Mobile Robot Kinematics: Part 2

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":

1.2:

Source Material

Covering the following sections from “Principles of Robot Autonomy”:

- 1.2:
  - 1.2.1 - 1.2.4 covered fully

Source Material

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

- 1.2:
  - 1.2.1 - 1.2.4 covered fully
  - 1.2.5 not covered

Kinematics and Dynamics

Kinematics

Kinematics is the study of the motion of physical systems, concerned with describing positions, velocities, and accelerations over time, without reference to the forces or torques that produce the motion.

Kinematics and Dynamics

Kinematics

Kinematics is the study of the motion of physical systems, concerned with describing positions, velocities, and accelerations over time, without reference to the forces or torques that produce the motion.

Kinematics and Dynamics

Kinematics

Kinematics is the study of the motion of physical systems, concerned with describing positions, velocities, and accelerations over time, without reference to the forces or torques that produce the motion.

Dynamics

Dynamics is the study of the motion of physical systems as determined by the forces and torques acting upon them. It seeks to relate a system's motion to the underlying physical causes of that motion, such as gravity, friction, or applied inputs.

Kinematics and Dynamics

Kinematics

Kinematics is the study of the motion of physical systems, concerned with describing positions, velocities, and accelerations over time, without reference to the forces or torques that produce the motion.

Dynamics

Dynamics is the study of the motion of physical systems as determined by the forces and torques acting upon them. It seeks to relate a system's motion to the underlying physical causes of that motion, such as gravity, friction, or applied inputs.

Kinematics and Dynamics

Kinematics

Kinematics is the study of the motion of physical systems, concerned with describing positions, velocities, and accelerations over time, without reference to the forces or torques that produce the motion.

Dynamics

Dynamics is the study of the motion of physical systems as determined by the forces and torques acting upon them. It seeks to relate a system's motion to the underlying physical causes of that motion, such as gravity, friction, or applied inputs.

Both perspectives are necessary to accurately describe a robot's motion. However, we might choose to focus on only one to yield a simpler model.

In mobile robotics we often ignore dynamics: why?

Most mobile robots utilize a layered control architecture. These controllers handle the “dirty work” of dynamics (voltage, torque, friction) so that the higher-level system can simply treat the robot as a device that accepts a velocity.

In mobile robotics we often ignore dynamics: why?

Most mobile robots utilize a layered control architecture. These controllers handle the “dirty work” of dynamics (voltage, torque, friction) so that the higher-level system can simply treat the robot as a device that accepts a velocity.

Kinematic models are sufficient when physical forces are small: Many indoor robots move at speeds where centrifugal forces and inertia are small.

In mobile robotics we often ignore dynamics: why?

Most mobile robots utilize a layered control architecture. These controllers handle the "dirty work" of dynamics (voltage, torque, friction) so that the higher-level system can simply treat the robot as a device that accepts a velocity.

Kinematic models are sufficient when physical forces are small: Many indoor robots move at speeds where centrifugal forces and inertia are small.

Solving dynamics equations is typically much slower than kinematic equations. Dynamics equations often require more knowledge of parameters (e.g. mass, centre-of-mass, friction coefficients, ...).

When Dynamics CANNOT be Ignored

Kinematics-only if...	Incorporate Dynamics if...
Slow/Moderate speed	High-speed maneuvers
Heavy-duty motors	Limited torque/underactuated
Wheeled/Flat ground	Legged or Aerial robots
Rigid environment	Soft or slippery terrain

Generalized Coordinates

The *generalized coordinates* of a robot is the set of variables that specify the position of every point on the robot.

Generalized Coordinates

The *generalized coordinates* of a robot is the set of variables that specify the position of every point on the robot.

Denoted $\mathbf{q}(t) \in \mathbb{R}^{n_g}$.

