# Sensors and Sensing
# Robot Kinematics and Odometry

Todor Stoyanov

Centre for Applied Autonomous Sensor Systems
Örebro University
Sweden

ÖREBRO UNIVERSITET

# Outline

# Outline

# Disclaimer

This part of the lecture is based on slides by Dimitar Dimitrov and Robert Krug.
The original and more detailed overview can be found at:

Multi-body simulation, Dimitar Dimitrov: `http://www.aass.oru.se/Research/mro/drdv_dir/course_mbs_2011.html`

Robot control, Robert Krug: `http://www.aass.oru.se/Research/mro/rkg_dir/course_rc_2016.html`

# Preliminaries

Linear Algebra studies vector spaces and linear mappings between these spaces

## Basic Notation

| | |
|---|---|
| $:$ | such that |
| $\exists$ | there exists |
| $\forall$ | for all |
| $\mathcal{A} = \{a, b, c\}$ | $\mathcal{A}$ is a set with elements a, b and c |
| $a \in \mathcal{A}$ | a is an element of the set $\mathcal{A}$ |
| $\mathbb{R}$ | set of real numbers $(1, \pi, e, \sqrt{2}, -3, ...)$ |
| $\mathbb{C}$ | set of complex numbers |
| $z = x + iy$ | a complex number $(z \in \mathbb{C})$ |

# Preliminaries

| | |
|---|---|
| $\mathbb{R}^{m \times n}$ | set of $m \times n$ matrices of real numbers |
| $\mathbb{R}^{n}$ | set of $n \times 1$ matrices (n-vectors) of real numbers |
| $A^T$ | the transpose of the matrix A |
| $f : \mathcal{X} \to \mathcal{Y}$ | function f maps the domain $\mathcal{X}$ into the co-domain $\mathcal{Y}$. For example $f : \mathbb{R}^n \to \mathbb{R}$. |
| $\mathcal{R}(A)$ | range space of the matrix A |
| $\mathcal{N}(A)$ | nullspace of the matrix A |
| $\lambda_i(A)$ | $i^{th}$ eigenvalue of A |

# Preliminaries

We will make frequent use of the quantifiers $\exists$ ("there exsists") and $\forall$ ("for all"). Statements containing them must be parsed carefully.

For example let $\mathcal{C}$ be the set of all cats, and let $\mathcal{D}$ be the set of all dogs. The following statements mean entirely different things:

$$\exists c \in \mathcal{C} \quad : \forall d \in \mathcal{D} \ c \text{ runs faster than } d$$
$$\forall c \in \mathcal{C} \ \ \exists d \in \mathcal{D} : \ c \text{ runs faster than } d$$

# Vector Spaces

A real vector space is a collection of objects with nice properties regarding adding and scaling them. It consists of:

(i) a set $\mathcal{V}$;

(ii) a vector sum $+ : \mathcal{V} \times \mathcal{V} \to \mathcal{V}$;

(iii) a scalar multiplication $\cdot : \mathbb{R} \times \mathcal{V} \to \mathcal{V}$,

which satisfy the following properties:

# Vector Spaces

1. $x + y = y + x, \quad \forall x, y \in \mathcal{V}$ (commutativity of addition)
2. $x + (y + z) = (x + y) + z, \quad \forall x, y, z \in \mathcal{V}$ (associativity of addition)
3. $\exists v_0 \in \mathcal{V} : x + v_0 = v_0 + x = x, \quad \forall x \in \mathcal{V}$ (existence of additive identity)
4. $\exists (-x) \in \mathcal{V} : x + (-x) = v_0, \quad \forall x \in \mathcal{V}$ (existence of additive inverse)
5. $(\alpha\beta)x = \alpha(\beta x), \quad \forall \alpha, \beta \in \mathbb{R}, \quad \forall x \in \mathcal{V}$ (associativity of scalar multiplication)
6. $\alpha(x + y) = \alpha x + \alpha y, \quad \forall \alpha \in \mathbb{R}, \quad \forall x, y \in \mathcal{V}$ (distributive rule)
7. $(\alpha + \beta)x = \alpha x + \beta x, \quad \forall \alpha, \beta \in \mathbb{R}, \quad \forall x \in \mathcal{V}$ (distributive rule)
8. $1x = x, \quad \forall x \in \mathcal{V}$ (multiplicative identity)

# Vectors

We denote vectors using lowercase characters. For example

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

is a 3D-vector, or simply a 3-vector.

