Simulation of Humanoid Robot

Introduction :

1.Project Overview:

This project focuses on importing a robot's Unified Robot Description Format (URDF) file into MATLAB and enabling the robot to move on the ground. The objective is to simulate the robot's motion in a controlled environment, providing a platform to test and analyze various movement algorithms. By achieving this, we aim to bridge the gap between theoretical robot design and practical implementation in a simulation environment.

2.Background and Motivation

Robotic systems are widely used in industries like manufacturing and healthcare. Simulating these systems helps in understanding their behavior, testing designs, and saving development costs. URDF files are a common standard for describing robot models.

MATLAB is a powerful tool for simulation and visualization, making it a great choice for this project. By importing URDF files into MATLAB, developers can refine robot designs and test movement algorithms easily. This project aims to simplify simulation workflows so researchers and engineers can focus on innovation instead of setup challenges.

3.MATLAB Software

This project utilizes MATLAB 2024b, along with the following libraries and toolboxes:

Toolboxes:

- **Multibody Simscape Toolbox**
    - For simulating the robot's physical dynamics.
    - Used to model complex mechanical systems with multiple interconnected bodies.
- **Robotics System Toolbox**
    - For importing and visualizing URDF files.
    - Includes tools for kinematic and dynamic analysis of robot models.

Libraries:

- **Rigid Body and Rigid Body Transforms**
  - To define and manage the robot’s structure and coordinate systems.
- **Spatial Contact Force Block**
  - Used for modeling physical interactions, such as contact forces between the robot and the ground.
- **Signal Editor**
  - Provides a platform to design, edit, and test control input signals.

Problem Statement and Objectives:

1.Problem Statement:

This project simulates the motion of a robot described by a Unified Robot Description Format (URDF) file in MATLAB. The goal is to enable realistic motion and interaction modeling, providing a platform to test and refine movement algorithms in a virtual environment

2.Objectives:

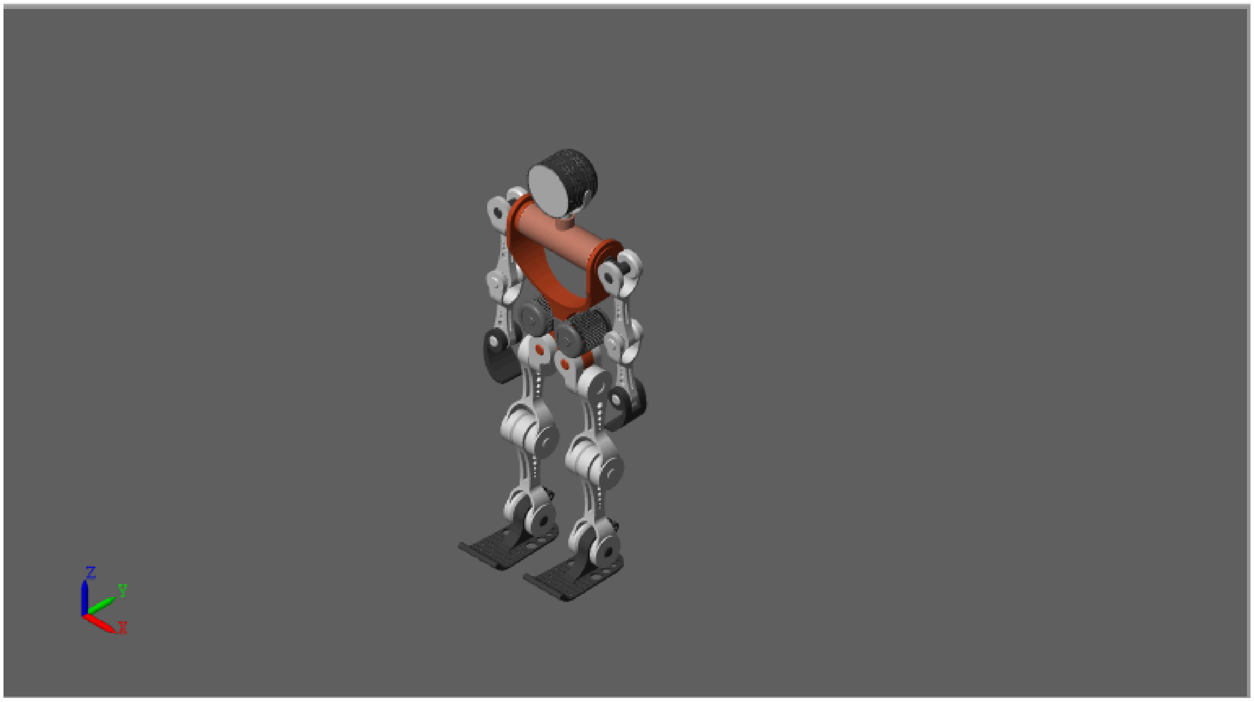
1. Import and visualize a robot's URDF file in MATLAB.
2. Simulate the robot's motion on a flat surface.
3. Implement algorithms for basic kinematics and dynamics.
4. Provide a framework for testing control signals and analyzing movement.

