Multibody Dynamic Simulation And Load Analysis of A Robotic Manipulator Using Pybullet

V S Sagarika[1] , Thanmayee R[2], Prof. Dr. N.Saravanan[3], Prof. X. Anitha Sarafin[4]
[1, 2, 3, 4]Dept of Computer science and Engineering
[1, 2, 3, 4] Misrimal Navajee Munoth Jain Engineering College, Chennai, TamilNadu – 600097

Abstract- *Robotic manipulators are extensively used in industrial automation for precision-driven tasks such as object handling, assembly, and material transfer. Accurate motion and load analysis are essential to ensure operational stability and prevent mechanical failure. This paper presents a multibody dynamic simulation framework for evaluating the motion behaviour and load-bearing capacity of a robotic manipulator using PyBullet. A multi-degree-of-freedom robotic arm model was developed using URDF with realistic mass and inertia properties. The system integrates forward kinematics, inverse kinematics, and Newton–Euler dynamic modelling to analyse joint torque under varying payload conditions. In addition, the framework incorporates a digital twin–based simulation environment with a graphical user interface for parameter input and control, along with AI-based object detection using the YOLO model for identifying target objects within the simulation. Real-time simulation data, including joint angles, torque values, positional parameters, and end-effector trajectories, were logged and analyzed for performance evaluation. Experimental results demonstrate the relationship between payload mass and torque requirements, identifying safe operational thresholds. The developed system was successfully implemented and tested, demonstrating that physics-based digital twin simulation provides an efficient and reliable approach for analysing robotic manipulator dynamics before hardware implementation.*

Keywords: Robotic Manipulator, Multibody Dynamics, PyBullet, Digital Twin, Object Detection, Load Analysis, Torque Evaluation, Simulation.

I. INTRODUCTION

Robotic manipulators play a vital role in modern industrial and manufacturing environments, including applications such as automated assembly, packaging, welding, material handling, precision machining, and pick-and-place operations. These systems are designed to perform repetitive and high-precision tasks with speed, consistency, and reliability. As industrial automation continues to evolve toward higher efficiency and reduced human intervention, robotic manipulators have become fundamental components of smart factories and Industry 4.0 ecosystems.

One of the most critical challenges in robotic system design is ensuring that the manipulator can safely and efficiently handle assigned payloads without exceeding mechanical or structural limits. When a robotic arm lifts an object, each joint experiences dynamic torque due to link mass, payload weight, inertia effects, and gravitational forces. If the required torque exceeds the actuator’s rated capacity, it may result in instability, energy inefficiency, accelerated wear, or mechanical failure. Therefore, accurate torque estimation and load feasibility analysis are essential during the design and validation stages of robotic manipulators

Traditionally, load testing and performance validation require physical prototypes and repeated experimental trials, which are expensive, time-consuming, and potentially unsafe under overload conditions. Simulation-based validation has therefore emerged as a cost-effective and efficient approach for analyzing robotic performance prior to real-world implementation. Physics-based simulation platforms such as PyBullet enable realistic modeling of rigid-body dynamics, including gravity, collision detection, joint constraints, and real-time torque computation.

Multibody dynamics, based on the classical mechanics of Newton and Euler, provides a mathematical framework for analyzing interconnected rigid bodies subjected to forces and torques. In robotic manipulators, each link and joint forms part of a dynamic chain where motion in one segment influences the entire system. By applying Newton–Euler equations, it becomes possible to compute the torque required at each joint for a given motion trajectory and payload condition.

In addition to dynamic modeling, kinematic analysis plays an important role in robotic motion evaluation. Forward kinematics determines the end-effector position based on joint configurations, while inverse kinematics computes the joint angles required to reach a desired target position.

Integrating kinematic modeling with multibody dynamics enables a more accurate evaluation of robotic motion behavior and load feasibility.

This research presents a simulation-based framework for analyzing the motion behavior and load capacity of a robotic manipulator using PyBullet. A URDF-based multi-degree-of-freedom robotic arm model is developed with realistic mass and inertia parameters, and lifting operations are simulated under different payload conditions while monitoring joint torque values. In addition, the system incorporates a digital twin–based environment with a graphical user interface for parameter input and control. An AI-based object detection module using the YOLO model is also integrated to detect objects and provide target coordinates for robotic manipulation. Simulation data such as joint angles, torque values, and end-effector trajectories are logged and analyzed for performance evaluation. The developed system was successfully implemented and demonstrates that physics-based digital twin simulation provides an efficient and reliable approach for evaluating robotic manipulator performance before hardware implementation.

II. LITERATURE REVIEW

Digital Twin (DT) technology has become a fundamental component of Industry 4.0, enabling real-time synchronization between physical systems and their virtual counterparts.

