{Learn, Create, Innovate};

Mobile robots

Localisation in presence of uncertainties

MCR²

Manchester **Robotics**

Motion-based Localisation (Dead Reckoning)

$$\begin{bmatrix} s_{x,k} \\ s_{y,k} \\ s_{\theta,k} \end{bmatrix} = \begin{bmatrix} s_{x,k-1} \\ s_{y,k-1} \\ s_{\theta,k-1} \end{bmatrix} + \begin{bmatrix} \Delta d \cos\left(s_{\theta,k-1}\right) \\ \Delta d \sin\left(s_{\theta,k-1}\right) \\ \Delta\theta \end{bmatrix}$$

Motion-based Localisation (Dead Reckoning)

The pose estimation of a mobile robot is **always associated with some uncertainty** with respect to its state parameters.

From a geometric point of view, the error in differential-drive robots is classified into three groups:

- **Range error:** it is associated with the computation of Δd over time.
- **Turn error:** it is associated with the computation of $\Delta\theta$ over time.
- **Drift error:** it is associated with the difference between the angular speed of the wheels and it affects the error in the angular rotation of the robot.

Motion-based Localisation (Dead Reckoning)

Kinematic model for a differential robot model

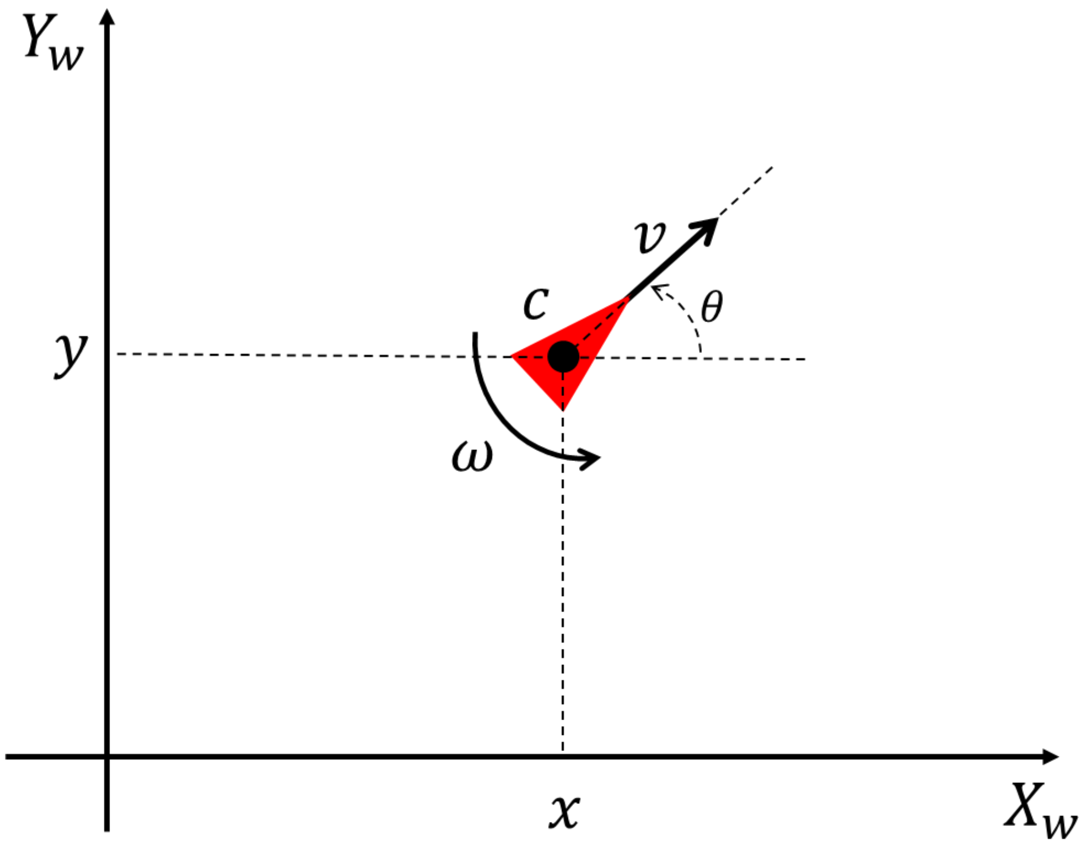
$$\frac{d}{dt}\begin{bmatrix} s_x \\ s_y \\ s_\theta \end{bmatrix} = \begin{bmatrix} \cos(s_\theta) & 0 \\ \sin(s_\theta) & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} v \\ \omega \end{bmatrix}$$

The robot pose

$$\mathbf{s}_k = [s_x \quad s_y \quad s_\theta]^T$$

The robot inputs

$$\mathbf{u}_k = [v \quad \omega]^T$$



Motion-based Localisation (Dead Reckoning)

- If Δt is the sampling time, then it is possible to compute the incremental linear and angular displacements, Δd and $\Delta\theta$, as follows:

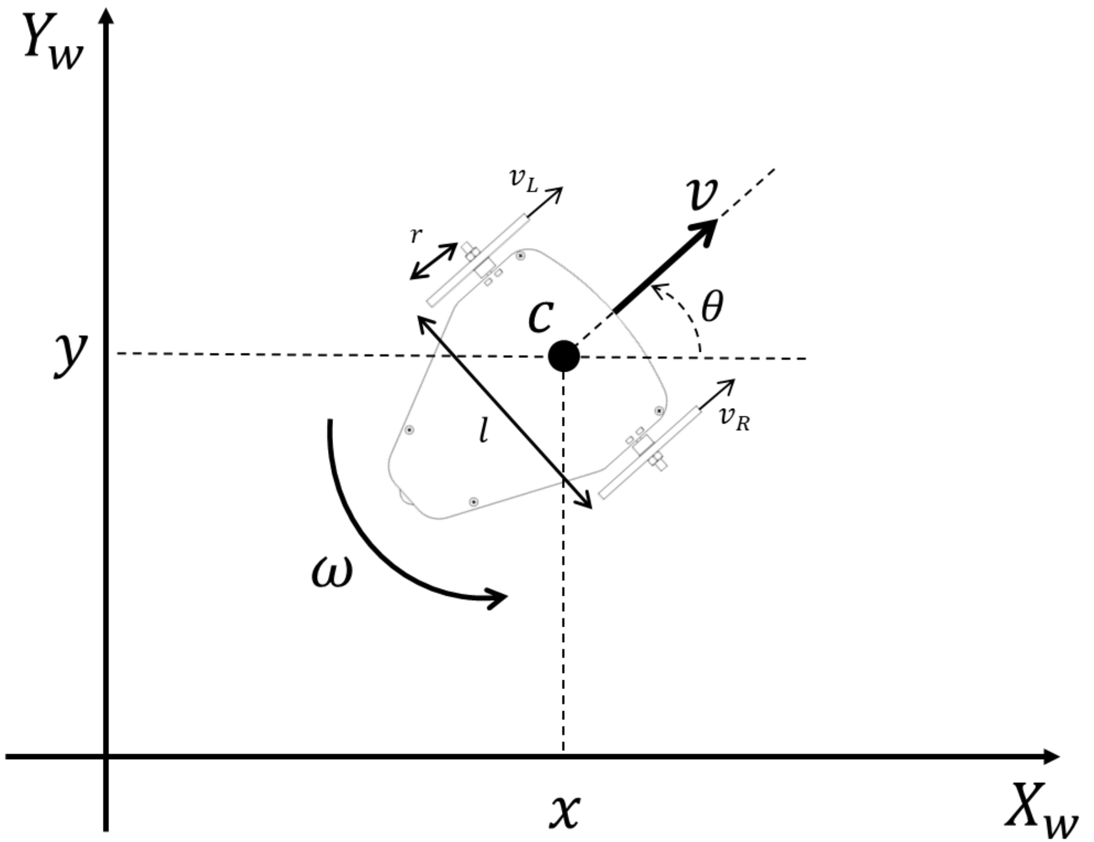
$$\Delta d = v \cdot \Delta t \qquad\qquad \Delta\theta = \omega \cdot \Delta t$$

$$\begin{bmatrix} \Delta d \\ \Delta\theta \end{bmatrix} = \begin{bmatrix} 1/2 & 1/2 \\ 1/l & -1/l \end{bmatrix} \begin{bmatrix} \Delta d_r \\ \Delta d_l \end{bmatrix}$$

To compute the pose of the robot at any given time step, the kinematic model must be numerically integrated.

This approximation follows the **Markov assumption** where the current robot pose depends only on the previous pose and the input velocities.

$$\begin{bmatrix} s_{x,k} \\ s_{y,k} \\ s_{\theta,k} \end{bmatrix} = \begin{bmatrix} s_{x,k-1} \\ s_{y,k-1} \\ s_{\theta,k-1} \end{bmatrix} + \begin{bmatrix} \Delta d \cos(s_{\theta,k-1}) \\ \Delta d \sin(s_{\theta,k-1}) \\ \Delta\theta \end{bmatrix}$$



Motion-based Localisation (Dead Reckoning)

So, in order to make a realistic simulation, we must include the uncertainties in the state-space model.

$$\mathbf{s}_k = \mathbf{h}(\mathbf{s}_{k-1}, \mathbf{u}_k) + \mathbf{q}_k$$

- what is $\mathbf{q}_k$???

Motion-based Localisation (Dead Reckoning)

So, in order to make a realistic simulation, we must include the uncertainties in the state-space model.

$$\mathbf{s}_k = \mathbf{h}(\mathbf{s}_{k-1}, \mathbf{u}_k) + \mathbf{q}_k$$

- what is $\mathbf{q}_k$???

Probabilities, a suitable mathematical tool to represent the uncertainties???

Fundamentals in probabilities

- What is probability?

History of probability

The theory of probability has always been associated with gambling and many most accessible examples still come from that activity.

Although gambling dates back thousands of years, the birth of modern probability is considered to be a 1654 letter from the Flemish aristocrat and notorious gambler Chevalier de Méré to the mathematician and philosopher Blaise Pascal. In essence the letter said:

"I used to bet even money that I would get at least one 6 in four rolls of a fair die. The probability of this is 4 times the probability of getting a 6 in a single die, i.e., $\frac{4}{6}=\frac{2}{3}$; clearly, I had an advantage and indeed I was making money. Now I bet even money that within 24 rolls of two dice I get at least one double 6. This has the same advantage ($\frac{24}{6^2}=\frac{2}{3}$), but now I am losing money. Why?"

As Pascal discussed in his correspondence with Pierre de Fermat, de Méré's reasoning was faulty; after all, if the number of rolls were 7 in the first game, the logic would give the nonsensical probability $\frac{7}{6}$.