- we will often write $x \in \mathbb{R}^3$ to specify the dimension of a vector
- $x_1$, $x_2$ and $x_3$ are the (scalar) components of $x$
- in general $x_i$ is the i-th component of $x$
- the expression $x = 0$ implies that the vectors $x$ and $0$ have the same dimensions and $x_i = 0$ for all i.
- $x$ is a column vector, $x^T = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}$ is a row vector

# Vector operations

## Vector addition and subtraction

Let $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^n$, then

$$x + y = \begin{bmatrix} x_1 + y_1 \\ x_2 + y_2 \\ \vdots \\ x_n + y_n \end{bmatrix}, \quad x - y = \begin{bmatrix} x_1 - y_1 \\ x_2 - y_2 \\ \vdots \\ x_n - y_n \end{bmatrix}$$

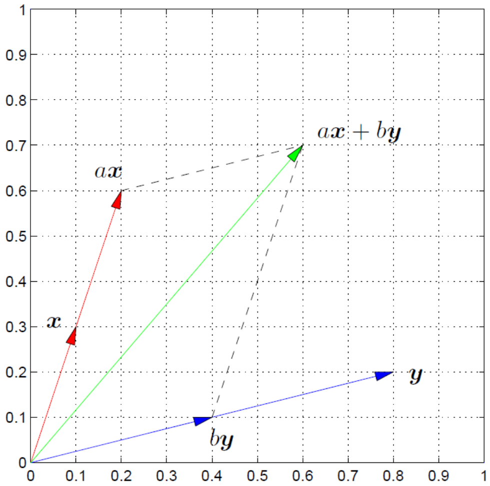
## Scalar-Vector multiplication

Let $a \in \mathbb{R}$ and $x \in \mathbb{R}^n$, then

$$ax = \begin{bmatrix} ax_1 \\ ax_2 \\ \vdots \\ ax_n \end{bmatrix}$$

# Example (vectors in the plane)

$x = (0.1, 0.3)$, $y = (0.8, 0.2)$, $a = 2$, $b = 0.5$, $ax + by = (0.6, 0.7)$.



# Inner products

The inner product (or dot product, or scalar product) of two n-vectors $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_n)$ is defined as

$$x_1 y_1 + x_2 y_2 + \cdots + x_n y_n$$

Hence, the inner product of x and y is given by $x^T y$

$$\begin{bmatrix} x_1 & x_2 & \ldots & x_n \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = x_1 y_1 + x_2 y_2 + \cdots + x_n y_n$$

# Inner products

The inner product (or dot product, or scalar product) of two n-vectors $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_n)$ is defined as

$$x_1 y_1 + x_2 y_2 + \cdots + x_n y_n$$

**Properties ($x, y, z \in \mathbb{R}^n$, $a \in \mathbb{R}$)**

- $x^T y = y^T x$
- $a(x^T y) = (ax)^T y$
- $z^T (x + y) = z^T x + z^T y$

# Euclidean norm

The Euclidean norm (or $\ell_2$ norm) of n-vector x is defined as

$$\|x\| = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2} = \sqrt{x^T x}$$

## Properties

A function $\|\cdot\| : \mathbb{R}^n \to \mathbb{R}$ is called a norm if the following conditions are satisfied:

- $\|x\| \geq 0$ for all $x \in \mathbb{R}^n$,
- $\|x\| = 0$ if and only if $x = 0$,
- $\|ax\| = |a|\|x\|$, for all $x \in \mathbb{R}^n, a \in \mathbb{R}$ (homogeneity),
- $\|x + y\| \leq \|x\| + \|y\|$ for all $x, y \in \mathbb{R}^n$ (triangle inequality)

A norm is a measure of the length of a vector.

# Angle between vectors

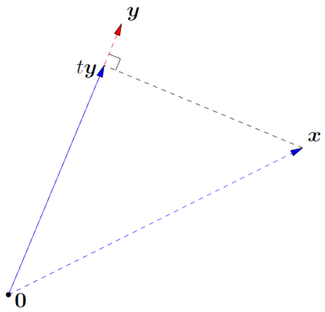
We define the unsigned angle $\theta$ ($\theta \in [0, \pi]$) between two n-vectors x and y as

$$\theta = \arccos\left(\frac{x^T y}{\|x\| \|y\|}\right)$$

- when x and y are aligned (i.e., $\theta = 0$), $x^T y = \|x\| \|y\|$
- when x and y are opposite (i.e., $\theta = \pi$), $x^T y = -\|x\| \|y\|$
- when x and y are orthogonal (i.e., $\theta = \frac{\pi}{2}$), $x^T y = 0$

$$\text{hence} \quad -1 \leq \frac{x^T y}{\|x\| \|y\|} \leq 1$$

# Projection of a vector on a line (through the origin)




ty is the projection of x on the line passing through the origin and y. If $\|y\| = 1$, then $t = x^T y$ (why?)

The inner product can be use to measure how parallel two vectors are

# Cross product of 3D-vectors

We denote the cross product of two vectors $x = (x_1, x_2, x_3)$, $y = (y_1, y_2, y_3)$ using $x \times y$, and define it as

$$x \times y = \begin{bmatrix} x_2 y_3 - x_3 y_2 \\ x_3 y_1 - x_1 y_3 \\ x_1 y_2 - x_2 y_1 \end{bmatrix}$$

## Properties

- $x \times (ax) = a(x \times x) = a0 = 0$
- Let $x \neq y$ and $z = x \times y$, then $z \perp x$ and $z \perp y$
- $x \times y = -y \times x$
- $u \times (x + y) = ux + uy$

# Linear independence & bases

Two n-vectors x and y are linearly independent if

$$ax + by = 0$$

only for $a = b = 0$. i.,e., x is not a scalar multiple of y.