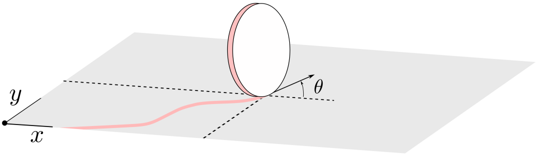
Generalized Coordinates

The *generalized coordinates* of a robot is the set of variables that specify the position of every point on the robot.

Denoted $\mathbf{q}(t) \in \mathbb{R}^{n_g}$.

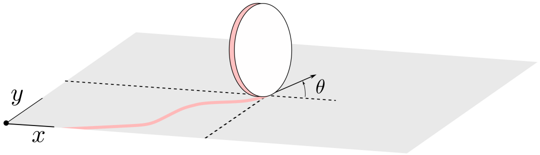
The generalized coordinates may be the same as the full system state $\mathbf{x}$ or they may represent a subset of the state (e.g. the state may include the generalized coordinates plus velocities).

Example: A Wheel



(x, y) are the coordinates of the wheel's contact point with the ground.

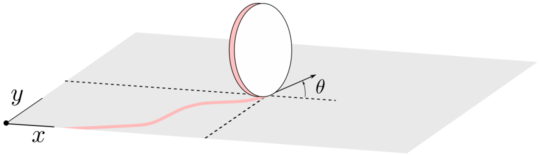
Example: A Wheel



(x, y) are the coordinates of the wheel's contact point with the ground.

θ is the heading angle, relative to the $x - axis$.

Example: A Wheel

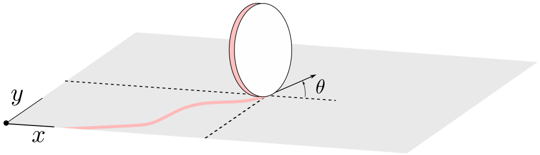


(x, y) are the coordinates of the wheel's contact point with the ground.

θ is the heading angle, relative to the $x - axis$.

Generalized coordinates: $\boldsymbol{q} = [x, y, \theta]^\top$.

Example: A Wheel



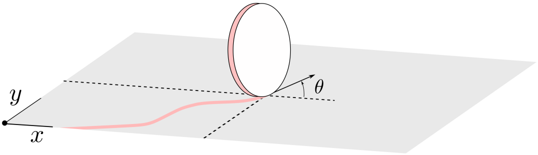
(x, y) are the coordinates of the wheel's contact point with the ground.

θ is the heading angle, relative to the $x - axis$.

Generalized coordinates: $\boldsymbol{q} = [x, y, \theta]^\top$.

Note that other choices of generalized coordinates are possible. For example, rather than the Cartesian coordinates (x, y) we could use polar coordinates.

Example: A Wheel



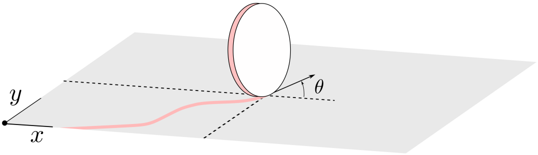
(x, y) are the coordinates of the wheel's contact point with the ground.

θ is the heading angle, relative to the $x - axis$.

Generalized coordinates: $\boldsymbol{q} = [x, y, \theta]^\top$.

Note that other choices of generalized coordinates are possible. For example, rather than the Cartesian coordinates (x, y) we could use polar coordinates.

Example: A Wheel



(x, y) are the coordinates of the wheel's contact point with the ground.

θ is the heading angle, relative to the $x - axis$.

Generalized coordinates: $\boldsymbol{q} = [x, y, \theta]^\top$.

Note that other choices of generalized coordinates are possible. For example, rather than the Cartesian coordinates (x, y) we could use polar coordinates.

Kinematic Constraints

Kinematic Constraints

Kinematic constraints are a set of constraints imposed on the generalized coordinates, $\boldsymbol{q}$, and generalized velocities, $\dot{\boldsymbol{q}}$. We express kinematic constraints mathematically as:

$$a_i(\boldsymbol{q}, \dot{\boldsymbol{q}}) = 0, \qquad i = 1, ..., k < n_g,$$

where k is the number of constraints and n_g is the number of generalized coordinates.