Random variables

- Discrete random variables
- Continuous random variables

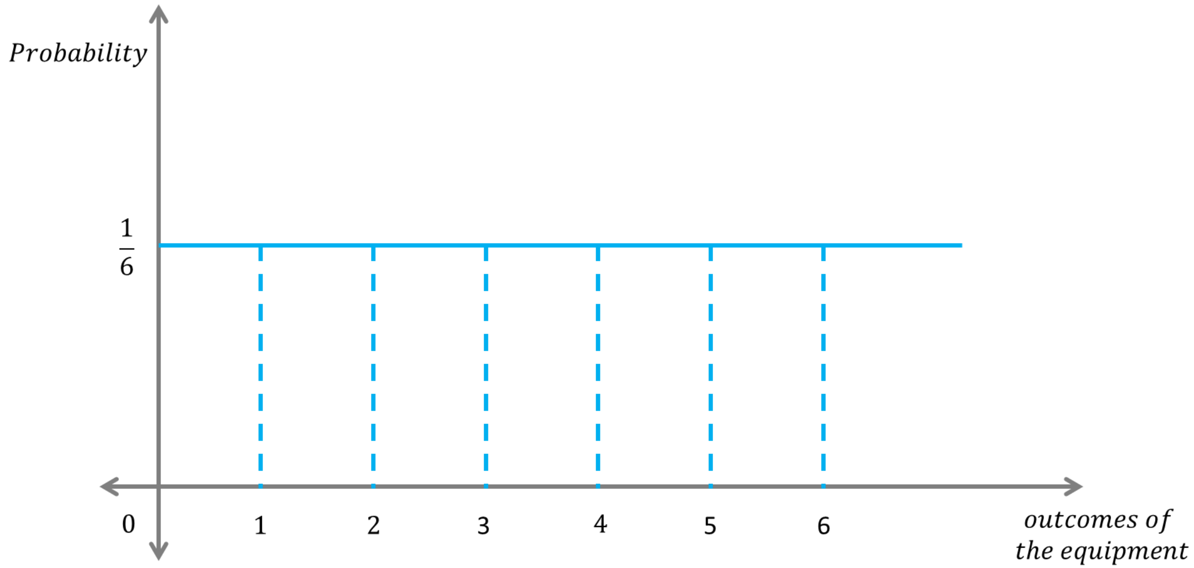
Discrete random variables

A discrete random variable takes only discrete values.

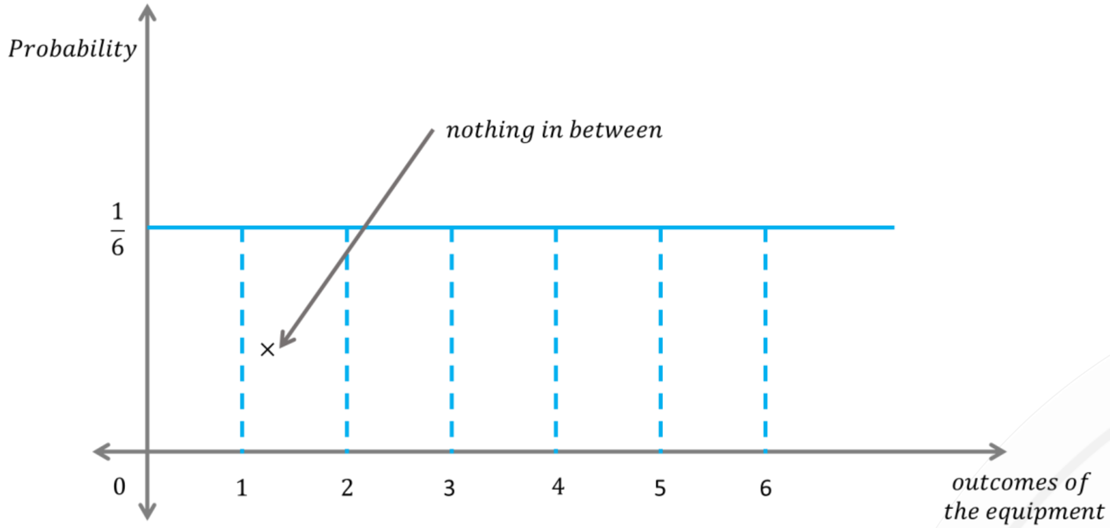
Example. Let's do the following experiment:

We throw (roll) a dice.

The results of this experiment can be 6 different outcomes, i.e., 1, 2, 3, 4, 5, and 6.



Discrete random variables



The probability of getting one value is the outcome divided by the total number of possibilities. So, probability of getting '1' is $\frac{1}{6}$.

Which is the probability to get all values???

Discrete random variables

It is 1:

$$\frac{1}{6}+\frac{1}{6}+\frac{1}{6}+\frac{1}{6}+\frac{1}{6}+\frac{1}{6}=\frac{6}{6}=1$$

So, if we roll a dice, the probability to obtain any of the six outcomes is 1. This is because we don't have any other possible outcomes.

Discrete random variables

The **mean of a discrete random variable X** is its weighted average. Each value of X is weighted by its probability. To find the mean of X, multiply each value of X by its probability, then add all the products. The mean of a random variable is called **the expected value of X**.

The expected value of *X*:

$$E[X] = \mu_X = \sum_{i=1}^{\infty} P_X(\boldsymbol{x}_i)\,\boldsymbol{x}_i$$

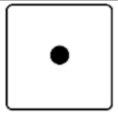
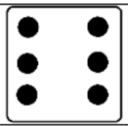
The variance of *X*:

$$\sigma^2 = Var(X) = \sum_{i=1}^{\infty} (\boldsymbol{x}_i - \mu)^2 P_X(\boldsymbol{x}_i)$$

Discrete random variables

- Example for the die roll:

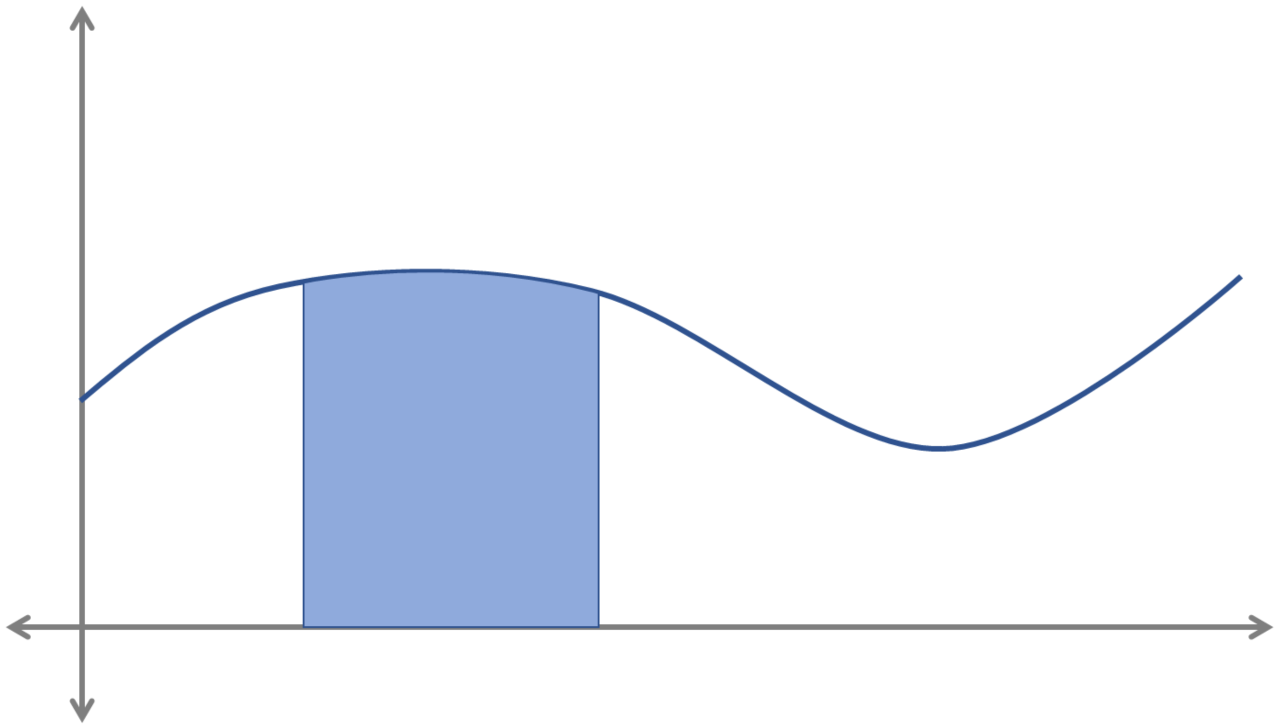
Expected Value Of A 6-Sided Die Roll

Visual (Die Face)	**Value** X	**Probability** P(X)	**Product** X*P(X)
	1	1/6	1*1/6 = 1/6
	2	1/6	2*1/6 = 2/6
	3	1/6	3*1/6 = 3/6
	4	1/6	4*1/6 = 4/6
	5	1/6	5*1/6 = 5/6
	6	1/6	6*1/6 = 6/6

Expected Value (sum of last column)
= 1/6 + 2/6 + 3/6 + 4/6 + 5/6 + 6/6
=21/6 OR 3.5

Continuous random variables

In this case we have infinity values, infinite outcomes. So, continuous probability distributions.



Continuous random variables

In this case, we cannot sum up the probabilities as we did in the case of Discrete Random Variables.

In discrete random variables: $P(X=1) = 1/6$, $P(X=1 \text{ OR } X=3)=1/6+1/6$

We cannot sum up infinite numbers. So, we have to integrate. If we integrate, then we obtain the Area.

So, in case of continuous random variables, if we want to compute the probability of the area out of all possibilities, we have to integrate the respective area.

Distributions for Continuous random variables

There are many distributions for continuous random variables. We'll study only two:

- Uniform distribution
- Gaussian distribution

Uniform Distribution

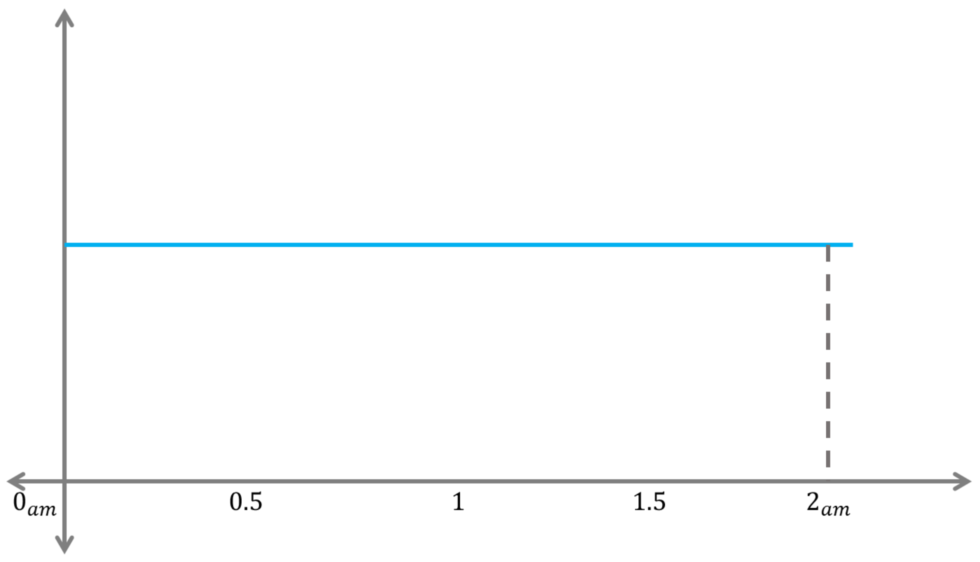
- Uniform distribution means “all the same”

Experiment: Receiving emails

Let X be a continuous random variable which is the time interval until we receive the next email, this can be (0, inf).