The linear span of two n-vectors x and y is the set of all vectors

$$ax + by, \quad \text{for all} \quad a, b \in \mathbb{R}$$

Vectors $x_i \in \mathbb{R}^n$, $i = 1, \ldots, n$ are said to form a basis for $\mathbb{R}^n$ if and only if

- the vectors $x_i$ span $\mathbb{R}^n$
- the vectors $x_i$ are linearly independent

# Matrices

$m \times n$-matrix A has the following form

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$$

- $a_{ij}$ are (scalar) coefficients
- $m$ is called row dimension
- $n$ is called column dimension
- the expression $A = 0$ implies that the matrices $A$ and $0$ have the same dimensions and $a_{ij} = 0$ for all $i, j$.

We specify the dimension of a matrix $A$ using $A \in \mathbb{R}^{m \times n}$.

# Matrix notation

Matrices will be denoted (mostly) by uppercase bold characters (for example $\mathbf{A}$).

- $\mathbf{A}_{ij}$ will denote the $(i,j)^{th}$ entry of $\mathbf{A}$
- $\mathbf{A}_i$ will denote the $i^{th}$ column of $\mathbf{A}$
- $\mathbf{a}_i^T$ will denote the $i^{th}$ row of $\mathbf{A}$

$$\mathbf{A} = \begin{bmatrix} | & | & & | \\ \mathbf{A}_1 & \mathbf{A}_2 & \dots & \mathbf{A}_n \\ | & | & & | \end{bmatrix} = \begin{bmatrix} — & \mathbf{a}_1^T & — \\ — & \mathbf{a}_2^T & — \\ & \vdots & \\ — & \mathbf{a}_m^T & — \end{bmatrix}$$

# Matrix transpose

The transpose of an $m \times n$-matrix

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$$

is given by

$$A^T = \begin{bmatrix} a_{11} & a_{21} & \dots & a_{m1} \\ a_{12} & a_{22} & \dots & a_{m2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1n} & a_{2n} & \dots & a_{mn} \end{bmatrix}$$

Note that the transpose of an $m \times n$-matrix is an $n \times m$-matrix

# Scalar multiplication and addition

## Addition of two $m \times n$-matrices A and B

$$A + B = \begin{bmatrix} a_{11} + b_{11} & a_{12} + b_{12} & \dots & a_{1n} + b_{1n} \\ a_{21} + b_{21} & a_{22} + b_{22} & \dots & a_{2n} + b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} + b_{m1} & a_{m2} + b_{m2} & \dots & a_{mn} + b_{mn} \end{bmatrix}$$

## Scalar-Matrix multiplication

$$bA = \begin{bmatrix} ba_{11} & ba_{12} & \dots & ba_{1n} \\ ba_{21} & ba_{22} & \dots & ba_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ ba_{m1} & ba_{m2} & \dots & ba_{mn} \end{bmatrix}$$

where b is a scalar

# Matrix-Vector product

Product of an $m \times n$-matrix $A$ and a $n$-vector $x$

$$Ax = \begin{bmatrix} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \end{bmatrix}$$

The column dimension of $A$ and the dimension of $x$ should be compatible (i.e., the same)

The matrix-vector product $Ax$ can be expressed as

$$Ax = A_1x_1 + A_2x_2 + \cdots + A_nx_n$$

where $A_i$ is the i–th column of $A$

# Example (matrix-vector product)

The cross product of two vectors $x = (x_1, x_2, x_3)$, $y = (y_1, y_2, y_3)$ can be formulated as a matrix-vector multiplication

$$x \times y = \begin{bmatrix} x_2 y_3 - x_3 y_2 \\ x_3 y_1 - x_1 y_3 \\ x_1 y_2 - x_2 y_1 \end{bmatrix} = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

$$= - \begin{bmatrix} 0 & -y_3 & y_2 \\ y_3 & 0 & -y_1 \\ -y_2 & y_1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

# Example (matrix-vector product)

A matrix that satisfies

$$\begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix}^T = -\begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix}$$

is called skew-symmetric. We will use $\tilde{x}$ to denote the skew-symmetric matrix associate with a 3-vector $x$.

# Matrix-Matrix product

Product of an $m \times n$ matrix A with a $n \times p$ matrix B

$$AB = \begin{bmatrix} a_1^T B_1 & a_1^T B_2 & \dots & a_1^T B_p \\ a_2^T B_1 & a_2^T B_2 & \dots & a_2^T B_p \\ \vdots & \vdots & \ddots & \vdots \\ a_m^T B_1 & a_m^T B_2 & \dots & a_m^T B_p \end{bmatrix}$$

- vectors $B_i$ (i.e., the columns of matrix B) and $a_j^T$ (i.e., the rows of matrix A) have dimension n.
- $a_j^T B_i$ is simply the inner product of the j-th row of A with the i-the column of B.