[1] Boschetti and Sinico present the design of a high-fidelity digital twin of a six-degree-of-freedom industrial robot using Simscape Multibody. Their work emphasizes accurate dynamic modeling by incorporating joint friction, reduction gears, and motor dynamics, demonstrating the importance of detailed physical modeling for reliable simulation and control validation. This study highlights the growing need for digital twins capable of supporting advanced control strategies such as computed torque control. Beyond modeling accuracy, intelligent control integration has gained attention in DT-based robotic systems.

[2] Cen et al. propose a Digital Twin–empowered robotic arm system integrating Proximal Policy Optimization (PPO) and Fuzzy PID control. Their approach addresses virtual–real mapping errors and physical execution inaccuracies by combining deep reinforcement learning with classical control techniques. The study demonstrates that DT frameworks can enhance adaptability, fault tolerance, and precision in robotic manipulation tasks.

[3] From a broader manufacturing perspective, Qin et al. provide a comprehensive review of Robot Digital Twin (RDT) systems in Industry 4.0 environments. They propose a four-layer architecture consisting of physical, connection, virtual, and supporting layers, integrating technologies such as IIoT, cloud computing, artificial intelligence, and advanced robotics. Their review identifies enabling technologies, application domains, current research trends, and key challenges, establishing a structured framework for future RDT development. Real-time synchronization between physical and virtual environments remains a major research focus.

[4] Huang et al. introduce SyncTwin, a fast digital twin framework for safe robotic grasping. Their system combines efficient 3D scene reconstruction, object tracking, and real-to-sim synchronization to support collision-free motion planning under dynamic and partially occluded conditions. The study demonstrates how continuous synchronization improves safety and execution accuracy in real-world robotic manipulation. In the context of engineering education, digital twins are increasingly used to extend laboratory capabilities.

[5] The study presented in describes the digital twin design of a Mitsubishi RV-2AJ 5-DOF industrial robot using photogrammetry and reverse engineering techniques. The virtual model was validated using forward kinematics in MATLAB and Tool Center Point (TCP) trajectory comparison, ensuring functional equivalence between the physical robot and its digital counterpart. This work highlights the role of DT technology in modern engineering education, especially in enabling virtual laboratories and Industry 4.0 training environments. Overall, existing literature demonstrates that digital twin technology enhances robotic system modeling, intelligent control integration, real-time synchronization, and educational applications. However, challenges remain in achieving high-fidelity dynamic modeling, reducing virtual–real mapping errors, ensuring real-time performance, and integrating multi-layer architectures effectively.

[6] Roy Featherstone presents a comprehensive study on rigid body dynamics algorithms for multibody systems in robotics. His work introduces efficient computational methods for analyzing the motion of articulated mechanisms using spatial vector algebra and recursive Newton–Euler formulations. These algorithms significantly reduce the computational complexity of dynamic calculations in robotic manipulators. The study demonstrates how accurate dynamic modeling enables precise torque estimation, motion prediction, and stability analysis in robotic systems, forming a theoretical foundation for modern multibody dynamic simulation frameworks.

[7] Mark W. Spong and co-authors provide a detailed framework for robotic system modeling, control, and dynamic

analysis in their work on robotic manipulators. The study explains the integration of forward kinematics, inverse kinematics, and dynamic modeling for analyzing robotic motion and control behavior. By applying classical mechanics and control theory, the authors demonstrate how robotic manipulators can be mathematically modeled to predict motion trajectories and actuator torque requirements. Their work serves as a foundational reference for designing simulation-based robotic analysis systems.

[8] Erwin Coumans introduces the development of the Bullet Physics Engine, a real-time physics simulation platform widely used in robotics research and virtual environments. The system enables accurate modeling of rigid body dynamics, collision detection, and constraint-based motion simulation. The framework supports robotic simulation through its extension, PyBullet, ● Collision geometry which provides Python-based APIs for controlling robotic models and analyzing physical interactions. This work highlights the importance of physics-based simulation environments for testing robotic systems before physical deployment.

[9] Joseph Redmon proposed the YOLO algorithm for real-time object detection in computer vision applications. Unlike traditional multi-stage detection frameworks, YOLO performs object localization and classification in a single neural network pass, significantly improving detection speed and efficiency. The study demonstrates that YOLO can accurately detect and track multiple objects in real time, making it suitable for robotic perception systems where rapid environmental awareness is required for tasks such as object grasping and manipulation.

[10]Michael Grieves introduced the concept of a digital twin as a simulated representation of a physical system that continuously exchanges data with its real-world counterpart. His research highlights how digital twin technology can be used to simulate, analyze, and optimize complex engineering systems throughout their lifecycle. In the context of robotics and manufacturing, digital twins enable virtual testing of robotic operations, predictive maintenance, and performance optimization. This approach allows engineers to validate system behavior through simulation before implementing changes in real hardware.