Methodology

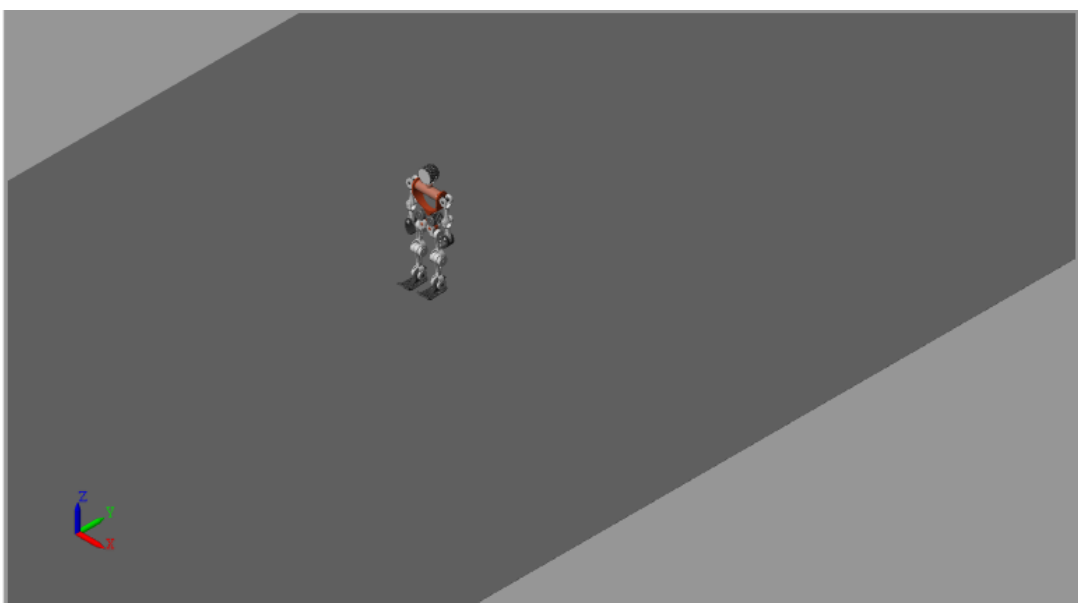
1.Approach:

The objective of this project is to model and simulate a robotic system with leg movement and interaction with the ground. The following steps outline the approach taken to achieve this:

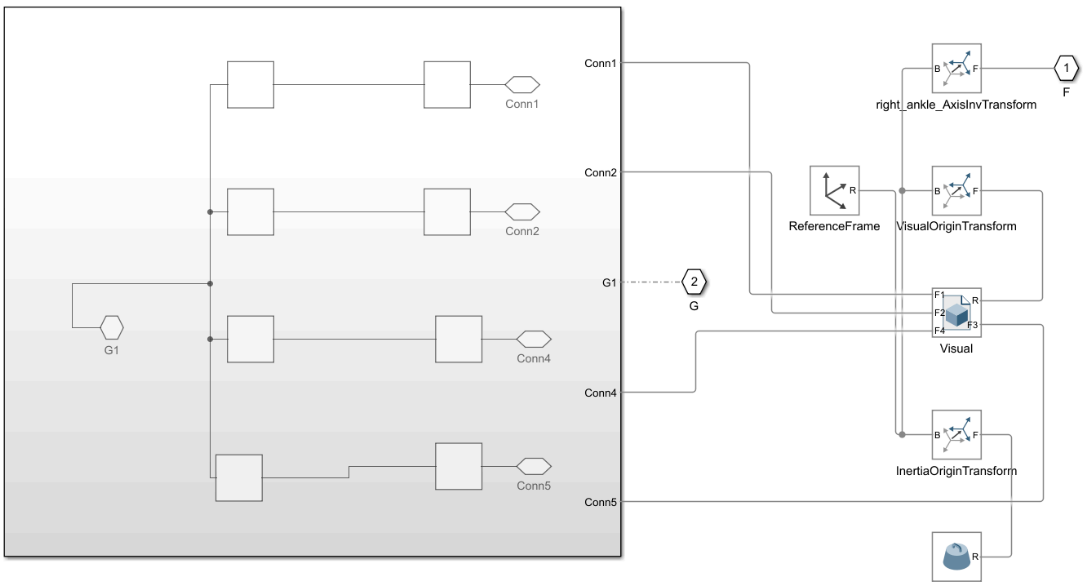
1. **Importing URDF Files**: The simulation begins with importing the URDF (Universal Robot Description Format) files from the MATLAB example. These files define the robot's structure, including the links, joints, and their properties. The robot's physical configuration is set up based on the imported URDF model.