# Properties of matrix-matrix product

The following properties hold for all matrices A, B and C with compatible dimensions

- In general $AB \neq BA$.
  There are exceptions. For example for all $n \times n$-matrices A, we have $AI = IA$ where I is the identity matrix

$$I = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix}$$

- $(AB)^T = B^T A^T$
- $(AB)C = A(BC)$
- $(A + B)C = AC + BC$
- $a(AB) = (aA)B = A(aB)$, where $a \in \mathbb{R}$

# Orthonormal square matrices

If an $n \times n$-matrix $Q$ is orthonormal then

- its columns are mutually orthogonal
- its rows are mutually orthogonal
- all columns and rows have unit length

## Properties

- $Q^T Q = I, \quad QQ^T = I$
- $\|Qx\| = \|x\|, \ x \in \mathbb{R}^n$ (preserves length)
- $(Qx)^T(Qy) = x^T y, \ x, y \in \mathbb{R}^n$ (preserves angle between vectors)
- $Q^T = Q^{-1}$
- $\det(Q) = \pm 1$

# Matrix representation of rotation

The group of rotations in $\mathbb{R}^2$ is known as the special orthogonal group $\mathbb{SO}(2)$. These are skew-symmetric orthonormal square matrices.

An example rotation matrix: $R = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$

# Eigenvalues and Eigenvectors

$\lambda \in \mathbb{C}$ is an eigenvalue of $A \in \mathbb{R}^{n \times n}$ if there is a nonzero vector $v \in \mathbb{C}^n$ such that $Av = \lambda v$

Eigenvector v is only scaled by $\lambda$ when acted upon by A

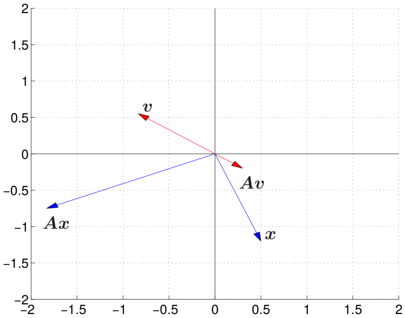



Figure 2: Scaling interpretation of eigenvector.

# Calculus

Calculus is the study of how things change

The derivative is a measurement of how a function changes when its input changes.

## Derivative of a scalar function of one variable

The derivative of $f(x)$ with respect to $x$ is given by

$$\frac{df(x)}{dx} = \lim_{\Delta x \to 0} \frac{f(x + \Delta x) - f(x)}{\Delta x}$$

## Derivative of a composite function (the chain rule)

A function composition represents the application of one function to the results of another. For instance, the functions $f : x \to y$ and $g : y \to z$ can be composed by first computing $f(x)$ and then applying a function $g$ to the output of $f(x)$. We denote this as $g(f(x))$. The chain rule is

$$\frac{dg(f(x))}{dx} = \frac{dg}{df}\frac{df}{dx}$$

# Derivative of a scalar function

**Example (scalar function of a single variable)**

$$f : x \rightarrow y, \quad x \in \mathbb{R}, \ y \in \mathbb{R}.$$

- $f$ is a scalar function because the output $y$ is a real number ($y \in \mathbb{R}$)
- $f$ is a function of a single variable because $x \in \mathbb{R}$

**Example (scalar multivariate function)**

$$f : x \rightarrow y, \quad x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \in \mathbb{R}^2, \ y \in \mathbb{R}.$$

# Partial derivative

## Partial derivative

When dealing with a multivariate function, we choose one variable of interest, and find the derivative of the function with respect to this variable, while keeping the others constant. This is called a partial derivative.

# Partial derivative

Let $f(x_1, x_2) = 2x_1^2 + 3x_2^2$, then

$$\frac{\partial f(x)}{\partial x_1} = 4x_1, \quad \frac{\partial f(x)}{\partial x_2} = 6x_2.$$

By combining the two partial derivatives, we obtain the gradient of f

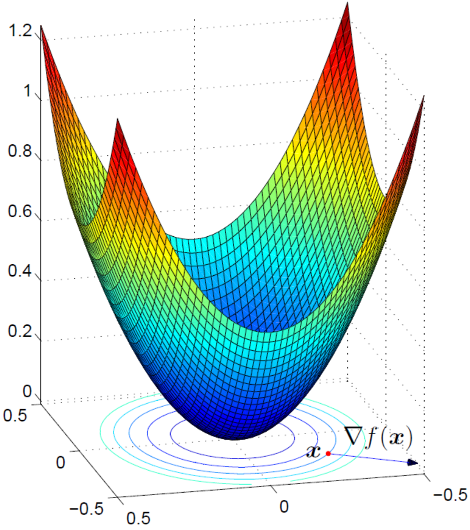
$$\nabla f(x) = \begin{bmatrix} \frac{\partial f(x)}{\partial x_1} \\ \frac{\partial f(x)}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 4x_1 \\ 6x_2 \end{bmatrix}.$$

At each point x, $\nabla f(x)$ points in the direction of greatest increase of the function.

# Example (gradient vector)

$$f(x_1, x_2) = 2x_1^2 + 3x_2^2$$
$$x = (-0.25, -0.25)$$
$$\nabla f(x) = \begin{bmatrix} -1 \\ -1.5 \end{bmatrix}$$

$\nabla f(x)$ is orthogonal to the level curve of f at x.