Uniform Distribution

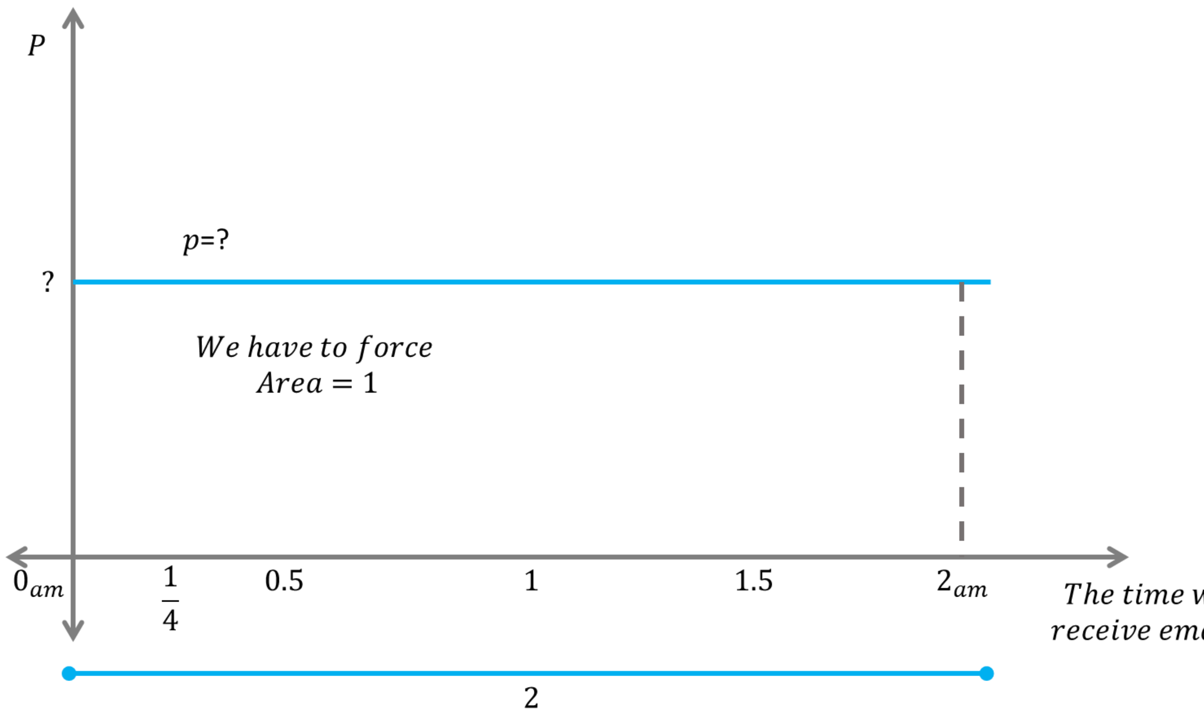
- We are interested in the probability of receiving emails between 12am and 2am (0 and 2).
- Only based on this information, we cannot say that receiving an email at 1am has bigger probability or less probability than receiving an email at other moments. So, the distribution is Uniform (the same). We did not say that at 1am is more likely to receive emails.



Uniform Distribution

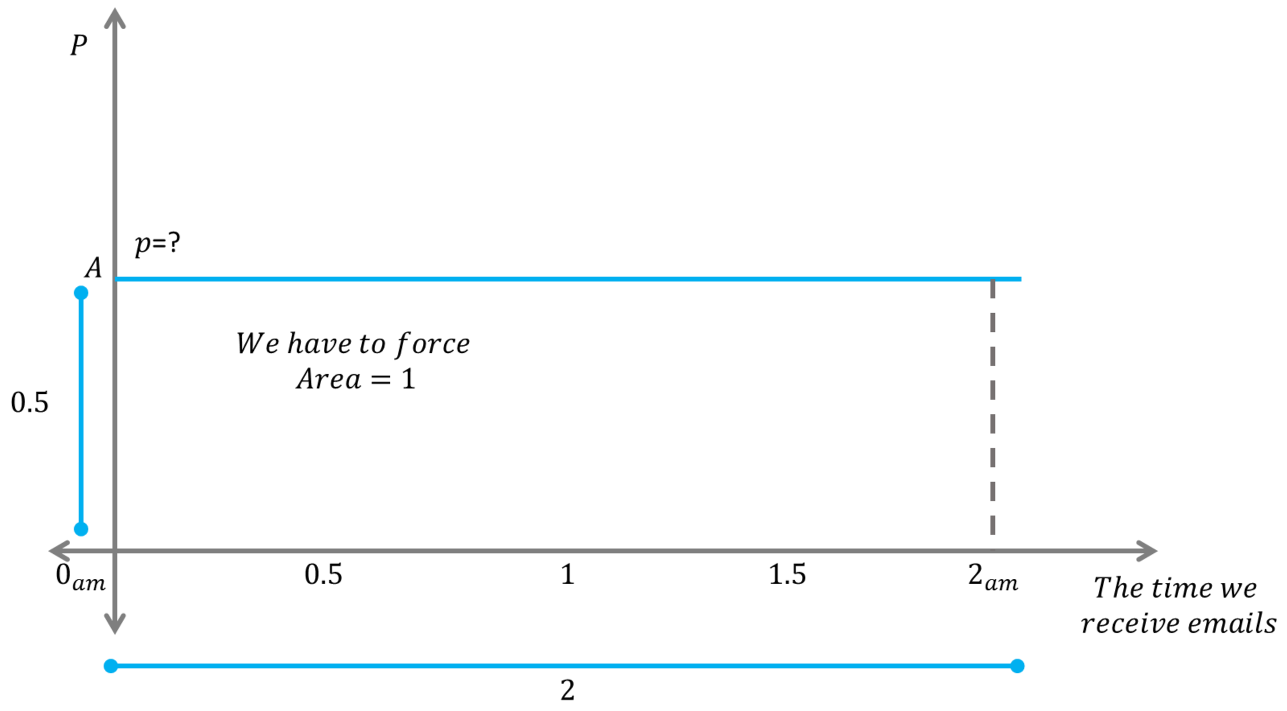
- We already agreed in case of discrete random variables that the probability must sump up at 1. Because the probability of all possible outcomes must be 1.

- However, for continuous random variables, the probability is related with area.

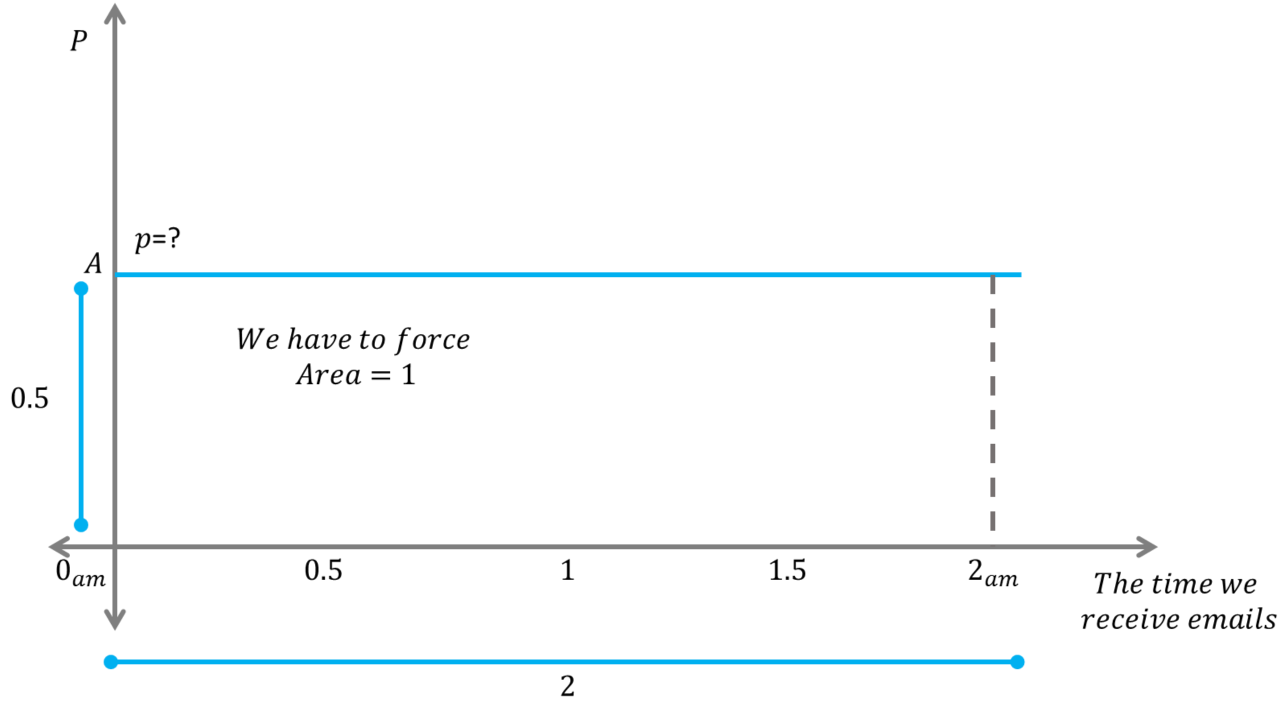


Uniform Distribution

- **Question:** Is the meaning of point A in the figure that we have a probability p=0.5?



