Differential Kinematics

Elizabeth Sklar

Department of Informatics
King's College London

12 November 2018
(version 1.0)

Outline

Differential Kinematics

We have talked about:

- Forward Kinematics—figuring out the end point of a robot arm given its starting point and angle for each link
- Inverse Kinematics—figuring out the angles necessary for each link, given a starting and ending point for a robot arm
- Mostly we have talked about this in the context of figuring out what positions to end up in, but not about how to get to these positions.
- So now we'll talk about how to get there.

- One way to program robots to move is through trial and error.
- We have talked about this kind of motion for mobile robots with path planning.

- With robot arms, it is more common to solve the problem mathematically, using Kinematics.
- So you can think of:
- Forward Kinematics as figuring out how a robot will move if its motors work in a given way.

- Inverse Kinematics as figuring out how to move the motors to get the robot to do what we want.

Differential Kinematics, p3

Here is a mathematical way to represent the problem.

Given:

- n joints:
  $\theta_1 \ldots \theta_n$ joint angles
- k end effectors:
  $\overrightarrow{s} = (s_1 \ldots s_k)^T$ end effector starting positions
  $\overrightarrow{t} = (t_1 \ldots t_k)^T$ end effector target positions

- Each position is a coordinate in space, e.g., (x, y) or (x, y, z).
- $e_i = t_i - s_i$ desired change in position of end effector i

- The Inverse Kinematics problem:
  Find values for all θ_j's such that $t_i = s_i(\theta)\ \forall\ i$
- There may be multiple solutions (if there is a solution at all).

KING'S College LONDON

- There are many ways to solve this problem.
- We will look at one incremental way.
- The advantage of an incremental approach is that it can also be used to choreograph the movement of the arm

- There are multiple incremental approaches, such as:
- Gradient Descent — start with an initial guess, and incrementally change values to reduce error
- Jacobian — we'll look at this one here

- The Jacobian approach is based on a matrix of partial derivatives, where each partial indicates the influence on the end effector position (p_i) by the joint angle (θ_j).
- The Jacobian matrix J is defined as:

$$J(\theta) = \left[\frac{\partial p_i}{\partial \theta_j}\right]_{i,j}$$

where $\frac{\partial p_i}{\partial \theta_j}$ is the partial derivative of p_i with respect to θ_j

- We assume that J is an $m \times n$ matrix of partial derivatives:

$$J(\theta) = \begin{bmatrix} \frac{\partial p_x}{\partial \theta_0} & \frac{\partial p_x}{\partial \theta_1} & \cdots & \frac{\partial p_x}{\partial \theta_n} \\ \frac{\partial p_y}{\partial \theta_0} & \frac{\partial p_y}{\partial \theta_1} & \cdots & \frac{\partial p_y}{\partial \theta_n} \\ \frac{\partial \omega}{\partial \theta_0} & \frac{\partial \omega}{\partial \theta_1} & \cdots & \frac{\partial \omega}{\partial \theta_n} \end{bmatrix}$$

($m = 3$ in the case of a 2-DOF arm)

- Before we go further let's have a quick review of Derivatives...

- A derivative is a "measure of the rate of change of one quantity with respect to another." [MW, p47]
- The slope of a line is a derivative: change in x with respect to change in y.

- A velocity is another derivative: change in *distance* with respect to *time*.
- Definition:

  The derivative $y = f(x)$ at x_0 is written $f'(x_0)$:

$$f'(x_0) = \frac{\Delta y}{\Delta x} = \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x}$$

KING'S College LONDON

Partial Derivatives

- The partial derivatives "of a function of several variables are its ordinary derivatives with respect to each variable separately." [MW, p686]

- When we take the partial derivative of a function with two variables, we imagine one of the variables is a constant and compute the partial with respect to the other variable.