# Derivative of a (single variable) vector function

## Example (vector function)

$$f : x \rightarrow y, \quad x \in \mathbb{R}, \quad y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \in \mathbb{R}^2.$$

$$y = f(x) = \begin{bmatrix} 3x \\ 2x^2 \end{bmatrix}.$$

## Derivative

For $f : \mathbb{R} \rightarrow \mathbb{R}^n$ we have

$$\frac{df(x)}{dx} = \begin{bmatrix} \frac{df_1(x)}{dx} \\ \vdots \\ \frac{df_n(x)}{dx} \end{bmatrix}.$$

# Derivative of a multivariate vector function

## Example (multivariate vector function)

$$f : x \to y, \quad x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \in \mathbb{R}^3, \quad y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \in \mathbb{R}^2.$$

Alternatively, we could write

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = f(x) = \begin{bmatrix} f_1(x_1, x_2, x_3) \\ f_2(x_1, x_2, x_3) \end{bmatrix}$$

# Derivative of a multivariate vector function

## Jacobian matrix

We combine the gradients $\nabla f_1$ and $\nabla f_2$ in a matrix J

$$J = \frac{\partial f}{\partial x} = \begin{bmatrix} \nabla f_1^T \\ \nabla f_2^T \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \frac{\partial f_1}{\partial x_3} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_3} \end{bmatrix}$$

The matrix J is called the Jacobian matrix.

# Jacobian of a composite function

## Example (multivariate composite function)

$$f : x(t) \to \mathbb{R}^m \quad x(t) \in \mathbb{R}^n, t \in \mathbb{R}.$$

For $y(t) \in \mathbb{R}^m$ we can write

$$y(t) = f(x(t)).$$

## The chain rule

The derivative of $f(x(t))$ with respect to $t$ can be expressed as

$$\dot{y} = \frac{df}{dt} = \frac{\partial f}{\partial x}\frac{dx}{dt} = J\dot{x}.$$

The above equation relates the derivative of the outputs $y$ to the derivative of the inputs $x$ through the Jacobian matrix $J = \frac{\partial f}{\partial x}$.

# Outline

# From Encoder Ticks to Odometry

- Generally, odometry refers to the use of sensor data to measure the relative displacement between different positions.
- In robotics, odometry is typically used to refer to the use of wheel encoder, coupled with a forward kinematic model.
- Other types of odometry exist as well: e.g. visual odometry, or inertial measurement units (IMUs).
- The position and orientation (or pose) of the robot, given a certain motor encoder state, depends on the physical configuration of the robot.

# From Encoder Ticks to Odometry

- Generally, odometry refers to the use of sensor data to measure the relative displacement between different positions.
- In robotics, odometry is typically used to refer to the use of wheel encoder, coupled with a forward kinematic model.
- Other types of odometry exist as well: e.g. visual odometry, or inertial measurement units (IMUs).
- The position and orientation (or pose) of the robot, given a certain motor encoder state, depends on the physical configuration of the robot.

# From Encoder Ticks to Odometry

- Generally, odometry refers to the use of sensor data to measure the relative displacement between different positions.
- In robotics, odometry is typically used to refer to the use of wheel encoder, coupled with a forward kinematic model.
- Other types of odometry exist as well: e.g. visual odometry, or inertial measurement units (IMUs).
- The position and orientation (or pose) of the robot, given a certain motor encoder state, depends on the physical configuration of the robot.

# From Encoder Ticks to Odometry

- Generally, odometry refers to the use of sensor data to measure the relative displacement between different positions.
- In robotics, odometry is typically used to refer to the use of wheel encoder, coupled with a forward kinematic model.
- Other types of odometry exist as well: e.g. visual odometry, or inertial measurement units (IMUs).
- The position and orientation (or pose) of the robot, given a certain motor encoder state, depends on the physical configuration of the robot.

# Kinematic Models

- A kinematic model describes the position and orientation of all physical components of a robot, given a number of input state parameters.
- The number of independent parameters of a model are the Degrees of Freedom (DoFs) of the robot.

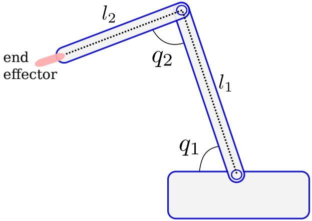



Figure: Kinematic Model of a 2 DoF manipulator

# Kinematic Models

- A kinematic model describes the position and orientation of all physical components of a robot, given a number of input state parameters.
- The number of independent parameters of a model are the Degrees of Freedom (DoFs) of the robot.