Kinematic Constraints

Kinematic Constraints

Kinematic constraints are a set of constraints imposed on the generalized coordinates, $\boldsymbol{q}$, and generalized velocities, $\dot{\boldsymbol{q}}$. We express kinematic constraints mathematically as:

$$a_i(\boldsymbol{q}, \dot{\boldsymbol{q}}) = 0, \qquad i = 1, ..., k < n_g,$$

where k is the number of constraints and n_g is the number of generalized coordinates.

Kinematic Constraints

Kinematic Constraints

Kinematic constraints are a set of constraints imposed on the generalized coordinates, $\boldsymbol{q}$, and generalized velocities, $\dot{\boldsymbol{q}}$. We express kinematic constraints mathematically as:

$$a_i(\boldsymbol{q}, \dot{\boldsymbol{q}}) = 0, \qquad i = 1, ..., k < n_g,$$

where k is the number of constraints and n_g is the number of generalized coordinates.

Intuitively: a kinematic constraint tells us one or both of the following:

"you can't be here"

Kinematic Constraints

Kinematic Constraints

Kinematic constraints are a set of constraints imposed on the generalized coordinates, $\boldsymbol{q}$, and generalized velocities, $\dot{\boldsymbol{q}}$. We express kinematic constraints mathematically as:

$$a_i(\boldsymbol{q}, \dot{\boldsymbol{q}}) = 0, \qquad i = 1, ..., k < n_g,$$

where k is the number of constraints and n_g is the number of generalized coordinates.

Intuitively: a kinematic constraint tells us one or both of the following:

"you can't be here"

"you can't move in this direction"

Pfaffian Constraints

Many robotic constraints are linear in $\dot{\boldsymbol{q}}$, yielding Pfaffian form

$$\mathbf{a}_i^\top(\boldsymbol{q})\dot{\boldsymbol{q}} = 0, \qquad i = 1, \ldots, k < n_g.$$

Pfaffian Constraints

Many robotic constraints are linear in $\dot{\boldsymbol{q}}$, yielding Pfaffian form

$$\mathbf{a}_i^\top(\boldsymbol{q})\dot{\boldsymbol{q}} = 0, \qquad i = 1, ..., k < n_g.$$

Example: we are modelling a robot that lives only on the $x-$axis, but we choose to represent its generalized coordinates as $\boldsymbol{q} = [x, y]^\top$.

Pfaffian Constraints

Many robotic constraints are linear in $\dot{\boldsymbol{q}}$, yielding Pfaffian form

$$\mathbf{a}_i^\top(\boldsymbol{q})\dot{\boldsymbol{q}} = 0, \qquad i = 1, ..., k < n_g.$$

Example: we are modelling a robot that lives only on the $x-$axis, but we choose to represent its generalized coordinates as $\boldsymbol{q} = [x, y]^\top$.

We could constrain its velocity to be along the $x-$axis through the following Pfaffian-form kinematic constraint:

Pfaffian Constraints

Many robotic constraints are linear in $\dot{\boldsymbol{q}}$, yielding Pfaffian form

$$\mathbf{a}_i^\top(\boldsymbol{q})\dot{\boldsymbol{q}} = 0, \qquad i = 1, ..., k < n_g.$$

Example: we are modelling a robot that lives only on the $x-$axis, but we choose to represent its generalized coordinates as $\boldsymbol{q} = [x, y]^\top$.

We could constrain its velocity to be along the $x-$axis through the following Pfaffian-form kinematic constraint:

$$\begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = 0$$

Pfaffian Constraints

Many robotic constraints are linear in $\dot{\boldsymbol{q}}$, yielding Pfaffian form

$$\mathbf{a}_i^\top(\boldsymbol{q})\dot{\boldsymbol{q}} = 0, \qquad i = 1, \ldots, k < n_g.$$

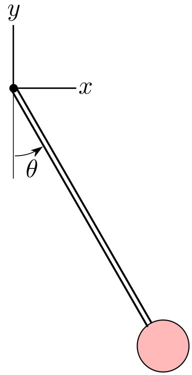
Example: we are modelling a robot that lives only on the $x-$axis, but we choose to represent its generalized coordinates as $\boldsymbol{q} = [x, y]^\top$.

