Mobile Robot Kinematics: Part 1

Adapted from "Principles of Robot Autonomy" by D. Gammelli, J. Lorenzetti, K. Luo, G. Zardini, and M. Pavone

January 8, 2026

Source Material

Covering the following sections from "Principles of Robot Autonomy":

1.1:

Source Material

Covering the following sections from "Principles of Robot Autonomy":

1.1:

1.1.1 covered lightly (ignoring linearity and time-invariance)

Source Material

Covering the following sections from "Principles of Robot Autonomy":

1.1:

1.1.1 covered lightly (ignoring linearity and time-invariance)

1.1.2 covering only Example 1.1.3

Motion Planning Needs Models

Motion planning and control are fundamental components of robotic autonomy.

Motion Planning Needs Models

Motion planning and control are fundamental components of robotic autonomy.

But to do either, we need some model of robot motion.

Motion Planning Needs Models

Motion planning and control are fundamental components of robotic autonomy.

But to do either, we need some model of robot motion.

We start by discussing *state* and *state-space models*.

Motion Planning Needs Models

Motion planning and control are fundamental components of robotic autonomy.

But to do either, we need some model of robot motion.

We start by discussing *state* and *state-space models*.

We will then look at the distinction between *dynamics* and *kinematics*.

Motion Planning Needs Models

Motion planning and control are fundamental components of robotic autonomy.

But to do either, we need some model of robot motion.

We start by discussing *state* and *state-space models*.

We will then look at the distinction between *dynamics* and *kinematics*.

Many robots are subject to *kinematic constraints* which are crucial to understand.

State Space Models

A state space model is a framework to describe a system (e.g. a moving robot) evolving over time.

State

The state of a system at time t_0 is a minimal set of variables $\mathbf{x}(t_0)$ such that, given the control input $\mathbf{u}(t)$ for all $t \geq t_0$, the future evolution of the system's state $\mathbf{x}(t)$ for all $t \geq t_0$ is uniquely determined.

State Space Models

A state space model is a framework to describe a system (e.g. a moving robot) evolving over time.

State

The state of a system at time t_0 is a minimal set of variables $\mathbf{x}(t_0)$ such that, given the control input $\mathbf{u}(t)$ for all $t \geq t_0$, the future evolution of the system's state $\mathbf{x}(t)$ for all $t \geq t_0$ is uniquely determined.

State Space Models

A state space model is a framework to describe a system (e.g. a moving robot) evolving over time.

State

The state of a system at time t_0 is a minimal set of variables $\mathbf{x}(t_0)$ such that, given the control input $\mathbf{u}(t)$ for all $t \geq t_0$, the future evolution of the system's state $\mathbf{x}(t)$ for all $t \geq t_0$ is uniquely determined.

Both $\mathbf{x}(t)$ and $\mathbf{u}(t)$ are vectors. $\mathbf{x}(t)$ is the *state vector* which is $n-$dimensional.

State Space Models

A state space model is a framework to describe a system (e.g. a moving robot) evolving over time.

State

The state of a system at time t_0 is a minimal set of variables $\mathbf{x}(t_0)$ such that, given the control input $\mathbf{u}(t)$ for all $t \geq t_0$, the future evolution of the system's state $\mathbf{x}(t)$ for all $t \geq t_0$ is uniquely determined.

Both $\mathbf{x}(t)$ and $\mathbf{u}(t)$ are vectors. $\mathbf{x}(t)$ is the *state vector* which is $n-$dimensional.

The state for a mobile robot will often be $\mathbf{x}(t) = [x, y, \theta]$.

State Space Models

A state space model is a framework to describe a system (e.g. a moving robot) evolving over time.

State

The state of a system at time t_0 is a minimal set of variables $\mathbf{x}(t_0)$ such that, given the control input $\mathbf{u}(t)$ for all $t \geq t_0$, the future evolution of the system's state $\mathbf{x}(t)$ for all $t \geq t_0$ is uniquely determined.

Both $\mathbf{x}(t)$ and $\mathbf{u}(t)$ are vectors. $\mathbf{x}(t)$ is the *state vector* which is $n-$dimensional.

The state for a mobile robot will often be $\mathbf{x}(t) = [x, y, \theta]$.

COMP 4766 Mobile Robot Kinematics: Part 1 January 8, 2026 4 / 12

State Space Models

A state space model is a framework to describe a system (e.g. a moving robot) evolving over time.