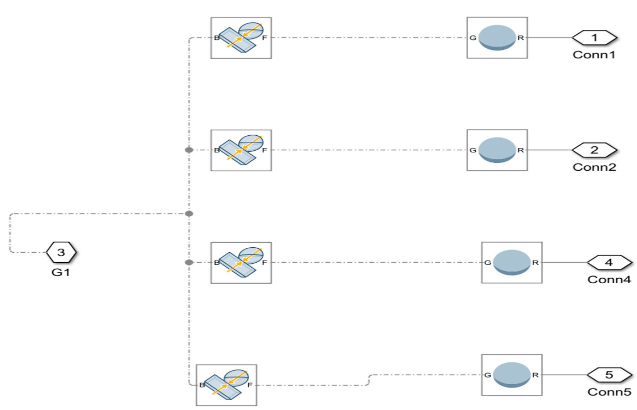
2. **Adding the Ground Frame**: To simulate the interaction between the robot and the environment, the ground frame is introduced using a Rigid Body and a Rigid Transform block. The rigid body block represents the fixed ground frame, and the rigid transform block is used to define the spatial relationship between the robot and the ground. This enables the robot to be correctly oriented and positioned in the simulation space.



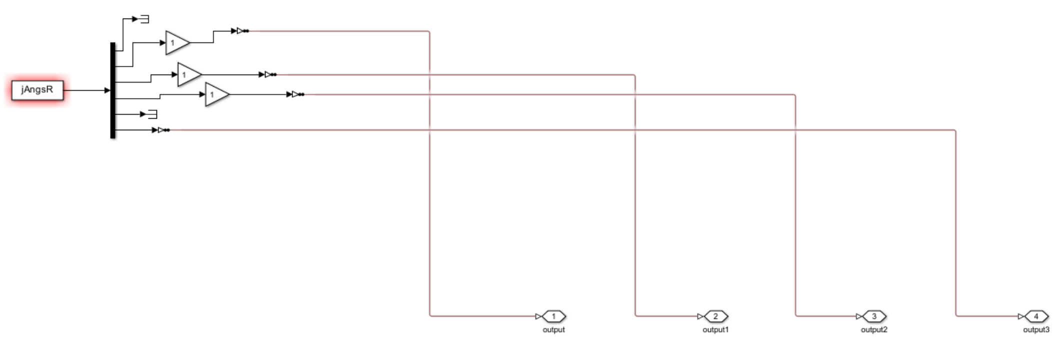
3. **Providing Contact for the Feet**: In order to model the interaction between the robot's feet and the ground, contact proxies are placed at the bottom of the robot's feet. These rigid bodies provide contact with the ground surface through the spatial contact force block. A Spatial Contact Force block is employed to simulate the forces generated during contact, allowing for realistic ground interaction forces to be applied to the robot during its movement.



We have established the contact proxies' positions at the bottom of the foot, attaching them to the subsystem block. This subsystem consists of spherical rigid bodies, each serving as a contact proxy with a radius of 0.1 meters. These contact proxies are connected to a spatial force block, which provides the necessary connection to the ground frame.

