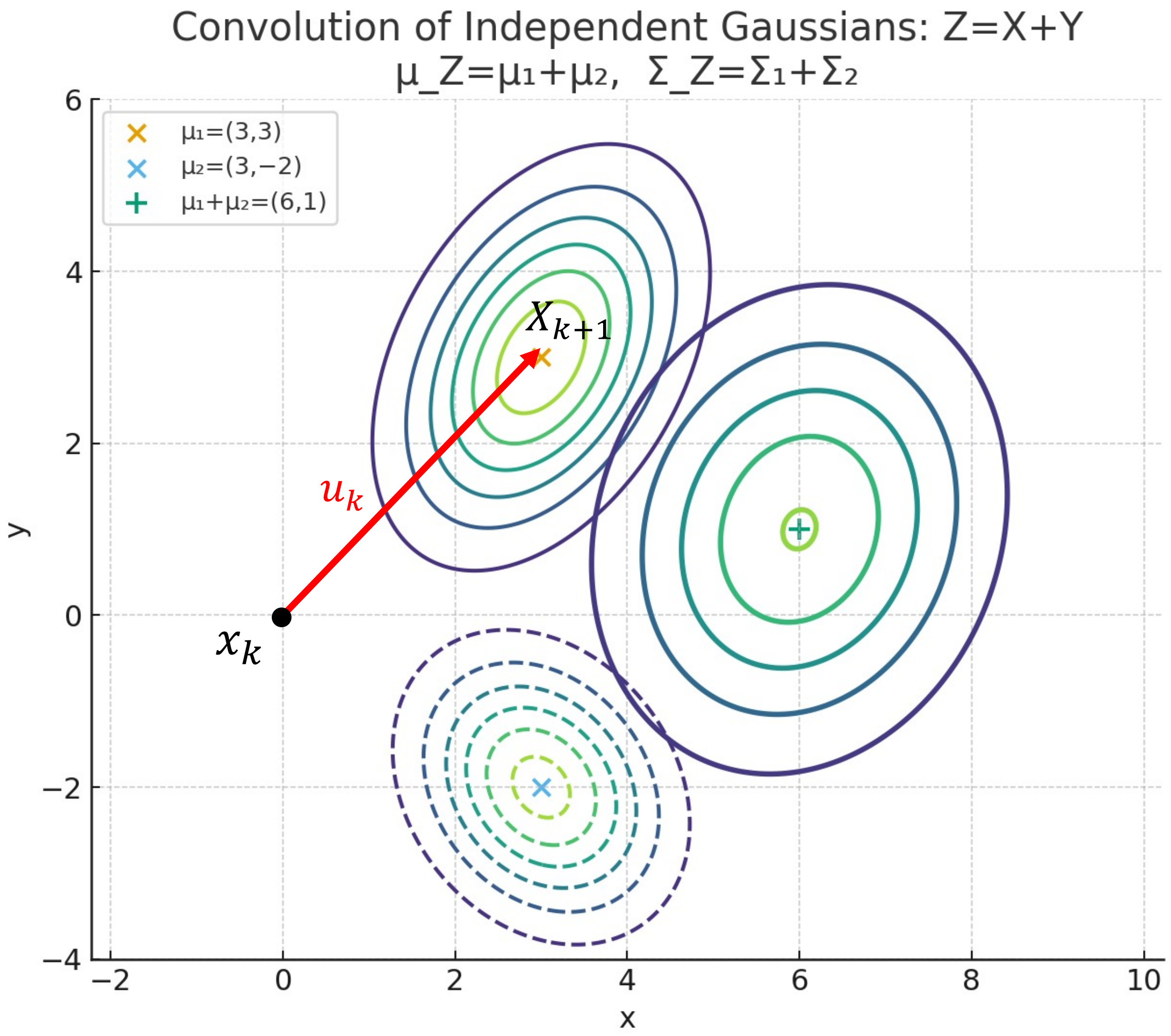
Bivariate Gaussians

For our motion model, we'll use

$$x_{k+1} = x_k + u_k + \eta_k$$

with $x_{k+1}, x_k, u_k, \eta_k \in \mathbb{R}^2$ and $\eta_k \sim N(0, \Sigma)$.