Uniform Distribution

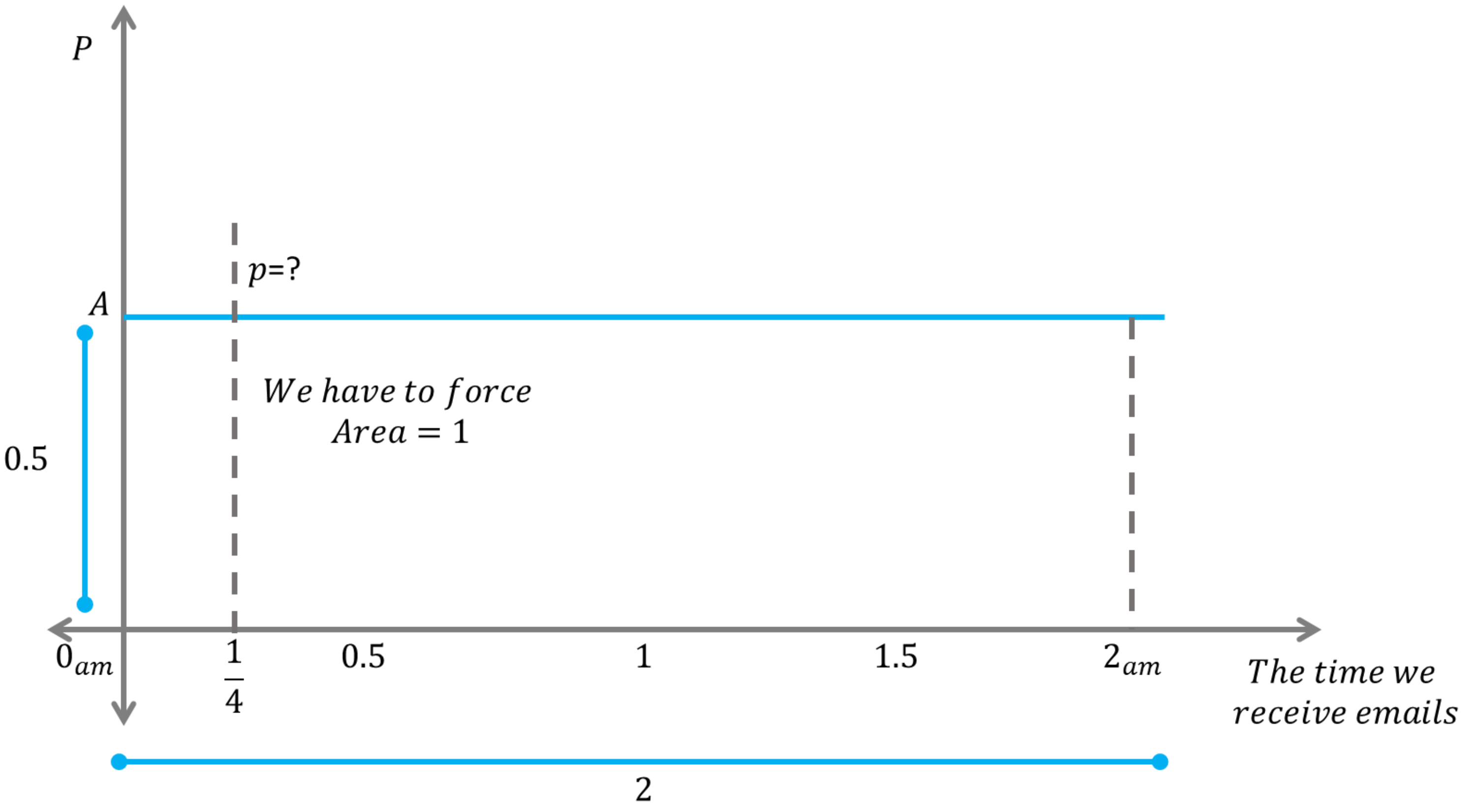


Question: Is the meaning of point A in the figure that we have a probability p=0.5?

No. This is just a value such that the area=1. The area is the probability for all possible outcomes.

Uniform Distribution

- **Question:** Which is the probability to receive an email at EXACT 12.15am?



Uniform Distribution

The probability of receiving an email at EXACT 12.15am is zero.

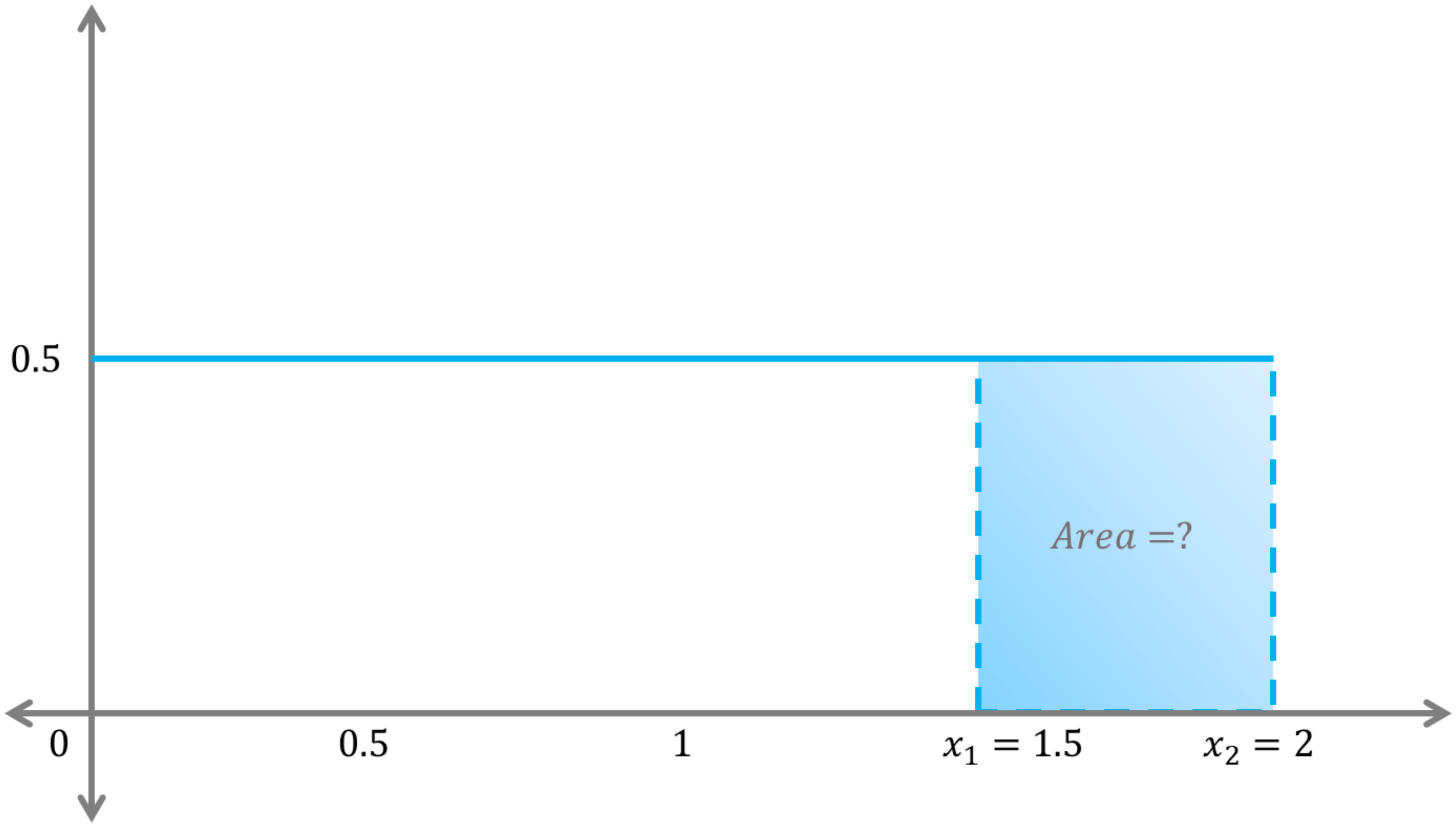
$$p(x = x_i) = 0 \qquad\qquad p\left(x = \frac{1}{4}\right) = 0$$

The probability of a continuous random variable to be equal with an outcome is zero.

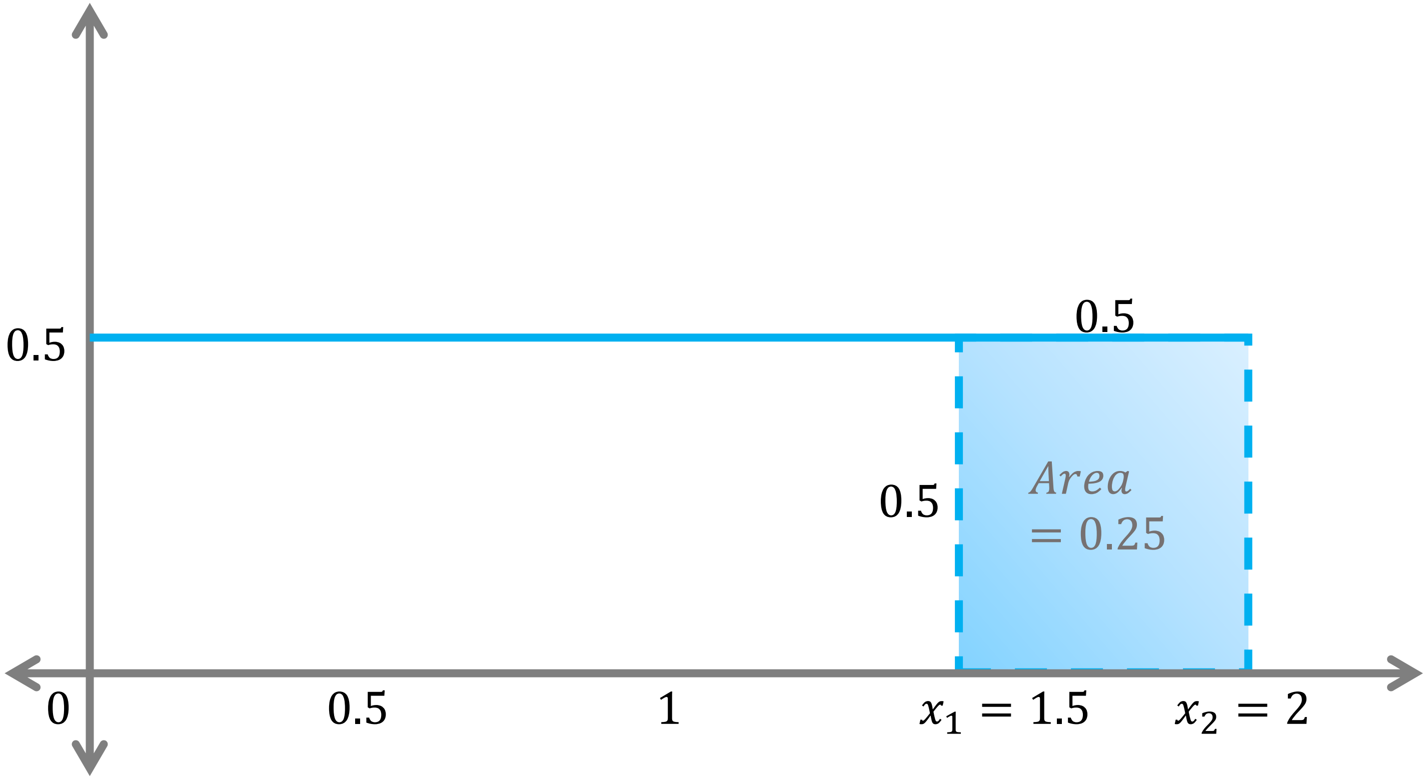
How many possible outcomes we have between 0am and 2am? We have infinite possible outcomes. So, the probability of a single outcome is $\frac{1}{\infty} = 0$.

Uniform Distribution

Which is the probability of receiving emails between 1.30am and 2am?



Uniform Distribution



$$p(x_1 < x < x_2) = 0.25 \quad (0.5 * 0.5)$$

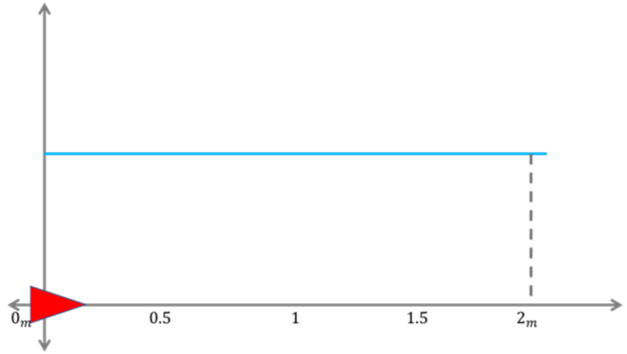
So, the probability of receiving emails between 1.30am and 2am is 0.25 (25%).

