Introduction to Robotics
Week 7: Inverse Kinematics
**Inverse kinematics**

Forward kinematics: compute the end-effector position (as an element of $SE(3)$) from joint angles $\theta_i$: compute the function

$$T: \text{joint space} \to SE(3): \theta \mapsto T(\theta)$$

Inverse kinematics: compute the (possible) joint angles from the position of the end-effector: compute the function

$$T^{-1}: SE(3) \to \text{joint space}: X \mapsto \theta.$$  

The inverse kinematics function is often *multi-valued.*
The two argument arctan function: \texttt{atan2}

Returns the angle between $x$-axis and vector $(x, y)$ in the plane. Unlike \texttt{atan}, which is valued in $(-\pi/2, \pi/2]$, \texttt{atan2} is valued in $(-\pi, \pi]$. It is a default trig. function in most programming languages.

\[
\texttt{atan2}(y, x) = \begin{cases} 
\text{atan}(y/x) & \text{if } x > 0 \\
\text{atan}(y/x) + \pi & \text{if } x < 0 \text{ and } y \geq 0 \\
\text{atan}(y/x) - \pi & \text{if } x < 0 \text{ and } y < 0 \\
\pi/2 & \text{if } x = 0 \text{ and } y > 0 \\
-\pi/2 & \text{if } x = 0 \text{ and } y < 0 \\
\text{undefined} & \text{if } x = 0 \text{ and } y = 0
\end{cases}
\]

\texttt{Matlab: \texttt{atan2(y,x)}}

\texttt{Discontinuity is forced on the user}

\texttt{NumPy: numpy.arctan2}
Analytic inverse kinematics

Recall: Law of cosines

\[ c^2 = a^2 + b^2 - 2ab \cos(\gamma), \]

where \( a, b, c \) are the lengths of the edges of the triangle, and \( \alpha, \beta, \gamma \) the angles opposite \( a, b \) and \( c \) respectively. We have \( L_1^2 + L_2^2 - 2L_1 L_2 \cos \beta = x^2 + y^2 \).

It follows

\[ \beta = \cos^{-1} \left( \frac{L_1^2 + L_2^2 - x^2 - y^2}{2L_1 L_2} \right). \]

Similarly, \( \alpha = \cos^{-1} \left( \frac{L_1^2 - L_2^2 + x^2 + y^2}{2L_1 \sqrt{x^2 + y^2}} \right). \)

Using \( \text{atan2} \) function, we get \( \gamma = \text{atan2}(y, x) \). The two possible solutions are

\[ \theta_1 = \gamma - \alpha, \theta_2 = \pi - \beta \text{ and } \theta_1 = \gamma + \alpha, \theta_2 = \beta - \pi \]

If \( x^2 + y^2 \notin [L_1 - L_2, L_1 + L_2] \), then no solutions exist.

\[ \theta_1 = \gamma + \alpha, \theta_2 = -\beta \]
Analytic inverse kinematics

In the 2R robot example, there were 2 DOFs for the end-effector, and 2 joint angles. This implied that there was a finite number of solutions. If there are more joint angles than DOFs of the end-effector, there may be an infinite number of solutions. We will mostly look at cases where #DOFs of end-effector = # joint angles. We thus assume in general

\[ T(\theta) = e^{[\mathcal{A}_1] \theta_1} \cdots e^{[\mathcal{A}_6] \theta_6} M \]

and we are given an end-effector pose \( X \in SE(3) \). We need to find \( T^{-1}(X) \).
Analytic inverse kinematics: Euler angles

Special case: **Euler angles**.
They are useful in evaluating inverse kinematic maps analytically.
The ZYX Euler angles can be used to represent an arbitrary rotation in \( \mathbb{R}^3 \) as follows:

\[
R(\alpha, \beta, \gamma) = \text{Rot}(\hat{z}, \alpha) \text{Rot}(\hat{y}, \beta) \text{Rot}(\hat{x}, \gamma)
\]

with \( \alpha, \gamma \in (-\pi, \pi) \) and \( \beta \in [-\pi/2, \pi/2) \) and

\[
\begin{align*}
\text{Rot}(\hat{z}, \alpha) &= \begin{bmatrix}
\cos \alpha & -\sin \alpha & 0 \\
\sin \alpha & \cos \alpha & 0 \\
0 & 0 & 1
\end{bmatrix}, \\
\text{Rot}(\hat{y}, \beta) &= \begin{bmatrix}
\cos \beta & 0 & \sin \beta \\
0 & 1 & 0 \\
-\sin \beta & 0 & \cos \beta
\end{bmatrix}, \\
\text{Rot}(\hat{x}, \gamma) &= \begin{bmatrix}
1 & 0 & 0 \\
0 & \cos \gamma & -\sin \gamma \\
0 & \sin \gamma & \cos \gamma
\end{bmatrix}.
\end{align*}
\]
Analytic inverse kinematics: Euler angles

Inverse kinematics ⇒ inverse problem: given $R \in SO(3)$, can we always find $\alpha, \beta, \gamma$ so that $R(\alpha, \beta, \gamma) = R$?
The answer is yes.
Explicitly, we have

$$R(\alpha, \beta, \gamma) = \begin{bmatrix} c_\alpha c_\beta & c_\alpha s_\beta s_\gamma - s_\alpha c_\gamma & c_\alpha s_\beta c_\gamma + s_\alpha s_\gamma \\ s_\alpha c_\beta & s_\alpha s_\beta s_\gamma + c_\alpha c_\gamma & s_\alpha s_\beta c_\gamma - c_\alpha s_\gamma \\ -s_\beta & c_\beta s_\gamma & c_\beta c_\gamma \end{bmatrix} = \begin{pmatrix} c_\alpha & c_\beta & c_\gamma \\ f_1 & f_2 & f_3 \\ f_4 & f_5 & f_6 \end{pmatrix}$$

$\sin \beta = -f_3$
Let $r_{ij}$ be the $ij^{th}$ entry of $R$.
We can first look at $r_{31}$. If $r_{31} \neq \pm 1$ set

\[
\beta = \atan2(-r_{31}, \sqrt{r_{11}^2 + r_{21}^2}), \quad \alpha = \atan2(r_{21}, r_{11}) \quad \text{and} \quad \gamma = \atan2(r_{32}, r_{33}).
\]

If $r_{31} = -1$, then $\beta = \pi/2$. There exists an infinite number of solutions for $\alpha$ and $\gamma$. One such solution is $\alpha = 0, \gamma = \atan2(r_{12}, r_{22})$.