- For example, given:

$$f(x) = x^2 + xy + y^2$$

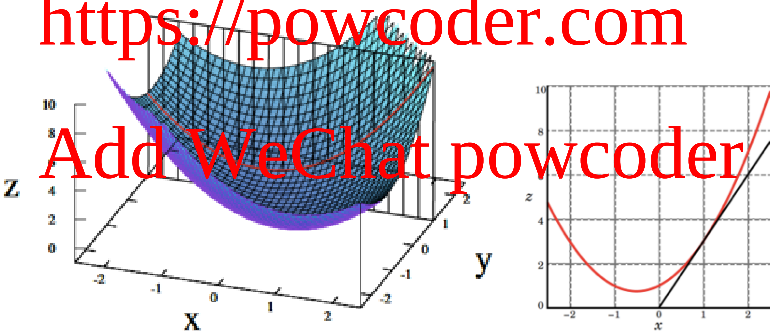
the partial derivative of $f(x)$ with respect to x is:

$$\frac{\partial f(x)}{\partial x} = 2x + y$$

- We pretend that y is a constant and compute the derivative of the function $f(x)$ with respect to x.

KING'S College LONDON

- Below (a) is an illustration of the function $z = f(x, y) = x^2 + xy + y^2$.
- The red line highlighted in (a), shows the value of the function at the point $y = 1$.
- The red line is shown within the xz plane in (b), illustrating the partial derivative of z with respect to x: $\frac{\partial z}{\partial x} = 2x + y$.



(a) (b)

Differential Kinematics and the Jacobian Method for computing Inverse Kinematics

- Differential Kinematics gives the relationship between the joint velocities (how the joints change) and the corresponding end effector and its angular velocity

$$\nu_e = \begin{bmatrix} \dot{p}_e \\ \omega_e \end{bmatrix} = J\dot{q}$$

where:
- ν_e is the angular velocity vector of the end effector
- $\dot{p}_e$ is the position velocity (change in position) of the end effector
- ω_e is the angular velocity (change in rotation) of the end effector
- J is a square matrix representing the Jacobian—the partial derivatives of joint positions with respect to the angular velocity of the joints
- $\dot{q}$ is a vector of joint velocities

- For the contribution to the linear velocity (translation), we have:

$$\dot{p}_e = \sum_{i=1}^{n} JP_i \dot{q}_i$$

- We compute $\dot{p}_e$ as the sum of the terms $JP_i * \dot{q}_i$, where JP is the Jacobian matrix containing the partial derivatives of the position components with respect to $\dot{q}$ (joint velocities).

- Each term represents the contribution of the velocity of a single joint (i) to the end effector, as if all the other joints were stationary.

$$\dot{p}_e = [JP_1 JP_2 \ldots JP_n] \begin{bmatrix} \dot{q}_1 \\ \vdots \\ \dot{q}_2 \\ \dot{q}_n \end{bmatrix}$$

- For the contribution to the angular velocity (rotation), we have:

$$\dot{\omega_e} = \sum_{i=1}^{n} J\theta_i \dot{q}_i$$

where ω_i is the rotation (orientation) of the i-th joint.

- We compute $\dot{p_e}$ as the sum of the terms $J\theta_i * \dot{q}_i$, where $J\theta$ is the Jacobian matrix containing the partial derivatives of the orientation components (joint angles) with respect to $\dot{q}$.
- Each term represents the contribution of the velocity of a single joint angle (i) to the end effector, as if all the other joints were stationary.
- Thus:

$$\dot{\omega_e} = [J\theta_1 J\theta_2 \dots J\theta_n] \begin{bmatrix} \dot{q}_1 \\ \vdots \\ \dot{q}_2 \\ \dot{q}_n \end{bmatrix}$$

KING'S College LONDON

Jacobian, p4

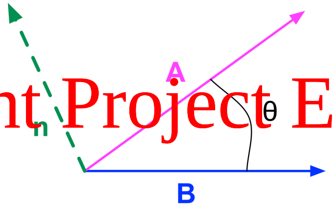
- We put these together in the Jacobian matrix:

$$J = \begin{bmatrix} \boxed{\text{position components}} \\ \\ \boxed{\text{orientation components}} \end{bmatrix} = \begin{bmatrix} JP \\ J\theta \end{bmatrix}$$