X_{k+1} is a bivariate Gaussian,
$X_{k+1} \sim N(x_k + u_k, \Sigma)$



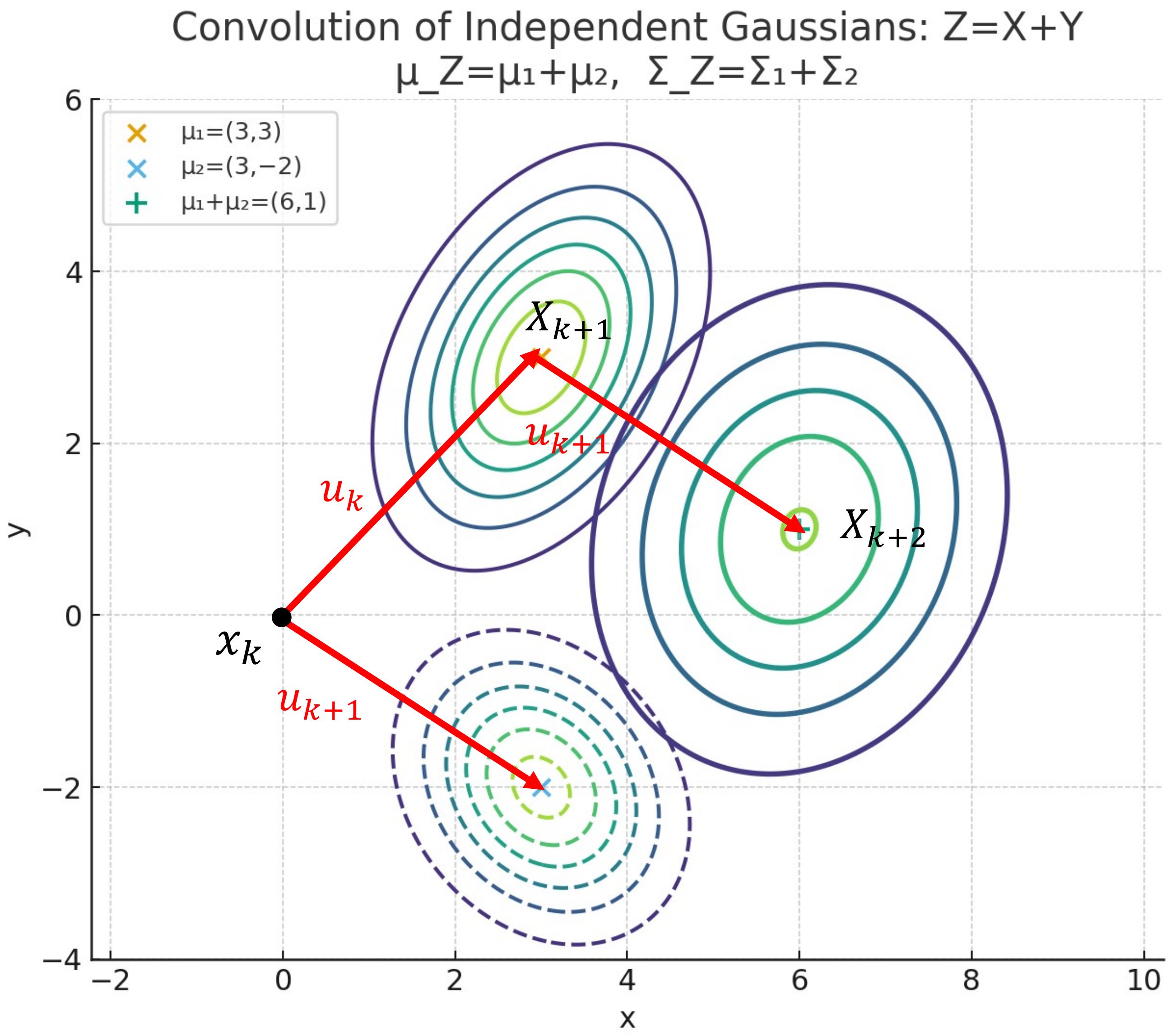
Bivariate Gaussians

For our motion model, we'll use

$$x_{k+1} = x_k + u_k + \eta_k$$

with $x_{k+1}, x_k, u_k, \eta_k \in \mathbb{R}^2$ and $\eta_k \sim N(0, \Sigma)$.

X_{k+2} is a bivariate Gaussian,
$X_{k+2} \sim N(x_k + u_k + u_{k+1}, \Sigma_1 + \Sigma_2)$



Multiple Time Steps

Conceptually, there's nothing new here.