If $r_{31} = 1$, then $\beta = -\pi/2$. There exists an infinite number of solutions for $\alpha$ and $\gamma$. One such solution is $\alpha = 0, \gamma = -\atan2(r_{12}, r_{22})$. 

\[
R(\alpha, \beta, \gamma) = \begin{bmatrix}
c_\alpha c_\beta & c_\alpha s_\beta s_\gamma - s_\alpha c_\gamma & c_\alpha s_\beta c_\gamma + s_\alpha s_\gamma \\
s_\alpha c_\beta & -s_\alpha s_\beta s_\gamma + c_\alpha c_\gamma & s_\alpha s_\beta c_\gamma - c_\alpha s_\gamma \\
-s_\beta & c_\beta s_\gamma & c_\beta c_\gamma
\end{bmatrix}
\]
Analytic inverse kinematics: 6R Puma arm

PUMA stands for Programmable Universal Machine for Assembly Industrial robot arm, developed for car manufacturing in late 1970's. Still widely used today.

Zero position:
Two shoulder joint intersect orthogonally at a common point. Joint axis 1 is aligned with $\hat{z}_0$, joint axis 2 with $\hat{y}_0$.
Joint axis 3 (elbow) in $\hat{x}_0, \hat{y}_0$ plane and parallel with joint axis 2
Joint 4-5-6 form a wrist. They intersect orthogonally at a common point and are aligned to the $\hat{z}_0, \hat{y}_0$ and $\hat{x}_0$ directions respectively.
Analytic inverse kinematics: 6R Puma arm

The inverse kinematics problem can be split into inverse orientation and position problems. (Not true for all mechanisms!)

Let \( p = (p_x, p_y, p_z) \) be position of the wrist center.
Assume \( (p_x, p_y) \neq (0, 0) \) We have that
\[ \theta_1 = \text{atan2}(p_y, p_x). \]
When \( (p_x, p_y) = (0, 0) \), we are in a singular configuration, there are infinitely many solutions for \( \theta_1 \).
Finding $\theta_2$ and $\theta_3$ reduces to IK for planar 2R robot. Applying what we had derived before to this case, we get

$$\cos \theta_3 = \frac{r^2 + p_z^2 - a_2^2 - a_3^2}{2a_2a_3} = D$$

We then have $\theta_3 = \text{atan2}(\sqrt{1-D^2}, D)$

We obtain for $\theta_2$

$$\theta_2 = \text{atan2}(p_z, r) - \text{atan2}(a_3 s_3, a_2 + a_3 c_3)$$
Recall that $X = e^{[\mathcal{S}_1] \theta_1} \cdots e^{[\mathcal{S}_6] \theta_6} M$, and we know $M$ and $X$ and have just figured out what $\theta_1, \theta_2, \theta_3$ are. We thus need to solve

$$e^{[\mathcal{S}_4] \theta_4} e^{[\mathcal{S}_5] \theta_5} e^{[\mathcal{S}_6] \theta_6} = e^{[-\mathcal{S}_3] \theta_3} e^{-[\mathcal{S}_2] \theta_2} e^{-[\mathcal{S}_1] \theta_1} XM^{-1},$$

where $\omega_4 = (0, 0, 1), \omega_5 = (0, 1, 0)$ and $\omega_6 = (1, 0, 0)$. Denote by $R$ the rotation component of $e^{[\mathcal{S}_3] \theta_3} e^{-[\mathcal{S}_2] \theta_2} e^{-[\mathcal{S}_1] \theta_1} XM^{-1}$. We thus need to find $\theta_4, \theta_5, \theta_6$ so that

$$\text{Rot}(\hat{z}, \theta_4) \text{Rot}(\hat{y}, \theta_5) \text{Rot}(\hat{x}, \theta_6) = R.$$ 

This is exactly the ZYX Euler angles problem we have solved.
Analytic inverse kinematics: 6R Puma arm

The real-life design involves an offset: a "shoulder". The analysis in this case is not much different.
**Numeric inverse kinematics**

Forward kinematics: compute the end-effector position (as an element of $SE(3)$) from joint angles $\theta_i$:
compute the function

$$T : \text{joint space} \rightarrow SE(3) : \theta \mapsto T(\theta)$$

Inverse kinematics: compute the (possible) joint angles from the position of the end-effector:
compute the function

$$T^{-1} : SE(3) \rightarrow \text{joint space} : X \mapsto \theta.$$

The inverse kinematics function is often *multi-valued*. When analytic solutions are are or impossible to come by, we can solve $T(\theta) - X = 0$ for $\theta$ numerically. We write the previous equation as $f(\theta) - x = 0$, where $x \in \mathbb{R}^m$ and $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$.

Iterations: Newton method.

$$T : C(\theta) \rightarrow W$$

DIY numerical procedures

**General problem:**

$$T : \mathbb{R}^n \rightarrow \mathbb{R}^m \Rightarrow x$$

$$T(\theta) = x$$

$$T(\theta) - x = 0$$