The Kinematics and Dynamics of a Humanoid Robot Arm for the FIRA Robot Games

**Kinematics**

Kinematics is the study of motion with relation to just the time variant properties of a system, without regard to the forces which are required to produce that motion.

For the robot arm, it is the analytical relationship between the joint positions and the end-effector position and orientation.

**The Ball's Trajectory**

After the ball is thrown, it can be modeled as a particle acting under gravity, using projectile trajectory maths and the equations of motion. This finds the throwing conditions in Cartesian coordinates.

**Inverse Kinematics**

Inverse kinematics finds the joint angles and positions required to place the end effector in the desired throwing position ($x_y$).

- Using the Cosine Law: $a^2 = b^2 + c^2 - 2bc \cdot \cos\theta$
- Using the Sine Law: $\frac{a}{\sin A} = \frac{b}{\sin B} = \frac{c}{\sin C}$

**Forward Kinematics**

Forward kinematics calculates the linear position and orientation of the end-effector from the joint positions. It relates the angular positions to the Cartesian positions:

$$(x, y) = ((x_1 + x_2 \cdot \sin\theta_2), (y_1 + x_2 \cdot \cos\theta_2), (y_1 - x_2 \cdot \sin\theta_2 - \frac{x_2}{2} \cdot \cos\theta_2))$$

The joint's angular speed from the Cartesian speed is found using the Jacobian mathematical method.

**Dynamics**

Dynamics is the relationship between the motion of an object and the motion’s cause, where mass and thus forces, as well as time-variant properties are studied.

The dynamics of a robotic system are made up of inertia effects and the work the system is doing.

The Generic Dynamic Equation:

\[ M(a) \cdot \ddot{a} + \tau = \tau_D \]

- \( M(a) \) is the vector of the robot's joint variables
- \( \tau \) is the actuator torque/force
- \( \tau_D \) is the sum of forces and moment (net torque)
- \( \ddot{a} \) is the Coriolis and centrifugal acceleration
- \( \dot{a} \) is the gravity vector

The dynamic matrices were derived using the Lagrange-Taylor method.

**Trajectory Planning**

Trajectory planning is a method of making a manipulator move from one position to another in a smooth and controlled manner, by giving each joint a smooth function of time to follow. The trajectory is described by several points through which a polynomial is fitted.

**Nonlinear Controllers**

The robot arm needs a controller to relate the desired trajectory to the motors' torque. A controller is effectively the dynamical relationship between the system's output and its actuation.

A practical mechanical system usually has nonlinear dynamics due to the damping and friction effects of the natural environment. A nonlinear controller will linearize the dynamic equation.

**Joint-Based Feedback Linearization Control**

\[ u_j = u_j(\dot{x}) - \frac{\partial x_j}{\partial x} \cdot \dot{x} + \frac{\partial x_j}{\partial \dot{x}} \cdot \ddot{x} + \frac{\partial x_j}{\partial \theta} \cdot \tau_j + \frac{\partial x_j}{\partial \dot{\theta}} \cdot \dot{\tau}_j \]

**Cartesian Feedback Linearization Control**

\[ u_c = u_c(\dot{x}) - \frac{\partial x_c}{\partial x} \cdot \dot{x} + \frac{\partial x_c}{\partial \dot{x}} \cdot \ddot{x} + \frac{\partial x_c}{\partial \theta} \cdot \tau_c + \frac{\partial x_c}{\partial \dot{\theta}} \cdot \dot{\tau}_c \]

---

By: Claire Tobin  
Supervisor: Dr. Guido Herrmann

University of Bristol  
Department of Mechanical Engineering