We could constrain its velocity to be along the $x-$axis through the following Pfaffian-form kinematic constraint:

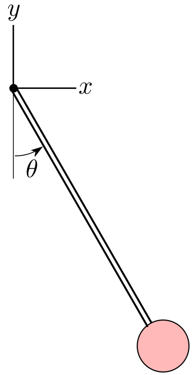
$$\begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = 0$$

Meanwhile, this example is rather contrived. It would be better to just represent $\boldsymbol{q}$ as x.

Example: Pendulum Constraint

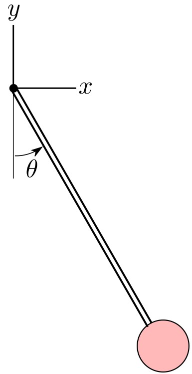


Example: Pendulum Constraint



Consider a pendulum: a mass at the end of a rod, swinging freely from the origin.

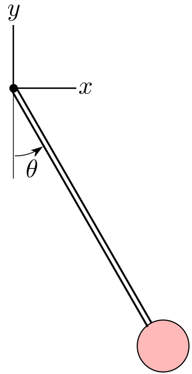
Example: Pendulum Constraint



Consider a pendulum: a mass at the end of a rod, swinging freely from the origin.

Choose $\boldsymbol{q} = [x, y]^\top$ wherer (x, y) are the coordinates of the mass.

Example: Pendulum Constraint

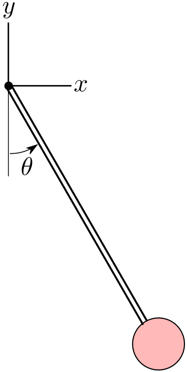


Consider a pendulum: a mass at the end of a rod, swinging freely from the origin.

Choose $\boldsymbol{q} = [x, y]^\top$ wherer (x, y) are the coordinates of the mass.

The rod is assumed to be rigid and of length L. This assumption yields a kinematic constraint.

Example: Pendulum Constraint

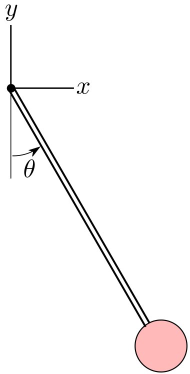


Consider a pendulum: a mass at the end of a rod, swinging freely from the origin.

Choose $\boldsymbol{q} = [x, y]^\top$ wherer (x, y) are the coordinates of the mass.

The rod is assumed to be rigid and of length L. This assumption yields a kinematic constraint.

Example: Pendulum Constraint



Consider a pendulum: a mass at the end of a rod, swinging freely from the origin.

Choose $\boldsymbol{q} = [x, y]^\top$ wherer (x, y) are the coordinates of the mass.

The rod is assumed to be rigid and of length L. This assumption yields a kinematic constraint.

Example: No-Slip Wheel

A rolling wheel with coordinates $[x, y, \theta]^\top$ maintains zero lateral velocity.

Example: No-Slip Wheel

A rolling wheel with coordinates $[x, y, \theta]^\top$ maintains zero lateral velocity.

Using basis vectors aligned with the wheel gives the constraint $\dot{x}\sin\theta - \dot{y}\cos\theta = 0$.

Example: No-Slip Wheel

A rolling wheel with coordinates $[x, y, \theta]^\top$ maintains zero lateral velocity.

Using basis vectors aligned with the wheel gives the constraint $\dot{x}\sin\theta - \dot{y}\cos\theta = 0$.

The Pfaffian form keeps the constraint linear in $\dot{x}$ and $\dot{y}$.