- and use it to compute the change in end effector position:

$$\nu_e = \begin{bmatrix} \dot{p}_e \\ \omega_e \end{bmatrix} = \begin{bmatrix} JP_1 & JP_2 & \dots & JP_n \\ J\theta_1 & J\theta_2 & \dots & J\theta_n \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \vdots \\ \dot{q}_n \end{bmatrix}$$

Quick review: Cross Product



(Imagine **A** and **B** lie in a plane that is flat on a table; then **n** is sticking straight up in the air, like a pencil perpendicular to the table top.)

- The cross product between two vectors, A and B, gives you a vector that is perpendicular to both of them, or the normal, n.
- The cross product can be calculated as:

$$A \times B = |A| \; |B| \; sin(\theta) \; n$$

where: A and B are our vectors,
θ is the angle between them,
and n is the unit vector perpendicular to both A and B

Quick review: Cross Product, p2

- The cross product, when A and B intersect at the origin $(0, 0)$ is computed as follows:

$$C = A \times B = \begin{bmatrix} A_x \\ A_y \\ A_z \end{bmatrix} \times \begin{bmatrix} B_x \\ B_y \\ B_z \end{bmatrix}$$

$$= \begin{bmatrix} 0 & -A_z & A_y \\ A_z & 0 & -A_x \\ -A_y & A_x & 0 \end{bmatrix} \begin{bmatrix} B_x \\ B_y \\ B_z \end{bmatrix}$$

$$= \begin{bmatrix} (0 * B_x) + (-A_z * B_y) + (A_y * B_z) \\ (A_z * B_x) + (0 * B_y) + (-A_x * B_z) \\ (-A_y * B_x) + (A_x * B_y) + (0 * B_z) \end{bmatrix}$$

$$= \begin{bmatrix} -A_z * B_y + A_y * B_z \\ A_z * B_x - A_x * B_z \\ -A_y * B_x + A_x * B_y \end{bmatrix}$$

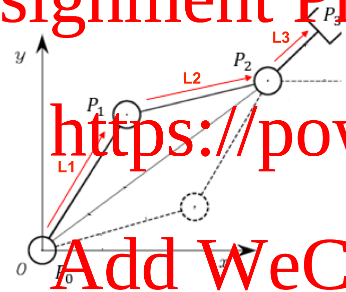
- For example, for a 3-link arm, J can be written as:

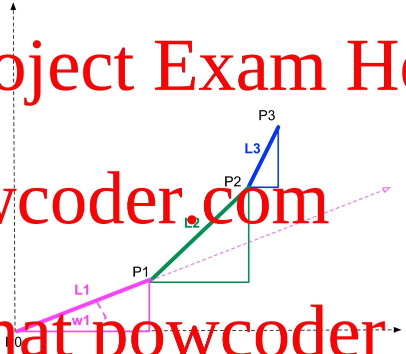
$$J = \begin{bmatrix} JP \\ JR \end{bmatrix} = \begin{bmatrix} \frac{\partial p_x}{\partial \theta_1} & \frac{\partial p_x}{\partial \theta_2} & \frac{\partial p_x}{\partial \theta_3} \\ \frac{\partial p_y}{\partial \theta_1} & \frac{\partial p_y}{\partial \theta_2} & \frac{\partial p_y}{\partial \theta_3} \\ \frac{\partial \omega}{\partial \theta_1} & \frac{\partial \omega}{\partial \theta_2} & \frac{\partial \omega}{\partial \theta_3} \end{bmatrix}$$

where:

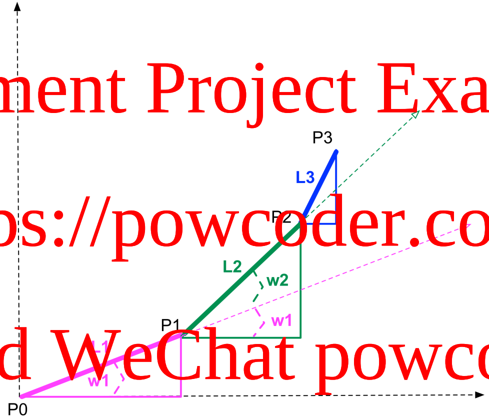
- (p_x, p_y) represent the position of the end effector
- ω represents the rotation of the end effector
- θ_i represents the angle at each of the 3 joints
- and

$$JP = \begin{bmatrix} \frac{\partial p_x}{\partial \theta_1} & \frac{\partial p_x}{\partial \theta_2} & \frac{\partial p_x}{\partial \theta_3} \\ \frac{\partial p_y}{\partial \theta_1} & \frac{\partial p_y}{\partial \theta_2} & \frac{\partial p_y}{\partial \theta_3} \end{bmatrix} \quad JR = \begin{bmatrix} \frac{\partial \omega}{\partial \theta_1} & \frac{\partial \omega}{\partial \theta_2} & \frac{\partial \omega}{\partial \theta_3} \end{bmatrix}$$

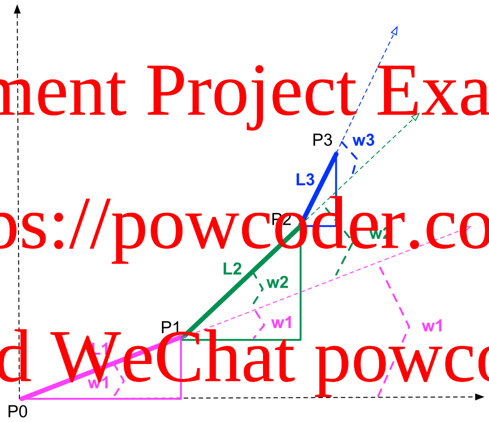




$$P0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad P1 = \begin{bmatrix} L_1 \ cos(\omega_1) \\ L_1 \ sin(\omega_1) \\ 0 \end{bmatrix}$$



$$P2 = \begin{bmatrix} L_1 \ cos(\omega_1) + L_2 \ cos(\omega_1 + \omega_2) \\ L_1 \ sin(\omega_1) + L_2 \ sin(\omega_1 + \omega_2) \\ 0 \end{bmatrix}$$



$$P3 = \begin{bmatrix} L_1 \ cos(\omega_1) + L_2 \ cos(\omega_1 + \omega_2) + L_3 \ cos(\omega_1 + \omega_2 + \omega_3) \\ L_1 \ sin(\omega_1) + L_2 \ sin(\omega_1 + \omega_2) + L_3 \ sin(\omega_1 + \omega_2 + \omega_3) \\ 0 \end{bmatrix}$$

- if we want to compute the position of the end effector, $P3$, we can use the Jacobian matrix to do that.
- Given:

$$P3_x = L_1 \; cos(\omega_1) + L_2 \; cos(\omega_1 + \omega_2) + L_3 \; cos(\omega_1 + \omega_2 + \omega_3)$$

the partial derivatives of $P3_x$ with respect to each ω_i are:

$$\frac{\partial P3_x}{\partial \omega_1} = -L_1 \; sin(\omega_1) - L_2 \; sin(\omega_1 + \omega_2) - L_3 \; sin(\omega_1 + \omega_2 + \omega_3)$$

$$\frac{\partial P3_x}{\partial \omega_2} = -L_2 \; sin(\omega_1 + \omega_2) - L_3 \; sin(\omega_1 + \omega_2 + \omega_3)$$

$$\frac{\partial P3_x}{\partial \omega_3} = -L_3 \; sin(\omega_1 + \omega_2 + \omega_3)$$

- and for $P3_y$:

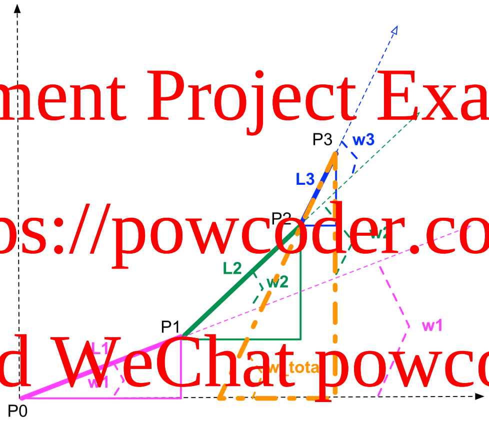
$$P3_y = L_1\ sin(\omega_1) + L_2\ sin(\omega_1 + \omega_2) + L_3\ sin(\omega_1 + \omega_2 + \omega_3)$$

the partial derivatives of $P3_y$ with respect to each ω_i are:

$$\frac{\partial P3_y}{\partial \omega_1} = L_1\ cos(\omega_1) + L_2\ cos(\omega_1 + \omega_2) + L_3\ cos(\omega_1 + \omega_2 + \omega_3)$$

$$\frac{\partial P3_y}{\partial \omega_2} = L_2\ cos(\omega_1 + \omega_2) + L_3\ cos(\omega_1 + \omega_2 + \omega_3)$$

$$\frac{\partial P3_y}{\partial \omega_3} = L_3\ cos(\omega_1 + \omega_2 + \omega_3)$$



- and for $\omega_{total} = \omega_1 + \omega_2 + \omega_3$:

$$\frac{\partial \omega_{total}}{\partial \omega_1} = 1 \qquad \frac{\partial \omega_{total}}{\partial \omega_2} = 1 \qquad \frac{\partial \omega_{total}}{\partial \omega_3} = 1$$

- We can put these all together into the Jacobian matrix as follows

$$J = \begin{bmatrix} \frac{\partial P3_x}{\partial \omega_1} & \frac{\partial P3_x}{\partial \omega_2} & \frac{\partial P3_x}{\partial \omega_3} \\ \frac{\partial P3_y}{\partial \omega_1} & \frac{\partial P3_y}{\partial \omega_2} & \frac{\partial P3_y}{\partial \omega_3} \\ \frac{\partial \omega_{total}}{\partial \omega_1} & \frac{\partial \omega_{total}}{\partial \omega_2} & \frac{\partial \omega_{total}}{\partial \omega_3} \end{bmatrix}$$

- Substitute for each of the partial derivatives the appropriate values from the previous slides
- Then we can use this matrix to determine how the end effector moves when each of the joint angles change—Differential Kinematics !!

- Remember where we started today:
  Differential Kinematics gives the relationship between the joint velocities (how the joints change) and the corresponding end effector and its angular velocity

$$\nu_e = \begin{bmatrix} \dot{p}_e \\ \omega_e \end{bmatrix} = J\dot{q} = \begin{bmatrix} JP_1 & JP_2 & \dots & JP_n \\ J\theta_1 & J\theta_2 & \dots & J\theta_n \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \vdots \\ \dot{q}_n \end{bmatrix}$$

where:

- ν_e is the angular velocity vector of the end effector
- $\dot{p}_e$ is the position velocity (change in position) of the end effector
- ω_e is the angular velocity (change in rotation) of the end effector
- $\dot{q}$ is a vector of joint velocities

- What if we know ν_e and we want to know $\dot{q}$?
- i.e., we want to determine what joint velocities are required to obtain a desired position and orientation of the end effector?