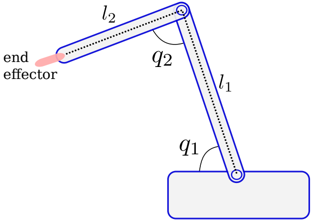



Figure: Kinematic Model of a 2 DoF manipulator

# Kinematic Models

- Forward kinematics computes the configuration of the robot, given the state variables.

$$\mathcal{K} : \mathbb{R}^{N} \rightarrow \mathbb{SE}(3)$$

- Inverse kinematics is the problem of finding the state variables, which would result in a given robot configuration.

$$\mathcal{K}^{-1} : \mathbb{SE}(3) \rightarrow \mathbb{R}^{N}$$

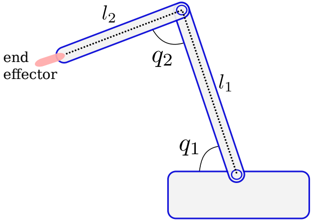



Figure: Kinematic Model of a 2 DoF manipulator

# Kinematic Models

- Forward kinematics computes the configuration of the robot, given the state variables.

$$\mathcal{K} : \mathbb{R}^N \rightarrow \mathbb{SE}(3)$$

- Inverse kinematics is the problem of finding the state variables, which would result in a given robot configuration.

$$\mathcal{K}^{-1} : \mathbb{SE}(3) \rightarrow \mathbb{R}^N$$

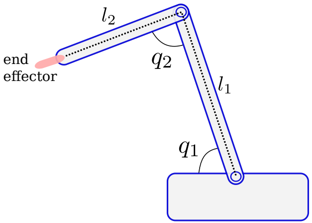



Figure: Kinematic Model of a 2 DoF manipulator

# Differential Drive

- A differential drive kinematic is based on individually controlled left and right wheels.
- The state space of this robot is $\mathrm{s} = \langle \mathrm{x}, \mathrm{y}, \theta \rangle$
- Given the control input (speeds of the two wheels in rad/s) as $\mathrm{u_l}, \mathrm{u_r}$, the motion of the robot is:

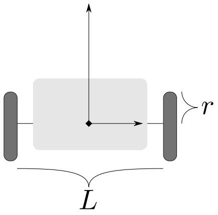



Figure: Differential Drive Kinematic Model

# Differential Drive

- A differential drive kinematic is based on individually controlled left and right wheels.
- The state space of this robot is $s = \langle x, y, \theta \rangle$
- Given the control input (speeds of the two wheels in rad/s) as $u_l, u_r$, the motion of the robot is:

$$\dot{x} = \frac{r}{2}(u_l + u_r)\cos\theta \quad (1)$$

$$\dot{y} = \frac{r}{2}(u_l + u_r)\sin\theta \quad (2)$$

$$\dot{\theta} = \frac{r}{L}(u_r - u_l) \quad (3)$$

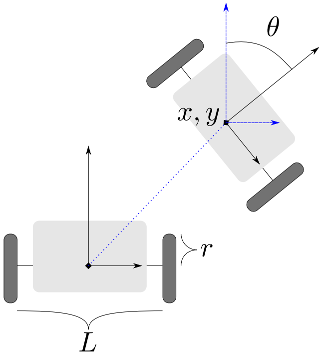



Figure: Differential Drive Kinematic Model

# Pose Space and Homogeneous Transforms

- We can obtain the new state of the robot, relative to the previous state by integrating equations 1-3 over dt.
- Tracking the robot pose is usually done relative to a stationary fixed reference system.
- We aggregate the states of the robot with the pose addition operator $\oplus$.
- The state of the robot is $s_x = s_0 \oplus s_1 \oplus s_2$, where $s_0, s_1, s_2$ are all relative to the previous state

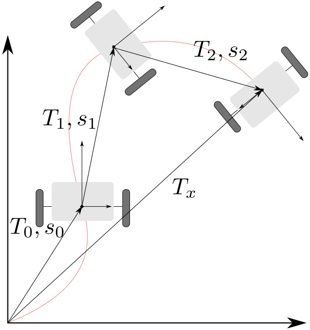



Figure: Tracking robot states

# Pose Space and Homogeneous Transforms

- An alternative representation is the Homogeneous Transform space.

$$T_i = \begin{pmatrix} R_i & t_i \\ 0 & 1 \end{pmatrix} \quad (4)$$

where $R_i$ is a rotation matrix and $t_i$ is a translation vector.

- In this example, the robot pose is constrained to a plane and thus:

$$T_i = \begin{pmatrix} \cos\theta_i & -\sin\theta_i & x_i \\ \sin\theta_i & \cos\theta_i & y_i \\ 0 & 0 & 1 \end{pmatrix} \quad (5)$$

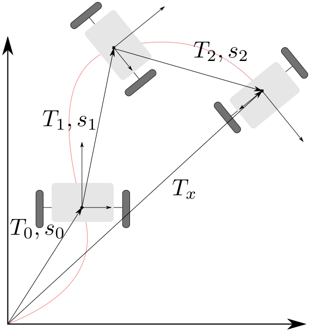



Figure: Tracking robot states

# Pose Space and Homogeneous Transforms

- The final robot pose $T_x$ relative to the global coordinate system is then obtained as:

$$T_x = T_2 T_1 T_0$$

- Note that as $T_i$ is a matrix, multiplication order matters and thus $T_a T_b \neq T_b T_a$

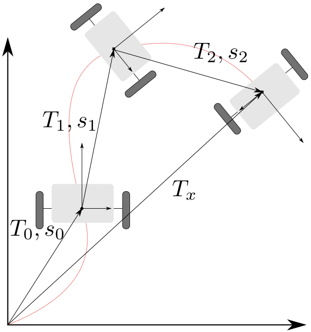



Figure: Tracking robot states

# Ackerman Drive

- An Ackerman drive is a simple model for car-like kinematics.
- The state variables of an Ackerman drive are $\mathrm{s} = \langle \mathrm{x_i}, \mathrm{y_i}, \theta_\mathrm{i}, \phi_\mathrm{i} \rangle$
- The control space $(\mathrm{v_i}, \omega_\mathrm{i})$ consists of forward velocity and turning rate.