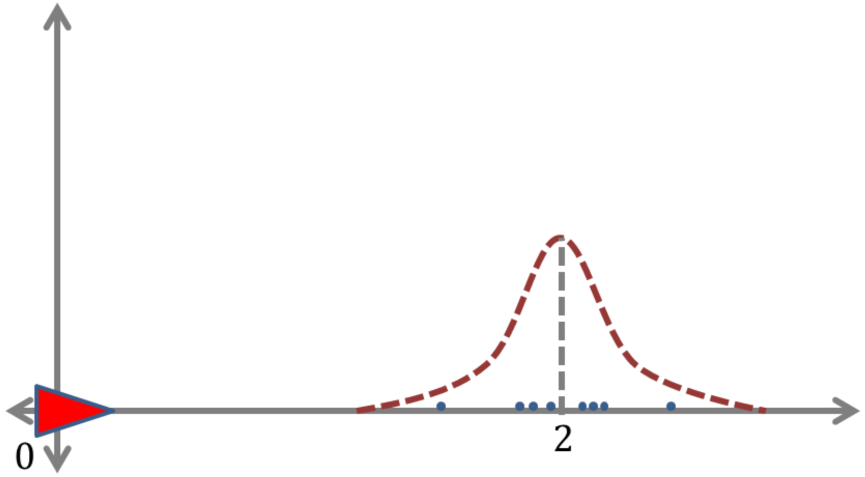
Uniform Distribution

- In robotics we don’t use uniform distributions. When we use sensors to measure a value, the measurements are distributed around the “true” value.

- For instance, we cannot say that moving a mobile robot for 2 meters we have a uniform distribution for the robot position between 0m and 2m.

Uniform Distribution vs Gaussian Distrinution

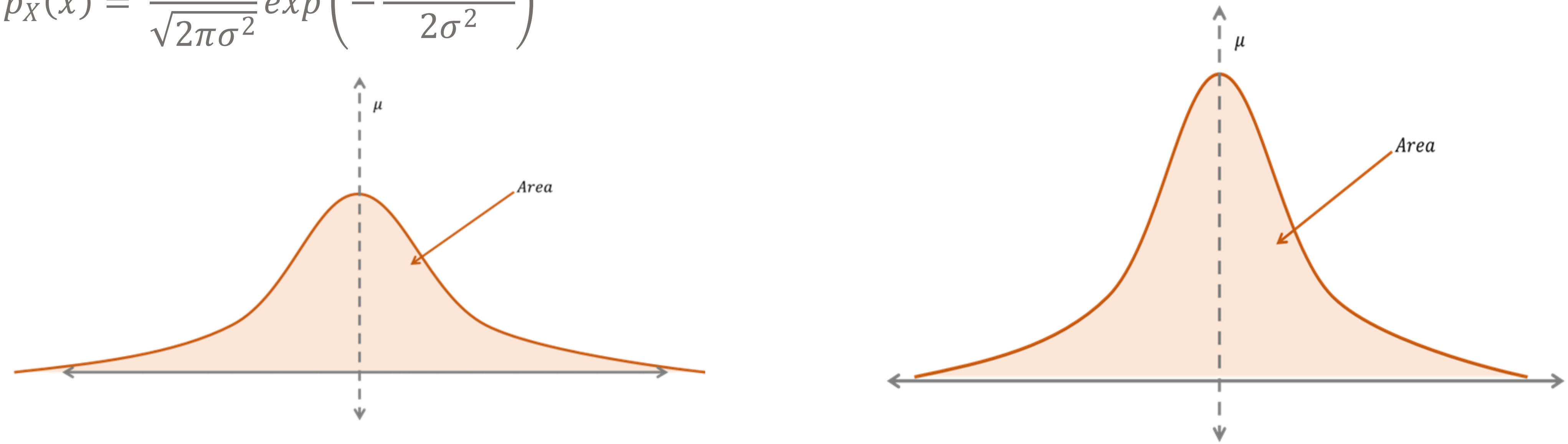




In this case we'll use a Gaussian distribution (normal distribution) to represent the uncertainty in the robot position. This distribution is a symmetric bell shape.

Gaussian Distribution

$$p_X(x) = \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

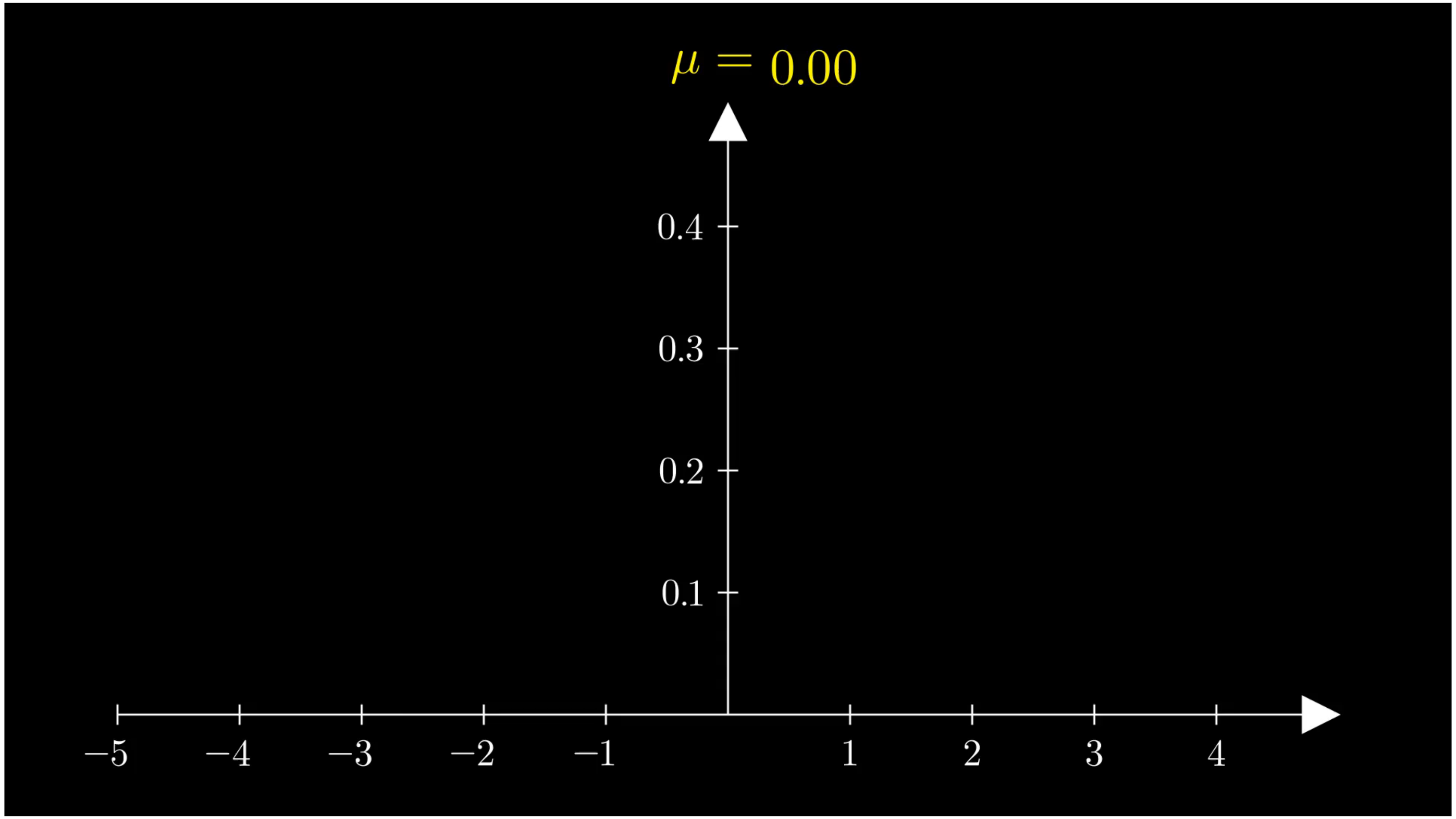


The beauty of the Gaussian Distribution is that we can describe it with just two parameters μ and σ.