Example: No-Slip Wheel

A rolling wheel with coordinates $[x, y, \theta]^\top$ maintains zero lateral velocity.

Using basis vectors aligned with the wheel gives the constraint $\dot{x}\sin\theta - \dot{y}\cos\theta = 0$.

The Pfaffian form keeps the constraint linear in $\dot{x}$ and $\dot{y}$.

Example: No-Slip Wheel

A rolling wheel with coordinates $[x, y, \theta]^\top$ maintains zero lateral velocity.

Using basis vectors aligned with the wheel gives the constraint $\dot{x}\sin\theta - \dot{y}\cos\theta = 0$.

The Pfaffian form keeps the constraint linear in $\dot{x}$ and $\dot{y}$.

$$\dot{x}\sin\theta - \dot{y}\cos\theta = 0$$

Holonomic Constraints

Holonomic constraints depend only on configuration: $h_i(\boldsymbol{\xi}) = 0$.

Holonomic Constraints

Holonomic constraints depend only on configuration: $h_i(\boldsymbol{\xi}) = 0$.

Differentiating h_i reproduces the equivalent Pfaffian form.

Holonomic Constraints

Holonomic constraints depend only on configuration: $h_i(\boldsymbol{\xi}) = 0$.

Differentiating h_i reproduces the equivalent Pfaffian form.

Holonomic systems contain only such integrable constraints.

Holonomic Constraints

Holonomic constraints depend only on configuration: $h_i(\boldsymbol{\xi}) = 0$.

Differentiating h_i reproduces the equivalent Pfaffian form.

Holonomic systems contain only such integrable constraints.

Holonomic Constraints

Holonomic constraints depend only on configuration: $h_i(\boldsymbol{\xi}) = 0$.

Differentiating h_i reproduces the equivalent Pfaffian form.

Holonomic systems contain only such integrable constraints.

$$\frac{dh_i(\boldsymbol{\xi})}{dt} = \frac{dh_i(\boldsymbol{\xi})}{d\boldsymbol{\xi}}\dot{\boldsymbol{\xi}} = 0$$

Holonomic Access

Holonomic constraints restrict the reachable configurations to an $(n-k)$-dimensional subset.

Holonomic Access

Holonomic constraints restrict the reachable configurations to an $(n-k)$-dimensional subset.

A pendulum mass remains on a circle of radius L, reducing the space to one dimension.

Holonomic Access

Holonomic constraints restrict the reachable configurations to an $(n - k)$-dimensional subset.

A pendulum mass remains on a circle of radius L, reducing the space to one dimension.

The no-slip wheel constraint is not integrable, so it does not limit which (x, y) can be reached.

Nonholonomic Constraints

Nonholonomic constraints are Pfaffian but cannot be integrated to $h_i(\boldsymbol{\xi}) = 0$.

Nonholonomic Constraints

Nonholonomic constraints are Pfaffian but cannot be integrated to $h_i(\boldsymbol{\xi}) = 0$.

They restrict allowable motions between configurations rather than the configurations themselves.

Nonholonomic Constraints

Nonholonomic constraints are Pfaffian but cannot be integrated to $h_i(\boldsymbol{\xi}) = 0$.

They restrict allowable motions between configurations rather than the configurations themselves.

Any system subject to at least one such constraint is nonholonomic.

Reminder: Null Space

Concept Recap

Reminder: Null Space

Concept Recap

The null space of a matrix A is the set of all vectors $\mathbf{x}$ such that $A\mathbf{x} = \mathbf{0}$, where $\mathbf{0}$ denotes the zero vector.

Reminder: Null Space

Concept Recap

The null space of a matrix A is the set of all vectors $\mathbf{x}$ such that $A\mathbf{x} = \mathbf{0}$, where $\mathbf{0}$ denotes the zero vector.

Emphasizing the zero vector clarifies that every component of $A\mathbf{x}$ must vanish simultaneously.