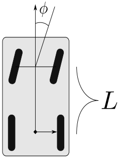



Figure: Ackerman Drive Kinematic Model

# Ackerman Drive

- An Ackerman drive is a simple model for car-like kinematics.
- The state variables of an Ackerman drive are $\mathrm{s} = \langle \mathrm{x_i}, \mathrm{y_i}, \theta_\mathrm{i}, \phi_\mathrm{i} \rangle$
- The control space $(\mathrm{v_i}, \omega_\mathrm{i})$ consists of forward velocity and turning rate.
- The state transition $\dot{\mathrm{s}}$ is:

$$\dot{\mathrm{x}}_\mathrm{i} = \mathrm{v_i} \cos \theta_\mathrm{i} \quad (6)$$

$$\dot{\mathrm{y}}_\mathrm{i} = \mathrm{v_i} \sin \theta_\mathrm{i} \quad (7)$$

$$\dot{\theta}_\mathrm{i} = \mathrm{v_i} \frac{\tan \phi_\mathrm{i}}{\mathrm{L}} \quad (8)$$

$$\dot{\phi}_\mathrm{i} = \omega_\mathrm{i} \quad (9)$$

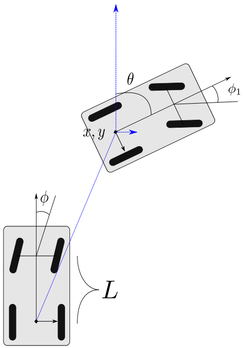



Figure: Ackerman Drive Kinematic Model

# Ackerman Drive

- The state transition $\dot{s}$ is:

$$\dot{x}_i = v_i \cos\theta_i \tag{6}$$

$$\dot{y}_i = v_i \sin\theta_i \tag{7}$$

$$\dot{\theta}_i = v_i \frac{\tan\phi_i}{L} \tag{8}$$

$$\dot{\phi}_i = \omega_i \tag{9}$$

- An Ackerman drive is a nonholonomic kinematic: i.e. not all of the DoFs are directly controllable

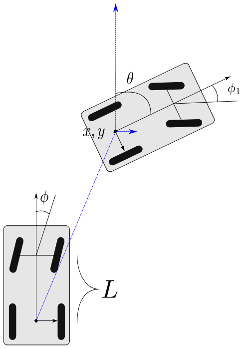



Figure: Ackerman Drive Kinematic Model

# Omnidrive

- An omni drive kinematic is based on omnidirectional wheels
- Kinematic configurations with 3, 4 or more wheels are commonly used.
- The state space is $s = \langle x, y, \theta \rangle$
- Given the speeds of each wheel $u_i$, the radius R and the angle offsets $\alpha_i$, $\dot{s}$ is

Figure: An omni wheel from Kornylak.

# Omnidrive

- An omni drive kinematic is based on omnidirectional wheels
- Kinematic configurations with 3, 4 or more wheels are commonly used.
- The state space is $s = \langle x, y, \theta \rangle$
- Given the speeds of each wheel $u_i$, the radius R and the angle offsets $\alpha_i$, $\dot{s}$ is

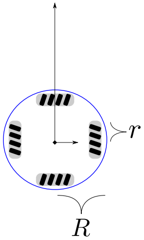



Figure: Omni Drive Kinematic Model

# Omnidrive

- An omni drive kinematic is based on omnidirectional wheels
- Kinematic configurations with 3, 4 or more wheels are commonly used.
- The state space is $\mathrm{s} = \langle \mathrm{x}, \mathrm{y}, \theta \rangle$
- Given the speeds of each wheel $\mathrm{u_i}$, the radius R and the angle offsets $\alpha_\mathrm{i}$, $\dot{\mathrm{s}}$ is

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} -\sin\alpha_1 & \dots & -\sin\alpha_n \\ \cos\alpha_1 & \dots & \cos\alpha_n \\ \frac{1}{R} & \dots & \frac{1}{R} \end{pmatrix} \begin{pmatrix} u_1 \\ .. \\ u_n \end{pmatrix} \tag{10}$$