- This can be obtained by inverting the Jacobian matrix like this:

$$\dot{q} = J^{-1}\nu_e$$

- So if we know J:

  then we first compute its inverse: J^{-1}

  and then we multiply the inverse by the angular velocity vector of the end effector (ν_e)

$\rightarrow$ and that will tell us how much each joint angle changes

KING'S College LONDON

Jacobian: Example Code

```
float[] solveJacobian( px, py, pw, w1, w2, w3, ṗx, ṗy, ṗw ) {
  float new_angles[] = { 0.0, 0.0, 0.0 };
  float νₑ[] = { ṗx − px, ṗy − py, ṗw − pw }; // velocity of hₑ
  float J[][] = new float[3][3]; // Jacobian
  J[0][0] = −L1 ∗ sin(w1) − L2 ∗ sin(w1 + w2) − L3 ∗ sin(w1 + w2 + w3);
  J[0][1] = −L2 ∗ sin(w1 + w2) − L3 ∗ sin(w1 + w2 + w3);
  J[0][2] = −L3 ∗ sin(w1 + w2 + w3);
  J[1][0] = L1 ∗ cos(w1) + L2 ∗ cos(w1 + w2) + L3 ∗ cos(w1 + w2 + w3);
  J[1][1] = L2 ∗ cos(w1 + w2) + L3 ∗ cos(w1 + w2 + w3);
  J[1][2] = L3 ∗ cos(w1 + w2 + w3);
  J[2][0] = 1;
  J[2][1] = 1;
  J[2][2] = 1;
  float Jinv[][] = inv3( J ); // compute J⁻¹
  float q̇[] = multiply3x1( Jinv, νₑ ); // angle changes
  new_angles[0] = w1 + q̇[0];
  new_angles[1] = w2 + q̇[1];
  new_angles[2] = w3 + q̇[2];
  return( new_angles );
} // end of solveJacobian()
```

Summary

- That's all!
- Okay it's a bit complicated
- But you should understand it better after completing Coursework 2 part 2

- Coursework 2 is due in 2 weeks (27th November)
- Look at starter and helper code uploaded on KEATS

- sum rule

$$\frac{d}{dx}(u+v) = \frac{du}{dx} + \frac{dv}{dx}$$

- product rule:

$$\frac{d}{dx}(u*v) = \frac{du}{dx}v + u\frac{dv}{dx}$$

- constant multiple rule:

$$\frac{d}{dx}A*u = A*\frac{du}{dx}$$

- power rule:

$$\frac{d}{dx}(x^n) = n * x^{(x-1)}$$

- power of a function rule:

$$\frac{d}{dx}u^n = n * u^{n-1}\frac{du}{dx}$$

- quotient rule:

$$\frac{d}{dx}\left(\frac{u}{v}\right) = \frac{\frac{du}{dx}v - u\frac{dv}{dx}}{v^2}$$

- reciprocal rule:

$$\frac{d}{dx}\left(\frac{1}{u}\right) = -\frac{1}{u^2}\frac{du}{dx}$$

- chain rule:

$$\frac{dy}{dx} = \frac{dy}{du} \cdot \frac{du}{dx}$$

if y is a function of u and u is a function of x

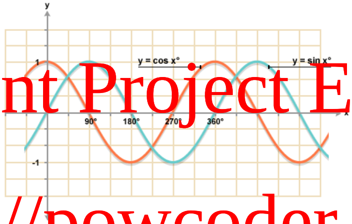
- quadratic function rule:

$$\frac{d(ax^2 + bx + c)}{dx} = 2ax + b$$

- linear function rule:

$$\frac{d(bx + c)}{dx} = b$$

$$\frac{d(c)}{dx} = 0$$



- The change in value of $sin(x)$ with respect to x is $cos(x)$:

$$\frac{d\ sin(x)}{dx} = cos(x)$$

- The change in value of $cos(x)$ with respect to x is $-sin(x)$:

$$\frac{d\ cos(x)}{dx} = -sin(x)$$

SNS Introduction to Autonomous Mobile Robots, by Roland Siegwart, Illah R. Nourbakhsh and Davide Scaramuzza (2011), MIT Press, chapter 3.

MW Calculus, by Jerrold Marsden and Alan Weinstein (1980), The Benjamin/Cummings Publishing Company, Inc.