Reminder: Null Space

Concept Recap

The null space of a matrix A is the set of all vectors $\mathbf{x}$ such that $A\mathbf{x} = \mathbf{0}$, where $\mathbf{0}$ denotes the zero vector.

Emphasizing the zero vector clarifies that every component of $A\mathbf{x}$ must vanish simultaneously.

Example: For $A = \begin{bmatrix} 1 & 2 \\ 3 & 6 \end{bmatrix}$, any scalar multiple of $\begin{bmatrix} 2 \\ -1 \end{bmatrix}$ lies in the null space because $A \begin{bmatrix} 2 \\ -1 \end{bmatrix} = \mathbf{0}$.

Instantaneous Motion Limits

Nonholonomic constraints confine $\dot{\boldsymbol{\xi}}$ to the null space of $a_i^\top(\boldsymbol{\xi})$.

Instantaneous Motion Limits

Nonholonomic constraints confine $\dot{\boldsymbol{\xi}}$ to the null space of $a_i^\top(\boldsymbol{\xi})$.

For the wheel, the null space is spanned by the rolling direction and pure spin.

Instantaneous Motion Limits

Nonholonomic constraints confine $\dot{\boldsymbol{\xi}}$ to the null space of $a_i^\top(\boldsymbol{\xi})$.

For the wheel, the null space is spanned by the rolling direction and pure spin.

Any feasible velocity is a combination of those motions.

Building Kinematic Models

After identifying constraints, express motion as $\dot{\boldsymbol{\xi}} = G(\boldsymbol{\xi})\boldsymbol{u}$.

Building Kinematic Models

After identifying constraints, express motion as $\dot{\boldsymbol{\xi}} = G(\boldsymbol{\xi})\boldsymbol{u}$.

Columns of $G(\boldsymbol{\xi})$ span the null space of $A^\top(\boldsymbol{\xi})$, guaranteeing constraint satisfaction.

Building Kinematic Models

After identifying constraints, express motion as $\dot{\boldsymbol{\xi}} = G(\boldsymbol{\xi})\boldsymbol{u}$.

Columns of $G(\boldsymbol{\xi})$ span the null space of $A^\top(\boldsymbol{\xi})$, guaranteeing constraint satisfaction.

Inputs $\boldsymbol{u}$ parameterize allowable motions of dimension $n - k$.

Example: Wheel Kinematic Model

The wheel's constraint matrix has a one-dimensional row space, so $G(\boldsymbol{\xi})$ has two columns.

Example: Wheel Kinematic Model

The wheel's constraint matrix has a one-dimensional row space, so $G(\boldsymbol{\xi})$ has two columns.

Choose columns $[\cos\theta \;\; \sin\theta \; 0]^\top$ (rolling) and $[0 \; 0 \; 1]^\top$ (spin).

Example: Wheel Kinematic Model

The wheel's constraint matrix has a one-dimensional row space, so $G(\boldsymbol{\xi})$ has two columns.

Choose columns $[\cos\theta \;\; \sin\theta \; 0]^\top$ (rolling) and $[0\;0\;1]^\top$ (spin).

Inputs u_1 and u_2 become forward speed and rotational rate, respectively.

Example: Wheel Kinematic Model

The wheel's constraint matrix has a one-dimensional row space, so $G(\boldsymbol{\xi})$ has two columns.

Choose columns $[\cos\theta \ \ \sin\theta \ 0]^\top$ (rolling) and $[0 \ 0 \ 1]^\top$ (spin).

Inputs u_1 and u_2 become forward speed and rotational rate, respectively.

Example: Wheel Kinematic Model

The wheel's constraint matrix has a one-dimensional row space, so $G(\boldsymbol{\xi})$ has two columns.

Choose columns $[\cos\theta \;\; \sin\theta \; 0]^\top$ (rolling) and $[0 \; 0 \; 1]^\top$ (spin).

Inputs u_1 and u_2 become forward speed and rotational rate, respectively.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$