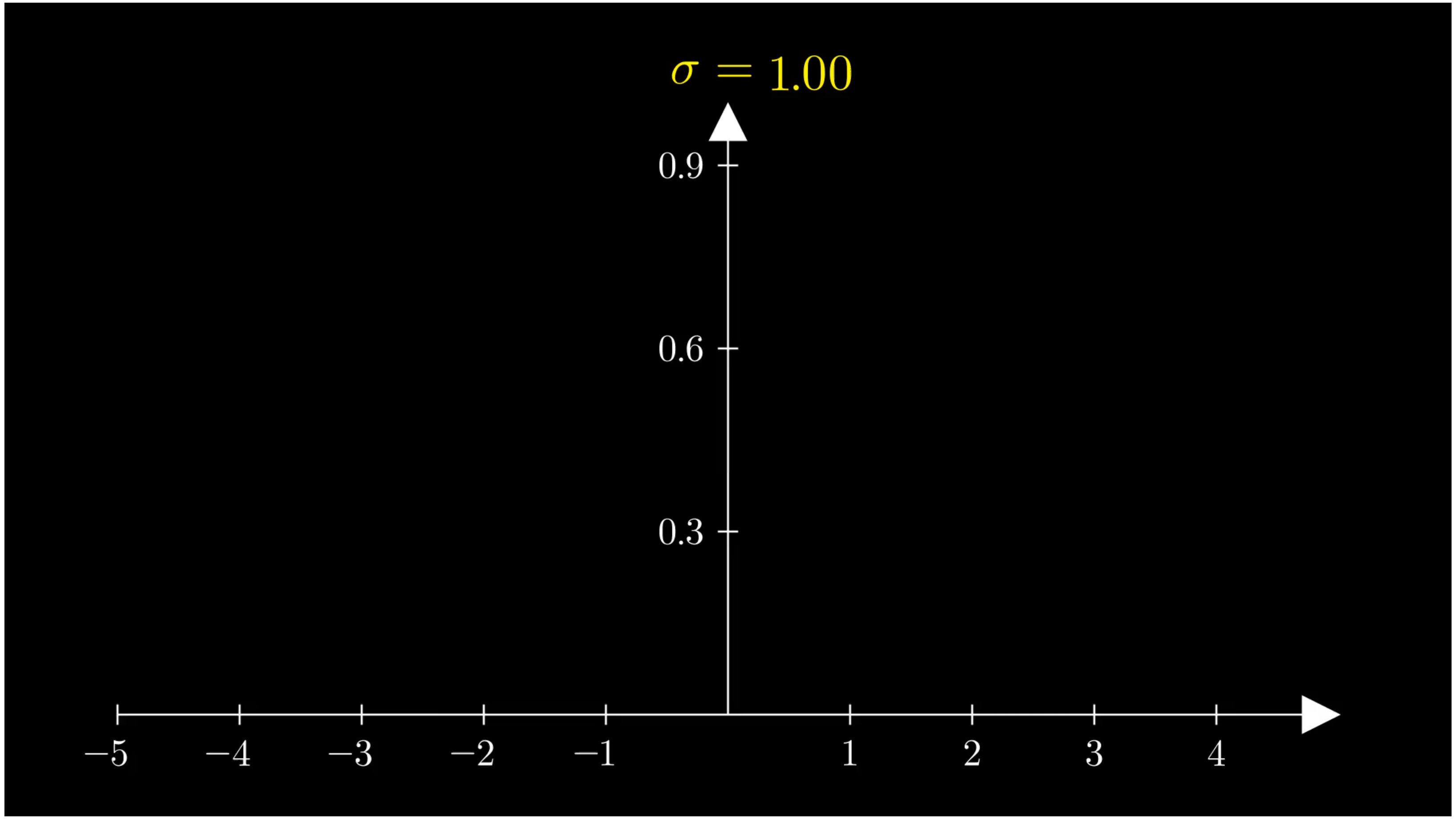
$$x \sim \mathcal{N}(\mu, \sigma)$$

$\sigma^2 variance$
σ $standard\ deviation$
μ is the mean

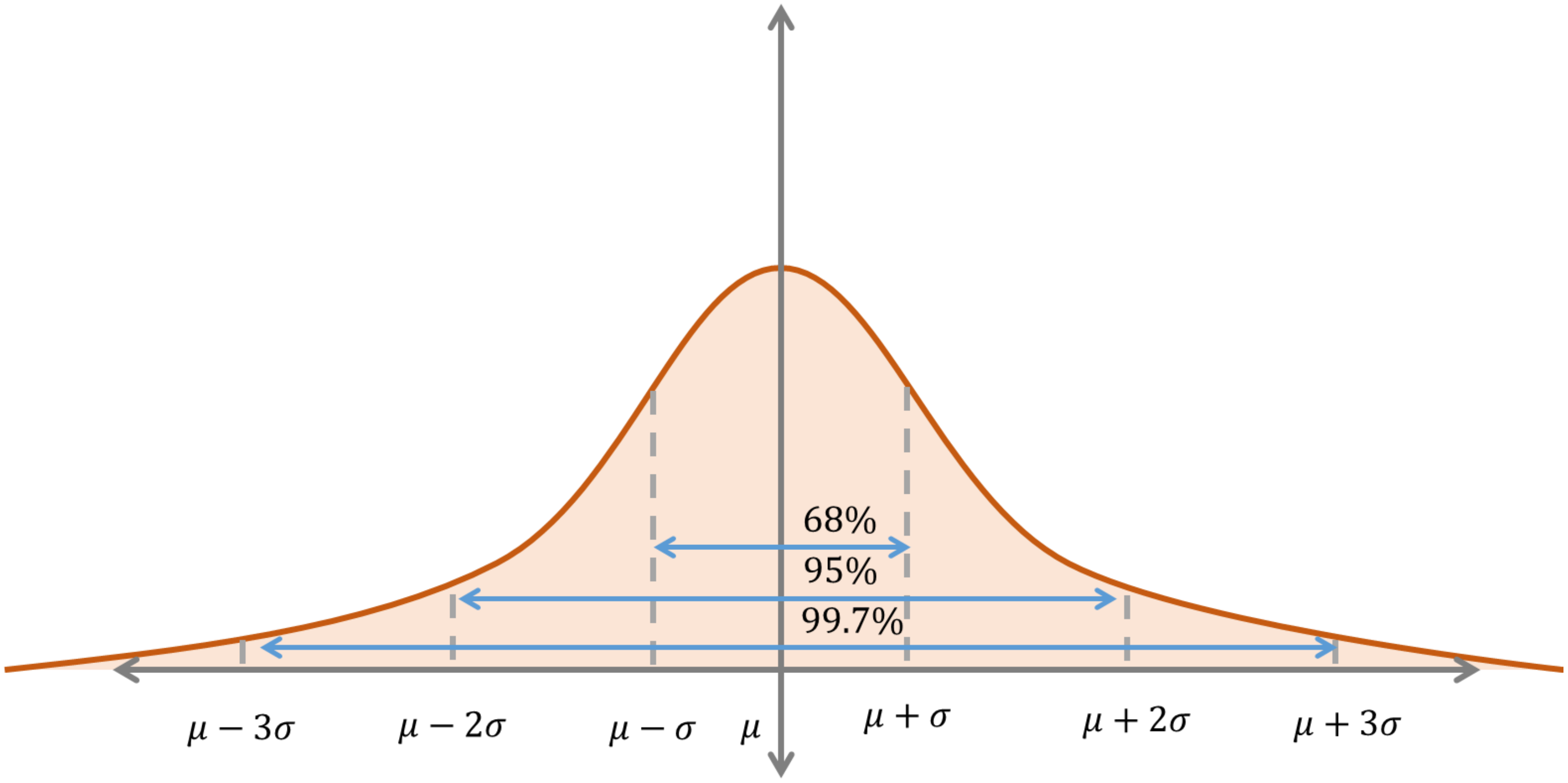
Gaussian Distribution



Gaussian Distribution



Gaussian Distribution



In robotics we use 3σ.

Motion-based Localisation (Dead Reckoning)

$$\begin{bmatrix} s_{x,k} \\ s_{y,k} \\ s_{\theta,k} \end{bmatrix} = \begin{bmatrix} s_{x,k-1} \\ s_{y,k-1} \\ s_{\theta,k-1} \end{bmatrix} + \begin{bmatrix} \Delta d \cos\left(s_{\theta,k-1}\right) \\ \Delta d \sin\left(s_{\theta,k-1}\right) \\ \Delta\theta \end{bmatrix}$$

$$\mathbf{s}_k = \mathbf{h}(\mathbf{s}_{k-1}, \mathbf{u}_k) + \mathbf{q}_k$$

So, which is the obvious way to represent the uncertainty in robot position?

Motion-based Localisation (Dead Reckoning)

$$\begin{bmatrix} s_{x,k} \\ s_{y,k} \\ s_{\theta,k} \end{bmatrix} = \begin{bmatrix} s_{x,k-1} \\ s_{y,k-1} \\ s_{\theta,k-1} \end{bmatrix} + \begin{bmatrix} \Delta d \cos\left(s_{\theta,k-1}\right) \\ \Delta d \sin\left(s_{\theta,k-1}\right) \\ \Delta\theta \end{bmatrix}$$

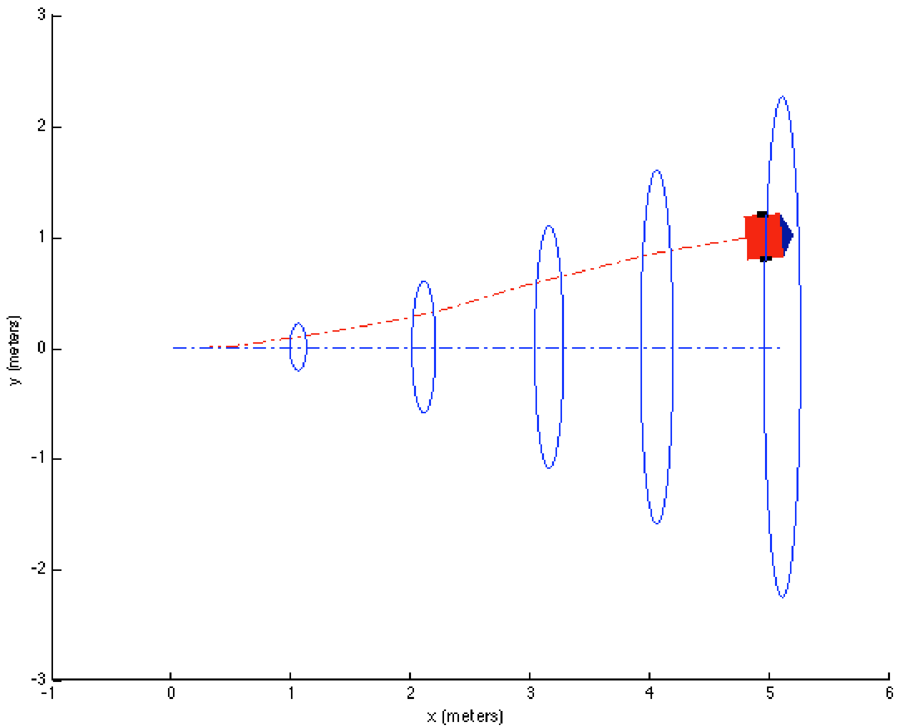
In the context of **probability**, the robot pose at time step k, denoted by $\mathbf{s}_k$, can be described as function of previous robot pose $\mathbf{s}_{k-1}$ and the current control input $\mathbf{u}_k = [v_k \quad \omega_k]^T$. This process is called the *robot motion model*.

$$\mathbf{s}_k = \mathbf{h}(\mathbf{s}_{k-1}, \mathbf{u}_k) + \mathbf{q}_k$$

where $\mathbf{q}_k$ is an additive **Gaussian noise** such that $\mathbf{q}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k)$, and $\mathbf{Q}_k$ is a positive semidefinite covariance matrix.

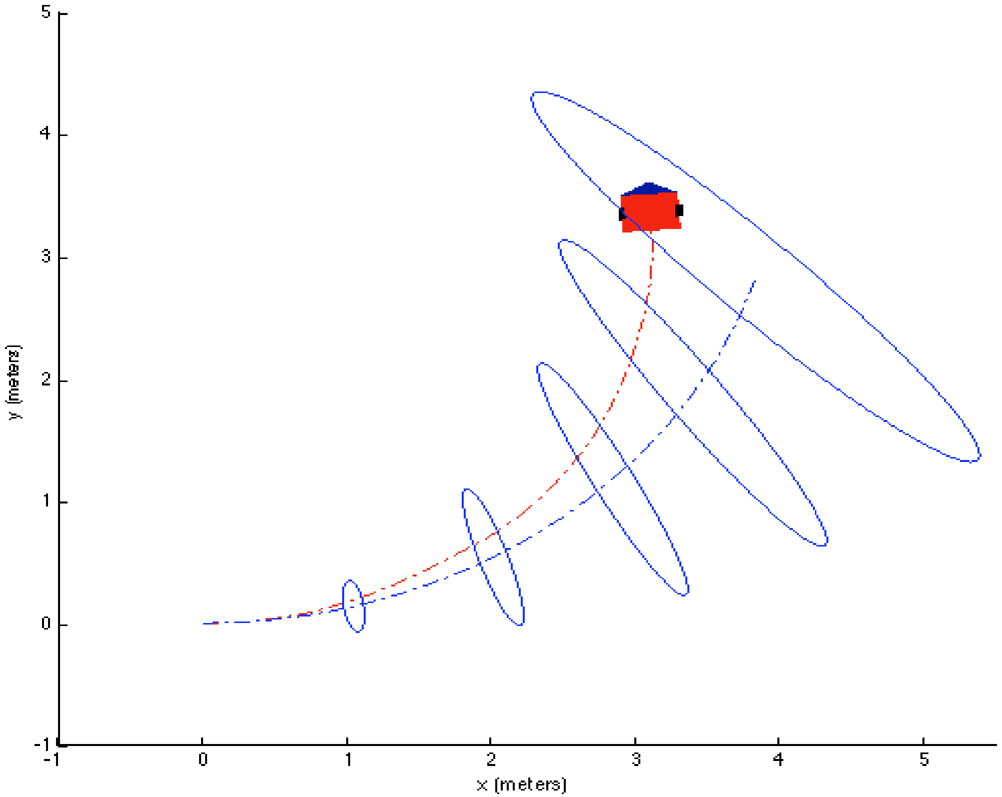
Motion-based Localisation (Dead Reckoning)

The joint uncertainty of s_x and s_y is represented by an ellipsoid around the robot. This ellipsoid is named **Ellipsoid of Confidence**. As the robot moves along the x-axis, its uncertainty along the y-axis increases faster than the x-axis due to the drift error.

