· LESSON 07 ·

Forward and Inverse Kinematics: Conceptual Introduction

Lesson Overview

In this lesson, and throughout this week, we focus on the introduction and application of kinematics, instrumental in the construction of mechatronics devices. You will learn the following:

- Why is kinematics required?
- Degrees of Freedom, trajectories, end-effectors and many other terms
- Forward and inverse kinematics

What is kinematics?

Kinematics pertains to the motion of bodies in a robotic mechanism without regard to the forces/torques that cause the motion. Since robotic mechanisms are by their very essence designed for motion, kinematics is the most fundamental aspect of robot design, analysis, control, and simulation. The Robotics community has focused on efficiently applying different representations of position and orientation and their derivatives with respect to time to solve foundational kinematics problems.

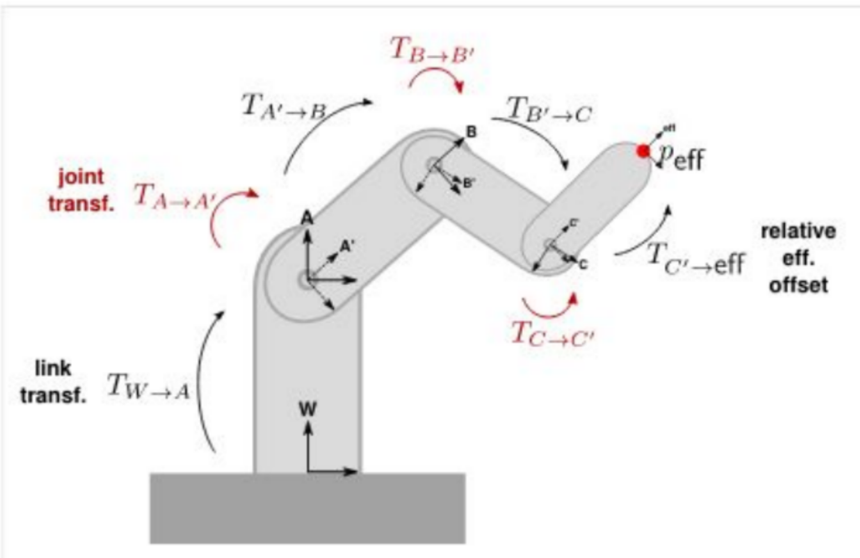
Kinematics is not concerned with forces and moments. That's the business of dynamics. So in kinematics people study mechanisms and try to understand how the motion of one body is related to the motion of another body.

Why do we need kinematics here?

Robots can make some pretty strange motions. This is true even despite the continual improvements in actuation and motion planning algorithms. Sometimes, you command a seemingly simple move from Point A to Point B, only for the robot to complete the instruction in the most convoluted way possible. It might completely change the configuration of all its joints before coming back to what seems to be exactly the same spot.

What causes robots to move in ways that we don't expect?

Often, it's because we don't have a full, intuitive understanding of the robot's workspace and kinematics.



Kinematic structure is a graph (usually tree or chain) of rigid links and joints.

Configuration space and workspace

A robot’s kinematic structure is defined by a set of links, which for most cases are defined to be rigid bodies, and joints connecting them constraining their relative movement, for example rotational or translational joints.

A robot's layout, at some instant in time, can be described by one of two methods:

1. A list of coordinates for each joint (typically an angle or translation distance) expressed relative to some reference frame, aka zero position.
2. A spatial representation of its links in the 2D or 3D world in which it operates, e.g., matrices describing the frame of each link relative to some world.

A wide variety of robot mechanisms can be described by categorizing their arrangement of joints and joint types. For the moment we will ignore the size and shape of links, and simply focus on broad categorization.

First, there are three typical joint types, each describing the form of relative transformations allowed between the two links to which it is attached:

Revolute: the attached links rotate about a common axis.

Prismatic: the attached links translate about a common axis.

Spherical: the attached links rotate about a point.

Second, mechanisms can be described by their topology, which describes how links and joints interconnect:

Serial: the links and joints form a single ordered chain, with the child link of one joint being the parent of the next.

Branched: each link can have zero or more child links, but cutting any joint would detach the system into two disconnected mechanisms. Like a human body, in which fingers are attached to the hand, toes are attached to the feet, and arms, legs, and head are attached to the torso.

Parallel: the series of joints forms at least one closed loop. I.e., there exist joints that, if cut, would not divide the system into two disconnected halves.

A third characterization defines whether the robot is affixed to the world or left free to move in space:

Fixed base: a base link is rigidly affixed to the world, like in an industrial robot.

Floating base: all links are free to rotate and translate in the workspace, like in a humanoid robot.

Mobile base: the workspace is 3D, but a base link can rotate and translate on a 2D plane, like in a car

Configurations and configuration space

As mentioned above, the configuration of a robot is a minimal set of coordinates defining the position of all links. For serial or branched fixed-base mechanisms, this is simply a list of individual joint coordinates. For floating/mobile bases, the configuration is slightly more complex, requiring the introduction of *virtual linkages* to account for the movement of the base link. The situation for parallel mechanisms is even more complex, and we will withhold this discussion for later.