➢ *Each time step adds a bit of Gaussian noise to the control input, introducing uncertainty that increases with the number of steps.*

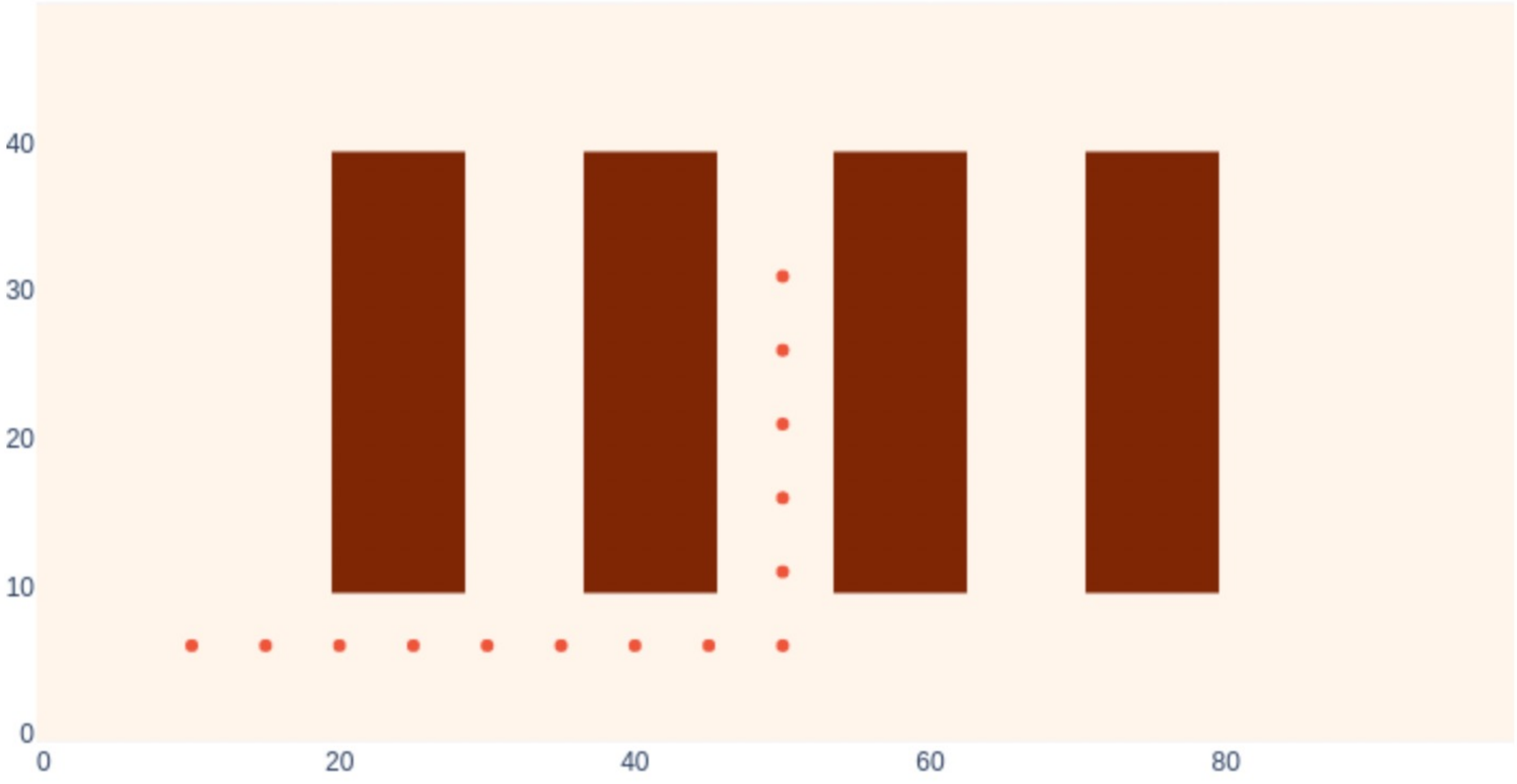
Mathematically, things are just as easy: add the covariance matrices!