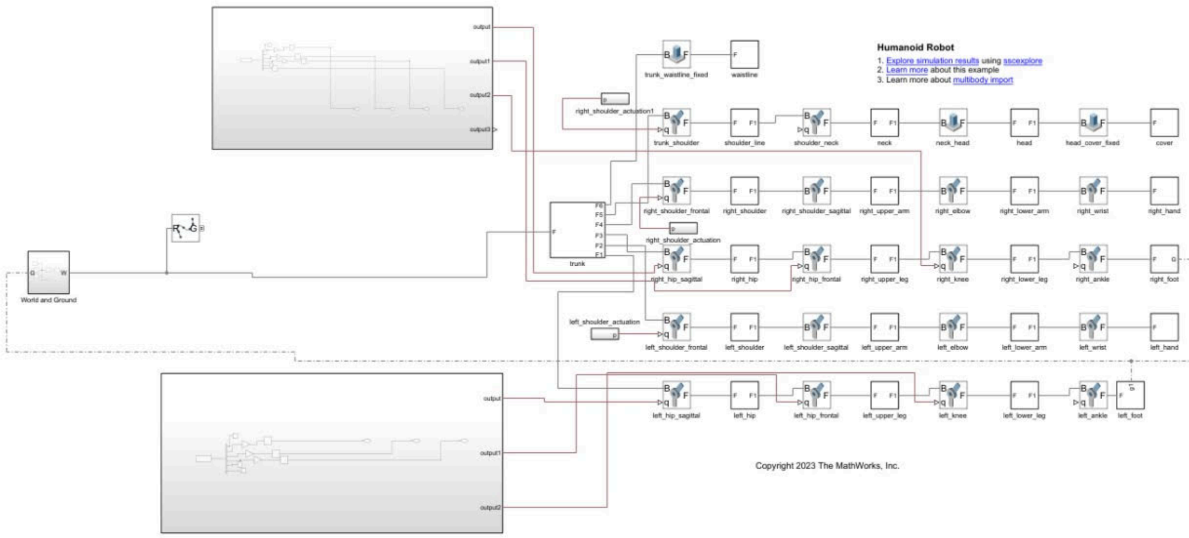


4. **Controlling Leg Movement**: For leg movement, we define two separate motion matrices: one for the right leg and one for the left leg. These matrices contain the required motion angles for key joints such as the hip, knee, and ankle.
   A Demux block is used to separate these angle matrices and direct the individual joint angles to the corresponding joints on the left and right sides of the robot. This allows for the coordinated movement of the robot's legs, where each leg's motion is controlled independently while maintaining synchronization.



- The input signal enters through the **"jAngsR"** block(matrix).
- The signal is distributed through a **signal splitter(Demux)** into multiple branches.
- Each branch passes through a **gain block** with a gain of 1, ensuring no amplification or attenuation occurs.
- The signals are routed to four distinct outputs (**output, output1, output2, and output3**) ,that are hip,knee,ankle angles

Similarly it is done for the left part of the body.



The above is the complete simscape block of the simulation

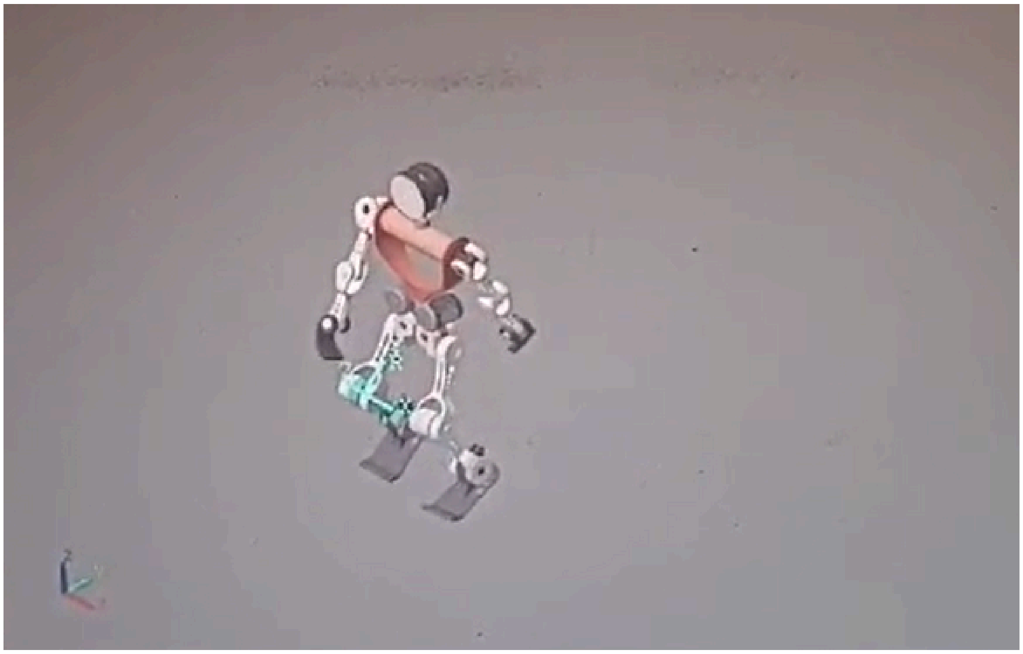
2.Data Collection :