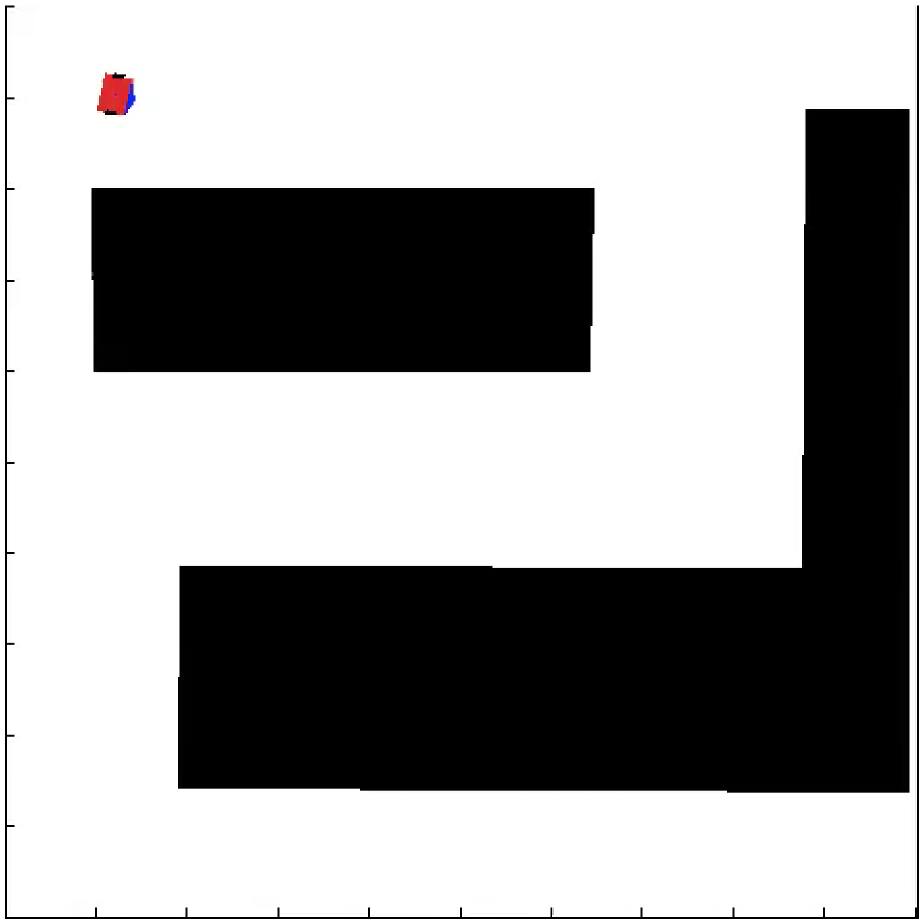


Motion-based Localisation (Dead Reckoning)

The uncertainty ellipsoid is no longer perpendicular to the motion direction as soon as the robot starts to turn.

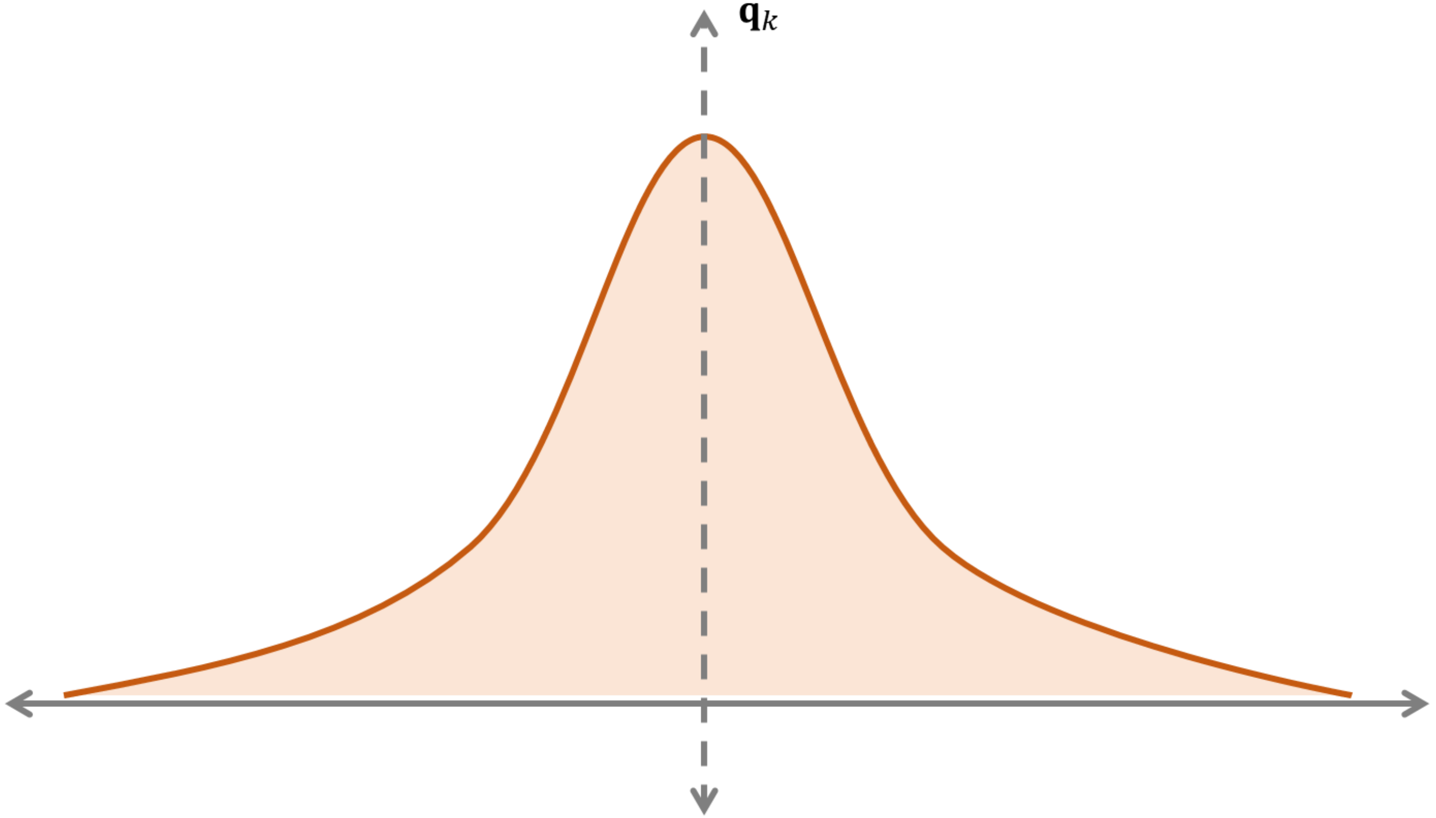


Dead Reckoning Localisation



Motion-based Localisation (Dead Reckoning)

$$\mathbf{s}_k = \mathbf{h}(\mathbf{s}_{k-1}, \mathbf{u}_k) + \mathbf{q}_k$$

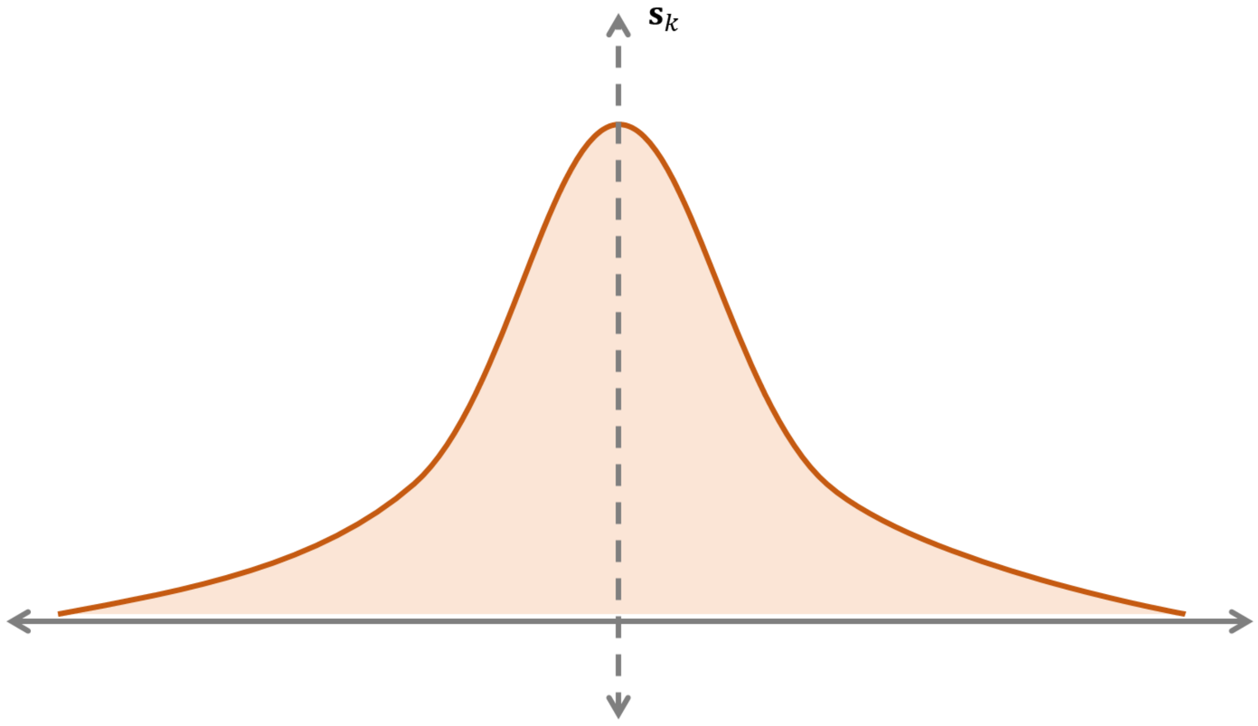


But …

Motion-based Localisation (Dead Reckoning)

$$\mathbf{s}_k = \mathbf{h}(\mathbf{s}_{k-1}, \mathbf{u}_k) + \mathbf{q}_k$$

But, we want $\mathbf{s}_k$ to be a Gaussian distribution as well.



How can we do this???

Motion-based Localisation (Dead Reckoning)

$$\mathbf{s}_1 \sim \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$$

$$\mathbf{s}_2 \sim \mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$$

$$\vdots$$

$$\mathbf{s}_{k-1} \sim \mathcal{N}(\boldsymbol{\mu}_{k-1}, \boldsymbol{\Sigma}_{k-1})$$

$$\mathbf{s}_k \sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

We said that the beauty of a Gaussian distribution is that it is a closed form, it is represented by only two parameters, $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$.

So, we just have to propagate, at every time instant, two parameters and we have the Gaussian distribution at the next time step.

Motion-based Localisation (Dead Reckoning)

We propagate the mean $\boldsymbol{\mu}$.

$$\boldsymbol{\mu}_k = \mathbf{h}(\boldsymbol{\mu}_{k-1}, \mathbf{u}_k)$$

We propagate the covariance $\boldsymbol{\Sigma}$. But we have a problem here… If we propagate $\boldsymbol{\Sigma}$ using a nonlinear function, then the new $\boldsymbol{\Sigma}$ will not represent the covariance for a Gaussian distribution anymore.