We will use two numerical methods to propagate uncertainty, and both will be applicable to the case of Gaussian noise in our motion model:

- **Markov Localization**: Divide the world into a grid and keep track of the probability mass that arrives to each grid cell as the robot moves.
- **Monte Carlo Localization**: Simulate lots of robots (generate samples from the noise distributions to simulate the motion model). The distribution of the simulated robots give insight to the probability distribution associated to the robot's location.

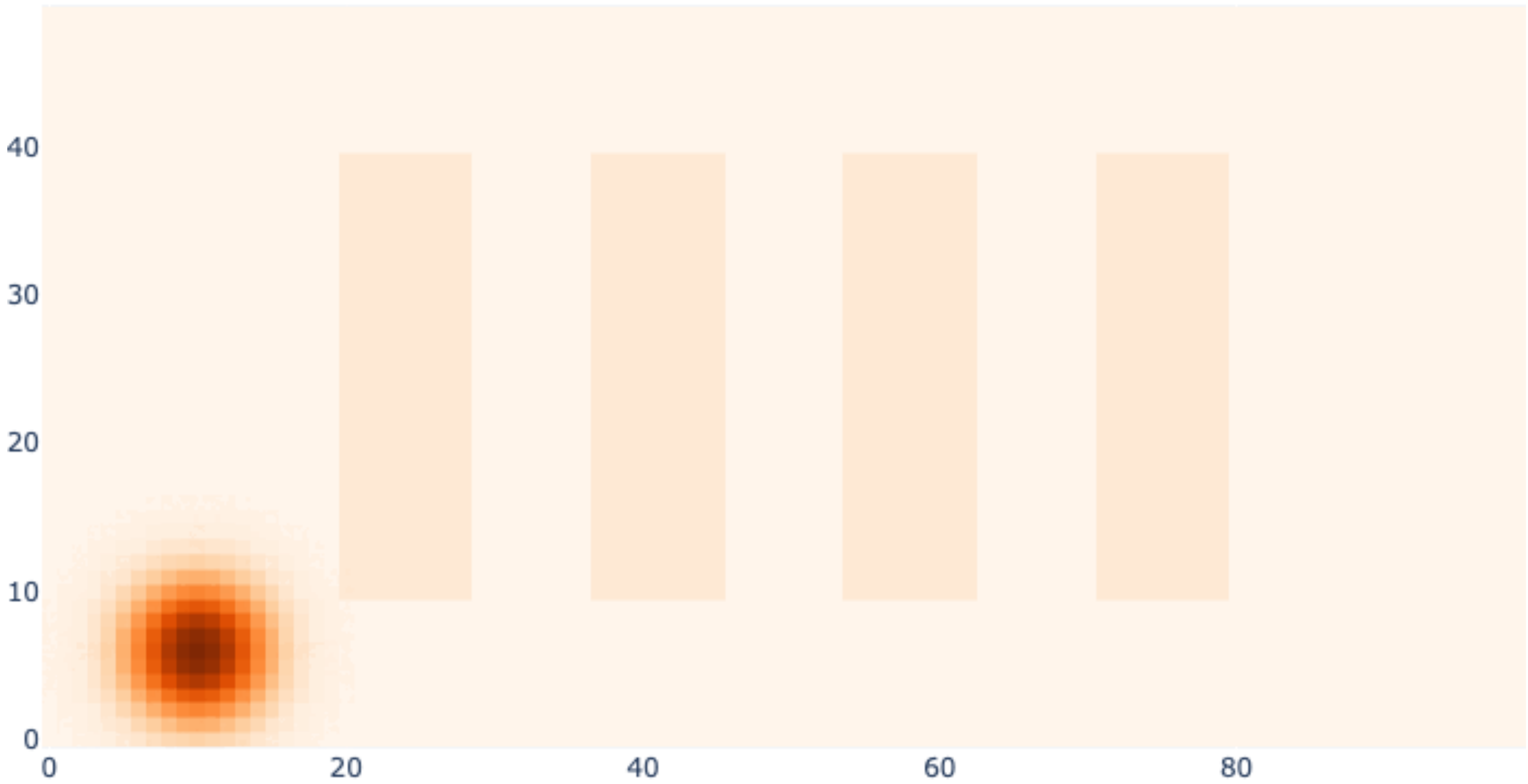
An example (ground truth) trajectory

- Robot starts out in bottom-left, goes right, then up in ”aisle 2”:



Propagation of uncertainty

- Finite elements version:



Next Time…

- Sensor models
- Markov Localization
- Monte Carlo Localization