State

The state of a system at time t_0 is a minimal set of variables $\mathbf{x}(t_0)$ such that, given the control input $\mathbf{u}(t)$ for all $t \geq t_0$, the future evolution of the system's state $\mathbf{x}(t)$ for all $t \geq t_0$ is uniquely determined.

Both $\mathbf{x}(t)$ and $\mathbf{u}(t)$ are vectors. $\mathbf{x}(t)$ is the *state vector* which is $n-$dimensional.

The state for a mobile robot will often be $\mathbf{x}(t) = [x, y, \theta]$.

(x, y) refers to the robot's $x-$ and $y-$ coordinates relative to some coordinate frame, and θ is the robot's rotation relative to that frame.

COMP 4766 Mobile Robot Kinematics: Part 1 January 8, 2026 4 / 12

State Space Models

A state space model is a framework to describe a system (e.g. a moving robot) evolving over time.

State

The state of a system at time t_0 is a minimal set of variables $\mathbf{x}(t_0)$ such that, given the control input $\mathbf{u}(t)$ for all $t \geq t_0$, the future evolution of the system's state $\mathbf{x}(t)$ for all $t \geq t_0$ is uniquely determined.

Both $\mathbf{x}(t)$ and $\mathbf{u}(t)$ are vectors. $\mathbf{x}(t)$ is the *state vector* which is $n-$dimensional.

The state for a mobile robot will often be $\mathbf{x}(t) = [x, y, \theta]$.

(x, y) refers to the robot's $x-$ and $y-$ coordinates relative to some coordinate frame, and θ is the robot's rotation relative to that frame.

Note: $\mathbf{x}$ is a vector, whereas x is a scalar.

COMP 4766 Mobile Robot Kinematics: Part 1 January 8, 2026 4 / 12

What variables we consider part of the state is dependent both on the scenario and the level of modelling.

What variables we consider part of the state is dependent both on the scenario and the level of modelling.

Mobile robot	...with linear (v) and angular (ω) speeds	...with landmarks	Arm with 6 joint angles
$\mathbf{x} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$	$\mathbf{x} = \begin{bmatrix} x \\ y \\ \theta \\ v \\ \omega \end{bmatrix}$	$\mathbf{x} = \begin{bmatrix} x \\ y \\ \theta \\ l_{1,x} \\ l_{1,y} \\ l_{2,x} \\ l_{2,y} \\ \vdots \end{bmatrix}$	$\mathbf{x} = \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_6 \\ \dot{q}_1 \\ \dot{q}_2 \\ \vdots \\ \dot{q}_6 \end{bmatrix}$

$\mathbf{u}(t)$ is considered the system's input.

$\mathbf{u}(t)$ is considered the system's input.

The inputs could be forces, torques, voltages, commands, or whatever we have direct control over.

$\mathbf{u}(t)$ is considered the system's input.

The inputs could be forces, torques, voltages, commands, or whatever we have direct control over.

$\mathbf{u}(t)$ is $m-$dimensional.

$\mathbf{u}(t)$ is considered the system's input.

The inputs could be forces, torques, voltages, commands, or whatever we have direct control over.

$\mathbf{u}(t)$ is $m-$dimensional.

The system may also have some output $\mathbf{y}(t)$ which typically comes from sensor data (e.g. GPS or camera pixels).

$\mathbf{u}(t)$ is considered the system's input.

The inputs could be forces, torques, voltages, commands, or whatever we have direct control over.

$\mathbf{u}(t)$ is $m-$dimensional.

The system may also have some output $\mathbf{y}(t)$ which typically comes from sensor data (e.g. GPS or camera pixels).

$\mathbf{y}(t)$ is $q-$dimensional.

$\mathbf{u}(t)$ is considered the system's input.

The inputs could be forces, torques, voltages, commands, or whatever we have direct control over.

$\mathbf{u}(t)$ is $m-$dimensional.

The system may also have some output $\mathbf{y}(t)$ which typically comes from sensor data (e.g. GPS or camera pixels).

$\mathbf{y}(t)$ is $q-$dimensional.

The process of inferring the state from the system's output is known as *state estimation*.

Model

A state-space model describes the evolution of a system through *state equations* and *observation equations*.

Model

A state-space model describes the evolution of a system through *state equations* and *observation equations*.

Model

A state-space model describes the evolution of a system through *state equations* and *observation equations*.

State equations are differential equations of the following form:

Model

A state-space model describes the evolution of a system through *state equations* and *observation equations*.