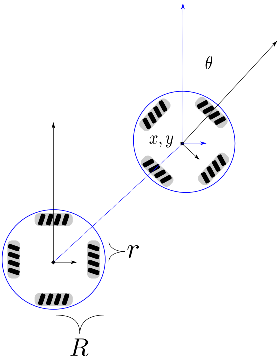



Figure: Omni Drive Kinematic Model

# Odometry Error Models

- Odometry is inherently noisy, as it integrates potentially noisy measurements over time.
- Typical sources of noise are differences in the wheel travel and wheel slip
- For a state space confined to a plane, errors in odometry are modeled as a multi-variate zero mean Gaussian $\mathcal{N}(0, \Sigma)$ with:

$$\Sigma = \begin{pmatrix} \sigma_{\mathrm{xx}} & \sigma_{\mathrm{xy}} & \sigma_{\mathrm{x}\theta} \\ \sigma_{\mathrm{yx}} & \sigma_{\mathrm{yy}} & \sigma_{\mathrm{y}\theta} \\ \sigma_{\theta\mathrm{x}} & \sigma_{\theta\mathrm{y}} & \sigma_{\theta\theta} \end{pmatrix} \tag{11}$$

- The parameters of the covariance matrix can be obtained by propagating the errors of the encoders through the kinematic model of the vehicle

# Error Propagation

Given a function $f(x) = Ax$ over a random variable x with covariance $\Sigma_x$, the covariance $\Sigma_f$ can be calculated as:

$$\Sigma_f = A\Sigma_x A^T$$

This works for linear functions. We can for example apply it directly to the omnidrive kinematic model (eq. 10):

$$\Sigma_s = \begin{pmatrix} -\sin\alpha_1 & \dots & -\sin\alpha_n \\ \cos\alpha_1 & \dots & \cos\alpha_n \\ \frac{1}{R} & \dots & \frac{1}{R} \end{pmatrix} \Sigma_u \begin{pmatrix} -\sin\alpha_1 & \dots & -\sin\alpha_n \\ \cos\alpha_1 & \dots & \cos\alpha_n \\ \frac{1}{R} & \dots & \frac{1}{R} \end{pmatrix}^T \tag{12}$$

# Error Propagation

To obtain the covarinace of the state s, we simply integrate over the time step dt.

In case the kinematic equation is not linear, we need to linearize. For example:

$$\begin{aligned} \dot{s}(t+dt) &= f(u(t+dt)) \quad (13)\\ &= f(u(t)) + \nabla f(u(t))(u(t+dt) - u(t))) \quad (14) \end{aligned}$$

Since $f(u(t))$ is constant at the linearization point, we can obtain the covariance as:

$$\Sigma_s = J \Sigma_u J^T \quad (15)$$

# Odometry Calibration

- If we have a second way of observing the state $s$, we can use these observations to calibrate the kinematic model parameters $\phi = (r, L)$
- Assume we have $z_t = s_t + \epsilon$ and $s_t = f(\phi, u(t))$
- We can now try to minimize the objective function

$$\sum_{t=0}^{t=T} \|s_t - z_t\|$$

with respect to $\phi$

# Odometry Calibration

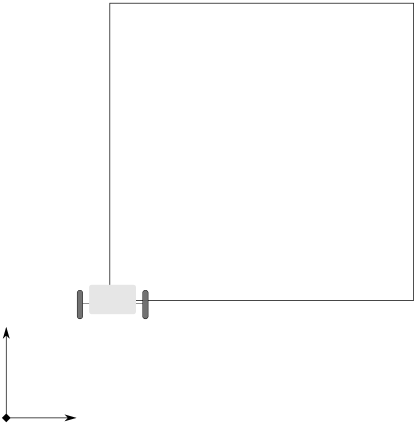


Figure: Odometry errors

# Odometry Calibration

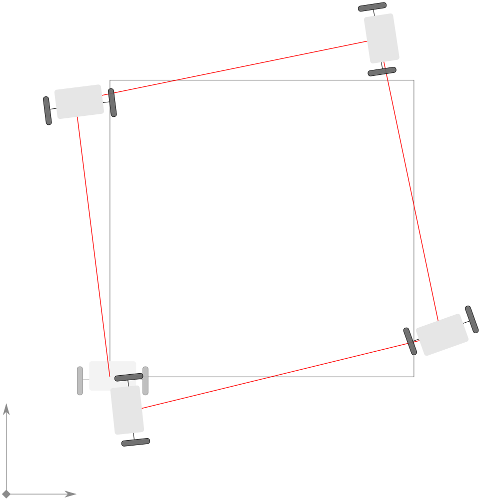


Figure: Odometry errors

# Odometry Calibration

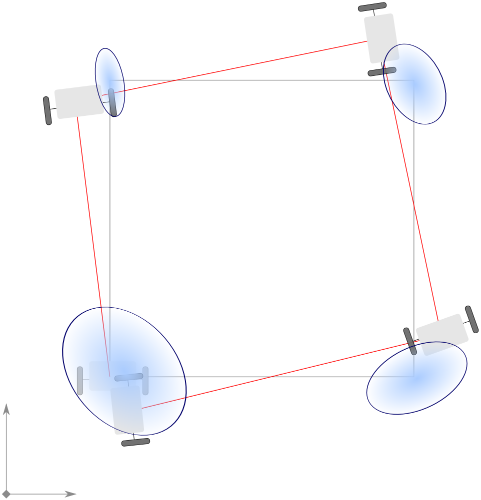


Figure: Odometry errors

# Sensors and Sensing
# Robot Kinematics and Odometry

Todor Stoyanov

Centre for Applied Autonomous Sensor Systems
Örebro University
Sweden