1. Degrees of Freedom

The *degrees of freedom* (dof) of a system define the span of its freely and independently moving dimensions, and the number of degrees of freedom is also known as its *mobility* *M*. In the case of a serial or branched fixed base mechanism, the degrees of freedom are the union of all individual joint degrees of freedom, and the mobility is the sum of the mobilities of all individual joints:

$$M = \sum_{i=1}^{n} f_i$$

where there are *n* joints and f_i is the mobility of the *i*'th joint, with $f_i = 1$ for revolute, prismatic, and helical joints, and $f_i = 3$ for spherical joints.

The degrees of freedom for a single joint are expressed as the offset of the two attached links from their layout in a given *reference frame*. For revolute joints, the one dof is a joint angle defining the offset from a joint's zero position along its axis of rotation. For prismatic joints, the one dof is a translation along the axis relative to its zero position. Spherical joint dofs can be represented by Euler angles.

2. Floating bases and virtual linkages

For floating and mobile bases, the movement of the robot takes place not only via joint movement but also of the overall translation and rotation of the mechanism in space. As a result the number of degrees of freedom are increased. To represent this in a more straightforward manner, we treat floating base robots as fixed-base robots by means of attaching a virtual linkage that expresses the mobility of the root link.

In 3D floating base robots, the virtual linkage is customarily treated as a 3P3R robot with degrees of freedom corresponding to the (x, y, z) translation of the root link and the Euler angle representation (ϕ, θ, ψ) of its rotation. Any Euler angle convention may be used for this linkage, except that it is often advisable not to use conventions that have a singularity at the identity. In the future we shall use roll-pitch-yaw (ZYX) convention.

As a result of the inclusion of the virtual linkage, for a floating base in 3D, the mobility is increased by 6:

$M = 6 + \sum_{i=1}^{n} f_i$. In 2D, or for mobile bases in 3D, mobility is increased by 3.

3. Configuration for parallel mechanism

Joint mobility is usually limited by mechanical limitations or physical stops. Such prismatic and revolute joints will be associated with *joint limits*, which define an interval of joint values $[a,b]$ that are valid irrespective of the configuration of the remaining links. Some revolute joints may have no stops, such as a motor driving a drill bit or wheel, and these are known as continuous rotation joints. The revolute joints associated with virtual linkages also have continuous rotation. In these cases, the joint's degree of freedom moves in $SO(2)$.

4. Configurations for parallel mechanisms and end effectors

Often, it is significantly harder to determine the configuration space of parallel mechanisms. We can no longer consider each joint independent, since the movement of each joint in a closed loop affects the movement of other joints. However, there is a formula to determine the mobility M of these mechanisms.

Conceptually, the formula calculates the number of dofs of the maximal coordinate representation, and then subtracts the number of dofs removed by each joint. That is, if there are n links and m joints, each with mobility $f_1, f_2, , f_m$, then the mobility is given by

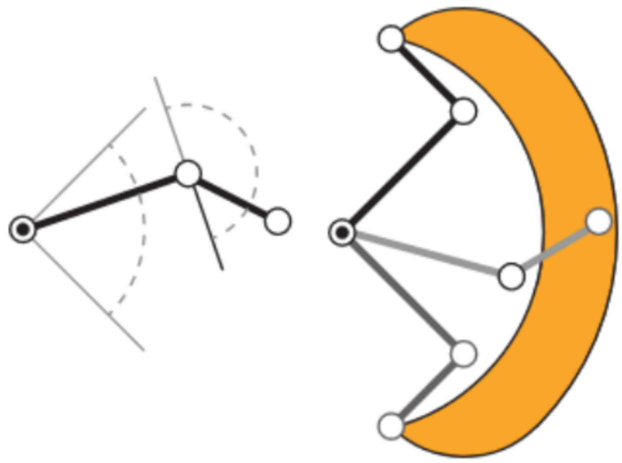
$$M = 3n - \sum_{j=1}^{m} (3 - f_j) \tag{3}$$

in 2D and

$$M = 6n - \sum_{j=1}^{m} (6 - f_j) \tag{4}$$

in 3D.

"Workspace" is somewhat of an overloaded term in robotics; it is also used to refer to the range of positions and orientations of a certain privileged link, known as the *end effector*. End effectors are typically at the far end of a serial chain of links, and are often where tool points are located since these links have the largest range of motion.



Above figure is an example of a 2R manipulator, with joint limits and a second link somewhat shorter than the first has a reachable (position) workspace that is a portion of an annulus (planar donut shape).

Now that we have a fair idea about why kinematics is used in robotics and mechatronics, let us have a look at the types of kinematics involved, forward and inverse kinematics.

FORWARD KINEMATICS

Forward kinematics is the process of calculating the frames of a robot's links, given a configuration and the robot's kinematic structure as input. The forward kinematics of a robot can be mathematically derived in closed form, which is useful for further analysis during mechanism design, or it can be computed in a software library in microseconds for tasks like motion prediction, collision detection, or rendering.

We shall only describe forward kinematics for serial and articulated robots.

INVERSE KINEMATICS

Inverse kinematics is the mathematical process of calculating the variable joint parameters needed to place the end of a kinematic chain, such as a robot manipulator or animation character's skeleton, in a given position and orientation relative to the start of the chain.

Inverse kinematics is also used to recover the movements of an object in the world from some other data, such as a film of those movements, or a film of the world as seen by a camera which is itself making those movements. This occurs, for example, where a human actor's filmed movements are to be duplicated by an animated character.

Refer to the attached slides for more info!