State equations are differential equations of the following form:

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

Model

A state-space model describes the evolution of a system through *state equations* and *observation equations*.

State equations are differential equations of the following form:

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

where $\dot{\mathbf{x}} = \frac{dx(t)}{dt}$.

Model

A state-space model describes the evolution of a system through *state equations* and *observation equations*.

State equations are differential equations of the following form:

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

where $\dot{\mathbf{x}} = \frac{dx(t)}{dt}$.

Observations equations have this form:

$$\mathbf{y}(t) = h(\mathbf{x}(t), \mathbf{u}(t))$$

Linear Models

A state-space model which is *linear* can be expressed the following form:

Linear Models

A state-space model which is *linear* can be expressed the following form:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= A(t)\mathbf{x}(t) + B(t)\mathbf{u}(t) \\ \mathbf{y}(t) &= C(t)\mathbf{x}(t) + D(t)\mathbf{u}(t) \end{aligned}$$

Linear Models

A state-space model which is *linear* can be expressed the following form:

$$\dot{\mathbf{x}}(t) = A(t)\mathbf{x}(t) + B(t)\mathbf{u}(t)$$
$$\mathbf{y}(t) = C(t)\mathbf{x}(t) + D(t)\mathbf{u}(t)$$

$A(t)$ is the dynamics matrix ($n \times n$).

Linear Models

A state-space model which is *linear* can be expressed the following form:

$$\dot{\mathbf{x}}(t) = A(t)\mathbf{x}(t) + B(t)\mathbf{u}(t)$$
$$\mathbf{y}(t) = C(t)\mathbf{x}(t) + D(t)\mathbf{u}(t)$$

$A(t)$ is the dynamics matrix ($n \times n$).

$B(t)$ is the control matrix ($n \times m$).

Linear Models

A state-space model which is *linear* can be expressed the following form:

$$\dot{\mathbf{x}}(t) = A(t)\mathbf{x}(t) + B(t)\mathbf{u}(t)$$
$$\mathbf{y}(t) = C(t)\mathbf{x}(t) + D(t)\mathbf{u}(t)$$

$A(t)$ is the dynamics matrix ($n \times n$).

$B(t)$ is the control matrix ($n \times m$).

$C(t)$ is the output/sensor matrix ($q \times n$).

Linear Models

A state-space model which is *linear* can be expressed the following form:

$$\dot{\mathbf{x}}(t) = A(t)\mathbf{x}(t) + B(t)\mathbf{u}(t)$$
$$\mathbf{y}(t) = C(t)\mathbf{x}(t) + D(t)\mathbf{u}(t)$$

$A(t)$ is the dynamics matrix ($n \times n$).

$B(t)$ is the control matrix ($n \times m$).

$C(t)$ is the output/sensor matrix ($q \times n$).

$D(t)$ is the direct matrix ($q \times m$).

If $A(t)$, $B(t)$, $C(t)$ and $D(t)$ are actually time-invariant, we can express the model more compactly,

$$\begin{aligned}\dot{\mathbf{x}}(t) &= A\mathbf{x}(t) + B\mathbf{u}(t)\\ \mathbf{y}(t) &= C\mathbf{x}(t) + D\mathbf{u}(t).\end{aligned}$$

We will often leave off the "(t)" even when it applies (e.g. $\mathbf{x}$ is always going to be a function of time, otherwise we don't need the model).

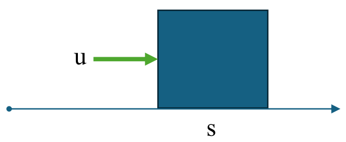
$$\begin{aligned}\dot{\mathbf{x}} &= A\mathbf{x} + B\mathbf{u}\\ \mathbf{y} &= C\mathbf{x} + D\mathbf{u}.\end{aligned}$$

Example: Double-Integrator

Consider a mass in space (i.e. no friction, no gravity). We can push the mass in a fixed direction, through an input force u.

Example: Double-Integrator

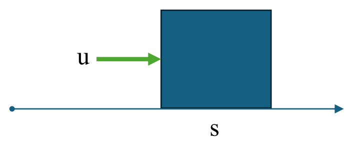
Consider a mass in space (i.e. no friction, no gravity). We can push the mass in a fixed direction, through an input force u.



s is the one-dimensional position of the mass.

Example: Double-Integrator

Consider a mass in space (i.e. no friction, no gravity). We can push the mass in a fixed direction, through an input force u.