Motion-based Localisation (Dead Reckoning)

To propagate $\mathbf{\Sigma}$ we use the following result:

Affine Transformation

Consider $X \sim \mathcal{N}(\boldsymbol{\mu}_X, \mathbf{\Sigma}_X)$ in $\mathbb{R}^n$, and let $Y = \mathbf{A}X + \boldsymbol{b}$ be the affine transformation, where $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $\mathbf{b} \in \mathbb{R}^m$. Then, the random vector $Y \sim \mathcal{N}(\boldsymbol{\mu}_Y, \mathbf{\Sigma}_Y)$ such that:

$$\boldsymbol{\mu}_Y = \mathbf{A}\boldsymbol{\mu}_X + \mathbf{b}$$

$$\mathbf{\Sigma}_Y = \mathbf{A}\mathbf{\Sigma}_X\mathbf{A}^T$$

Motion-based Localisation (Dead Reckoning)

So, to propagate $\boldsymbol{\Sigma}$ we need a linear map, a linear function. However, the kinematic model of the non-holonomic robot is nonlinear.

Solution: Linearisation

Linearisation of nonlinear systems

- Linearisation around equilibrium points
- Linearisation around an operating point

Linearisation of nonlinear systems

- As we stated in the first lecture, the state-space representation of a system can be used for both, liner and nonlinear systems.

If for any $x, y, k \in \mathbb{R}$

$$f(kx) = kf(x)$$
$$f(x+y) = f(x) + f(y)$$

then the function f is linear

A function which does not satisfy these conditions is a nonlinear function.

Linearisation around equilibrium points

Let us consider the dynamical system (autonomous):

$$\dot{x}(t) = f\big(x(t)\big), \qquad x(0) = x_0$$

$$f\colon \mathbb{R}^n \to \mathbb{R}^n$$

Definition 1 (equilibrium point): The point $x_e \in \mathbb{R}^n$ is an equilibrium point of the system if $f(x_e) = 0$, i.e.,

$$\dot{x}(t)\Big|_{x=x_e} = f(x_e) = 0$$

Linearisation around equilibrium points

Definition 2 (Stability of equilibrium points): An equilibrium point x_e is said to be stable if for any initial condition around the equilibrium point, the distance from the solution at any instant and the equilibrium point is bounded. Mathematically, for any $\epsilon > 0$ there exist δ such that:

$$\|x_0 - x_e\| < \delta \quad then \quad \|x(t) - x_e\| < \epsilon \quad \forall t \geq 0$$

Moreover, is said to be asymptotically stable if $\lim_{t\to\infty} x(t) = 0$.

If the equilibrium point x_e is not stable, it is said to be unstable.

Linearisation around equilibrium points

An equilibrium point is stable if the trajectory of the state tries to recover or stay close to the equilibrium point when the system slightly disturbed. In other words, it is able to keep this position even though small disturbances will disturb the system. When the equilibrium point is unstable, any small perturbation will lead to the loss of equilibrium and the equilibrium position will not be recovered.

In the linear case, we are only interested in the trivial equilibrium point: $x_e = 0$.

Linearisation around equilibrium points

A nonlinear system behaves approximately as a linear system near an equilibrium point (of course if the function f is continuous). This can be proved using Taylor expansion around the equilibrium point:

$$f(x) \cong f(x_e) + \mathcal{J}_f(x_e)(x - x_e)$$

where $\mathcal{J}_f$ is the Jacobian matrix. It is clear that if $f(x_e) = 0$ and we define a new state as $\Delta x = x - x_e$, then $f(x)$ becomes the matrix $\mathcal{J}_f$ times the vector Δx, if $x - x_e$ is close to zero, hence the nonlinear system becomes linear.

$$\frac{d}{dt}(\Delta x) \cong A\Delta x$$

where $A = \mathcal{J}_f(x_e)$ and if $\Delta x = x - x_e$, then $\frac{d}{dt}(\Delta x) = \dot{x}$.

Linearisation around equilibrium points

Definition 3 (Jacobian): Given a vectorial function $f: \mathbb{R}^n \rightarrow \mathbb{R}^n$, then the Jacobian matrix $\mathcal{J}_f \in \mathbb{R}^{n \times n}$ is defined by:

$$\mathcal{J}_f = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}$$

Linearisation around equilibrium points

So, we can analyse the equilibrium point by performing a linearisation and studying the properties of the Jacobian matrix at the equilibrium point.

The stability of the equilibrium point x_e of the system $\dot{x}(t) = f\big(x(t)\big),\ x(0) = x_0$, is equivalent to the stability of the linear system defined by $\dot{x} = Ax$, where $A = \mathcal{J}_f(x_e)$.

Linearisation around equilibrium points

Example: Pendulum system

$$\ddot{\theta} + \frac{g}{l}\sin(\theta) = 0$$

Let us define the states:

$$x_1 = \theta$$

$$x_2 = \dot{\theta}$$

So, the state-space representation of the pendulum system is:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -\frac{g}{l} sin(x_1) \end{cases}$$

Linearisation around equilibrium points

Hence, the vectorial function $f(x_1, x_2)$ is given by:

$$f(x_1, x_2) = \begin{bmatrix} x_2 \\ -\frac{g}{l}\sin(x_1) \end{bmatrix}$$

At the equilibrium points satisfies $f(x_1, x_2) = 0$, i.e.,

$$x_2 = 0$$
$$\sin(x_1) = 0$$

Linearisation around equilibrium points

For $\theta \in (-\pi, \pi]$, the equilibrium points of the pendulum system are $x_{e1} = (0,0)$ and $x_{e2} = (\pi, 0)$.

The Jacobian matrix is:

$$\mathcal{J}_f(x_1, x_2) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix}$$

$$\mathcal{J}_f = \begin{bmatrix} \frac{\partial}{\partial x_1} x_2 & \frac{\partial f_1}{\partial x_2} x_2 \\ \frac{\partial}{\partial x_1}\left(-\frac{g}{l}\sin(x_1)\right) & \frac{\partial}{\partial x_2}\left(-\frac{g}{l}\sin(x_1)\right) \end{bmatrix}$$

$$\mathcal{J}_f = \begin{bmatrix} 0 & 1 \\ -\frac{g}{l} cos(x_1) & 0 \end{bmatrix}$$

Linearisation around equilibrium points

At equilibrium point $x_{e1} = (0,0)$, the pendulum system can be linearised as:

$$\dot{x} = \mathcal{J}_f(0,0)x = \begin{bmatrix} 0 & 1 \\ -\frac{g}{l} & 0 \end{bmatrix} x$$

Question: How can we verify mathematically if the equilibrium point is stable???

Linearisation around equilibrium points

Answer: By checking the eigenvalues of $\mathcal{J}_f(0,0)$, i.e., the eigenvalues of the A matrix of the linearised pendulum system.

The eigenvalues of $\mathcal{J}_f(0,0)$ are given by:

$$\det\left(\mathcal{J}_f(0,0) - \lambda I\right) = \begin{bmatrix} -\lambda & 1 \\ -\frac{g}{l} & -\lambda \end{bmatrix} = \lambda^2 + \frac{g}{l} = 0$$

$$\Downarrow$$

$$\lambda = \pm\sqrt{\frac{g}{l}}j$$

So??? How is the equilibrium point???

Linearisation around equilibrium points

The real part of the eigenvalues is xero and they are on the imaginary axis, then the equilibrium point x_{e1} is marginally stable.

At equilibrium point $x_{e2} = (\pi, 0)$ we need to use a transformation of the space since the equilibrium point is not the origin, hence $\Delta x = x - x_2$. Thus the pendulum systems can be linearised around x_{e2} as:

$$\Delta\,\dot{x} = \mathcal{J}_f(\pi, 0)\Delta x = \begin{bmatrix} 0 & 1 \\ \frac{g}{l} & 0 \end{bmatrix} \Delta x$$

Linearisation around equilibrium points

The eigenvalues of $\mathcal{J}_f(\pi, 0)$ are given by:

$$\det\left(\mathcal{J}_f(\pi,0) - \lambda I\right) = \begin{bmatrix} -\lambda & 1 \\ \frac{g}{l} & -\lambda \end{bmatrix} = \lambda^2 - \frac{g}{l} = 0$$

$$\Downarrow$$

$$\lambda = \pm\sqrt{\frac{g}{l}}$$

So??? How is the equilibrium point???

Linearisation around equilibrium points

The eigenvalues of $\mathcal{J}_f(\pi, 0)$ are given by:

$$\det\left(\mathcal{J}_f(\pi,0) - \lambda I\right) = \begin{bmatrix} -\lambda & 1 \\ \frac{g}{l} & -\lambda \end{bmatrix} = \lambda^2 - \frac{g}{l} = 0$$

$$\Downarrow$$

$$\lambda = \pm\sqrt{\frac{g}{l}}$$

The real part of one eigenvalue is positive, therefore the equilibrium point x_{e2} is unstable.

Linearisation around equilibrium points

Animation for pendulum system:

https://www.youtube.com/watch?v=ovDWmGhMKyI&list=PLqCuMQTwnIP99CrzdPEroGhdAhzVfvWgR&index=11

Linearisation around an operating point

When the nonlinear system is required to be operating with an input that is different to zero, then we can no longer refer to as an equilibrium point of the system since this concept is related with the autonomous systems. In this case, we use the concept of operating point.

Linearisation around an operating point

Let us consider the nonlinear system given by:

$$\dot{x} = f(x, u)$$

$$y = h(x, u)$$

Then if $f(x_0, u_0) = 0$, the point (x_0, u_0) is referred to as an operating point. Under this condition, we can perform a linearisation of the system. Let us define the new input, state, and output as the variation around x_0, u_0, and y_0.

$$\Delta u = u - u_0$$

$$\Delta x = x - x_0$$

$$\Delta y = y - y_0$$

Linearisation around an operating point

We apply a Taylor expansion of $f(x,u)$ and $h(x,u)$ around the point (x_0,u_0) and we get:

$$f(x,u) \simeq f(x_0,u_0) + \mathcal{J}_f^x(x_0,u_0)(x - x_0) + \mathcal{J}_f^u(x_0,u_0)(u - u_0)$$

$$h(x,u) \simeq h(x_0,u_0) + \mathcal{J}_h^x(x_0,u_0)(x - x_0) + \mathcal{J}_h^u(x_0,u_0)(u - u_0)$$

where $f(x_0,u_0) = 0$ and $h(x_0,u_0) = y_0$.

The linearised system is given by:

$$\frac{d}{dt}(\Delta x) \simeq \mathcal{J}_f^x(x_0,u_0)\Delta x + \mathcal{J}_f^u(x_0,u_0)\Delta u$$

$$\Delta y \simeq \mathcal{J}_h^x(x_0,u_0)\Delta x + \mathcal{J}_h^u(x_0,u_0)\Delta u$$

where the superscripts x and u in the Jacobian matrices indicate the parameter that is considered as a variable.

Next lecture

- Mini challenge 2
- Ellipsoid of confidence