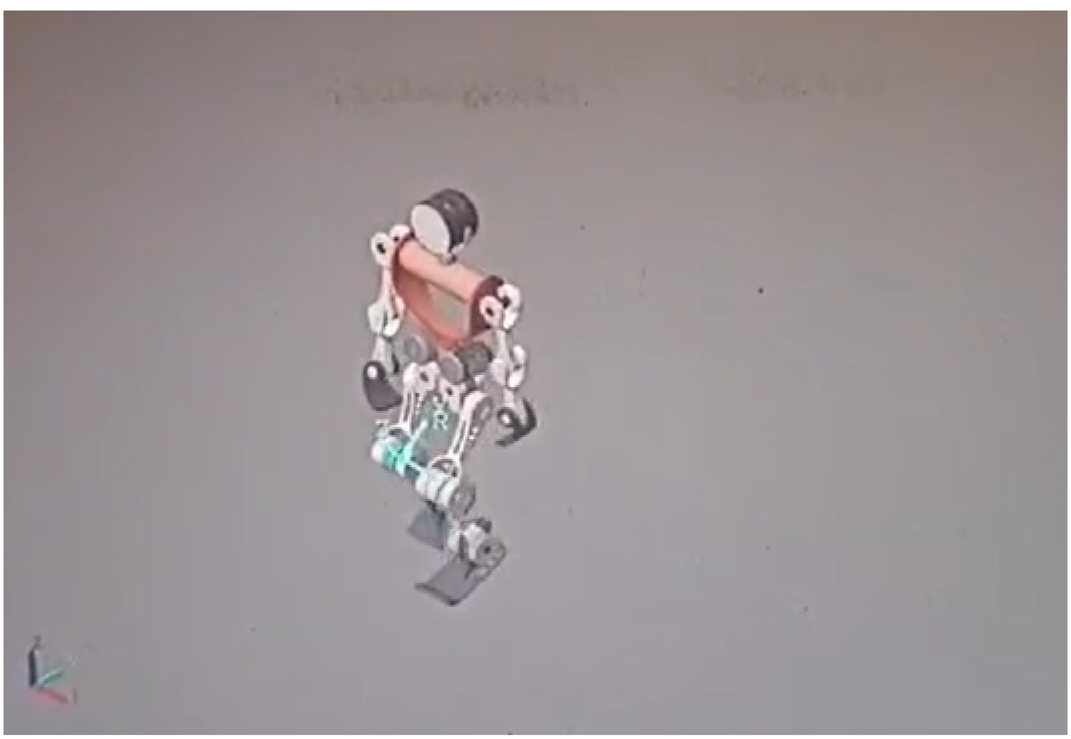
We obtained the joint angle matrices with the help of the **MATLAB Central** community and adjusted the values according to the model's specifications. The trial angles were then tested as per the model requirements.

<u>Result:</u>

The figure above illustrates a **simulation of a robotic system** .The key observations and outcomes are as follows:

1. **System Description**:
    - The system appears to be a **bipedal robot model** designed for motion simulation.
    - The design includes distinct components such as **joints, links, and actuators**, resembling a humanoid robotic system.
2. **Simulation Motion**:
    - The robot is shown in a specific **pose**, suggesting a dynamic walking simulation.
    - The mechanical parts demonstrate **realistic movement** and alignment, verifying the accuracy of the design.





Limitations

While the system performs as intended in this simulation, a few limitations should be considered:

1. **Scalability:** The current design uses simple gain blocks with a gain of 1. In more complex scenarios requiring dynamic signal amplification or attenuation, additional components or control mechanisms may be required.
2. **Signal Integrity:** If this system were implemented in hardware, factors like noise, latency, and signal degradation could affect performance.
3. **Complexity**: The architecture is currently basic. Expanding it to include feedback loops or more advanced processing might complicate the design.

Summary:

The project successfully simulated the motion of a humanoid robot described by a Unified Robot Description Format (URDF) file in MATLAB. By importing the robot model, defining its interaction with the ground, and implementing control mechanisms for joint movement, the simulation provided a realistic environment to test motion algorithms.

Future Scope

The project can be extended in several ways to enhance its capabilities:

1. **Integration with Deep Learning Models:** Advanced deep learning algorithms can be incorporated for improved motion control, gait optimization, and autonomous decision-making.
2. **Enhanced Mobility:** Future iterations can focus on dynamic movement capabilities such as running, jumping, or climbing, expanding the robot's usability in real-world applications.
3. **Feedback Mechanisms:** Introducing feedback control systems will enable adaptive behavior and improved stability in various terrains.
4. **Hardware Implementation:** Bridging the gap between simulation and hardware by deploying the model on a physical humanoid robot for real-world testing.
5. **Environment Interaction:** Including more complex environmental features, such as uneven surfaces or obstacles, for comprehensive interaction modeling.