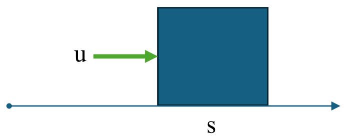
s is the one-dimensional position of the mass.

The system is governed by Newton's Second Law of Motion:

$$u = m\ddot{s}$$

Example: Double-Integrator

Consider a mass in space (i.e. no friction, no gravity). We can push the mass in a fixed direction, through an input force u.



s is the one-dimensional position of the mass.

The system is governed by Newton's Second Law of Motion:

$$u = m\ddot{s}$$

To recover s we would need to integrate $\ddot{s}$ twice, hence this is referred to as a double-integrator system.

To describe this second-order system, we define two state variables:

$x_1 = s$ (position)

To describe this second-order system, we define two state variables:

$x_1 = s$ (position)

$x_2 = \dot{s}$ (speed)

To describe this second-order system, we define two state variables:

$x_1 = s$ (position)

$x_2 = \dot{s}$ (speed)

To describe this second-order system, we define two state variables:

$x_1 = s$ (position)

$x_2 = \dot{s}$ (speed)

The derivatives of our states are:

$\dot{x}_1 = \dot{s} = x_2$

To describe this second-order system, we define two state variables:

$x_1 = s$ (position)

$x_2 = \dot{s}$ (speed)

The derivatives of our states are:

$\dot{x}_1 = \dot{s} = x_2$

$\dot{x}_2 = \ddot{s} = \frac{1}{m}u$

To describe this second-order system, we define two state variables:

$x_1 = s$ (position)

$x_2 = \dot{s}$ (speed)

The derivatives of our states are:

$\dot{x}_1 = \dot{s} = x_2$

$\dot{x}_2 = \ddot{s} = \frac{1}{m}u$

To describe this second-order system, we define two state variables:

$x_1 = s$ (position)

$x_2 = \dot{s}$ (speed)

The derivatives of our states are:

$\dot{x}_1 = \dot{s} = x_2$

$\dot{x}_2 = \ddot{s} = \frac{1}{m}u$

We can now write this in matrix form.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} u$$

To describe this second-order system, we define two state variables:

$x_1 = s$ (position)

$x_2 = \dot{s}$ (speed)

The derivatives of our states are:

$\dot{x}_1 = \dot{s} = x_2$

$\dot{x}_2 = \ddot{s} = \frac{1}{m}u$

We can now write this in matrix form.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} u$$

This is the state equation.

What about the observation equation? Actually, we can't obtain this without some definition for the output, $\mathbf{y}$.

What about the observation equation? Actually, we can't obtain this without some definition for the output, $\mathbf{y}$.

Lets define our output y (a scalar) as the position s:

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + [0]u$$

We now have the complete state-space model:

What about the observation equation? Actually, we can't obtain this without some definition for the output, $\mathbf{y}$.

Lets define our output y (a scalar) as the position s:

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + [0]u$$

We now have the complete state-space model:

Dynamics Matrix: $\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$

What about the observation equation? Actually, we can't obtain this without some definition for the output, $\mathbf{y}$.

Lets define our output y (a scalar) as the position s:

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + [0]u$$

We now have the complete state-space model:

Dynamics Matrix: $\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$

Control Matrix: $\mathbf{B} = \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix}$

What about the observation equation? Actually, we can't obtain this without some definition for the output, $\mathbf{y}$.

Lets define our output y (a scalar) as the position s:

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + [0]u$$

We now have the complete state-space model:

Dynamics Matrix: $\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$

Control Matrix: $\mathbf{B} = \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix}$

Output Matrix: $\mathbf{C} = \begin{bmatrix} 1 & 0 \end{bmatrix}$

COMP 4766 | Mobile Robot Kinematics: Part 1 | January 8, 2026 | 12 / 12

What about the observation equation? Actually, we can't obtain this without some definition for the output, $\mathbf{y}$.

Lets define our output y (a scalar) as the position s:

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + [0]u$$

We now have the complete state-space model:

Dynamics Matrix: $\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$

Control Matrix: $\mathbf{B} = \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix}$

Output Matrix: $\mathbf{C} = \begin{bmatrix} 1 & 0 \end{bmatrix}$

Direct Matrix: $\mathbf{D} = [0]$

COMP 4766 | Mobile Robot Kinematics: Part 1 | January 8, 2026 | 12 / 12