III. PROPOSED SOLUTION

The proposed solution introduces a structured multibody dynamic simulation framework for evaluating the torque requirements and payload feasibility of a robotic manipulator within a physics-based simulation environment. The primary objective of the system is to determine whether a robotic arm can safely lift and manipulate a given payload without exceeding the torque limits of its joint actuators.

Unlike traditional validation methods that rely on physical prototype testing or simplified static calculations, the proposed framework performs dynamic evaluation under realistic operating conditions. These conditions include gravitational forces, inertial effects, rigid-body interactions, and joint constraints. The system is implemented using the physics simulation engine PyBullet, which enables real-time rigid-body simulation and direct access to joint torque computation.

The proposed framework is structured into several functional modules that collectively perform modeling, simulation, analysis, and feasibility evaluation of robotic payload handling.

A. URDF-Based Multibody Modelling

The robotic manipulator is modeled using the Unified Robot Description Format (URDF), which provides a structured representation of links and joints. Each link in the robotic arm is defined with:

- Mass properties
- Inertia tensors
- Visual geometry

Revolute joints are defined with:

- Axis of rotation
- Joint limits
- Damping parameters
- Maximum torque constraints

This modelling approach ensures that the robot behaves as a realistic multibody rigid system rather than a simple kinematic chain. Accurate mass and inertia definition is critical because torque estimation directly depends on these physical parameters.

B. Physics-Based Dynamic Environment Initialization

The simulation environment is initialized with gravity and rigid-body dynamics enabled. A ground plane is introduced to simulate realistic environmental interaction. The physics engine computes:

- Gravitational forces
- Reaction forces between links
- Inertial effects during motion

- Constraint forces at joints

By enabling these physical factors, the simulation replicates real-world operating conditions of industrial robotic arms.

C. Integrated Kinematic Computation

Before dynamic evaluation, the robotic arm must reach a specific configuration suitable for lifting the payload. The system integrates:

1. Forward Kinematics – Computes the end-effector pose based on joint angles.
2. Inverse Kinematics – Determines required joint angles for a desired target position.

These computations ensure that the manipulator reaches a stable pre-lifting configuration before torque analysis begins. This step is essential because joint torque values vary depending on arm configuration.

D. Real-Time Torque Evaluation Mechanism

The core contribution of the proposed solution lies in real-time torque monitoring. Once the payload is attached to the end-effector and the lifting motion begins, the physics engine continuously computes joint torques required to maintain motion and stability.

The torque for each joint is influenced by:

- Link mass distribution
- Payload mass
- Distance of payload from base joint
- Angular acceleration
- Gravitational force

The system extracts peak torque values during motion execution and stores them for analysis.

E. Payload Feasibility Classification

After torque computation, the system performs actuator constraint validation. Each joint has a predefined maximum allowable torque based on motor specifications.

The feasibility logic is defined as:

- If ,$\tau\text{-}computed.$≤,$\tau\text{-}max.$ <for all joints→Load Feasible>
- If any joint exceeds ,$\tau\text{-}max.$<→Load not Feasible>

This automated classification provides immediate feedback regarding whether the robotic arm can safely lift the specified payload.

F. Incremental Load Testing Strategy

To determine the maximum payload capacity, the system supports incremental load testing. Payload mass is gradually increased in controlled steps, and torque values are recorded for each iteration.

This process helps in:

- Identifying the critical joint limiting payload capacity
- Determining safe operational boundaries
- Understanding torque distribution patterns

Typically, the base or shoulder joint experiences the highest torque due to cumulative load effects.

G. Data Logging and Performance Analysis

All simulation data including:

- Joint positions
- Joint velocities
- Computed torques
- Payload values
- Timestamp information

are logged into structured CSV files. This data enables graphical visualization of torque trends and supports experimental validation.

H. Key Innovations of the Proposed Solution

The proposed solution differs from existing approaches in several ways:

1Combines both kinematic and dynamic modeling within a unified framework.
2. Performs real-time torque monitoring rather than static approximation.
3. Automates payload feasibility decision-making.
4. Eliminates early-stage hardware testing.
5. Provides scalable architecture for adding advanced control modules.

IV. SYSTEM ARCHITECTURE

The system architecture has been designed as a modular, simulation-driven framework that integrates robotic modeling, physics-based dynamics, torque monitoring, and payload feasibility validation into a unified workflow. The architecture ensures scalability, accuracy, and repeatability while minimizing hardware dependency.

The complete system is divided into six major layers:
1. User Interaction Layer
2. Robotic Modeling Layer
3. Kinematic Processing Layer
4. Multibody Simulation Layer
5. Torque Monitoring and Data Logging Layer
6. Load Validation andOutput Layer

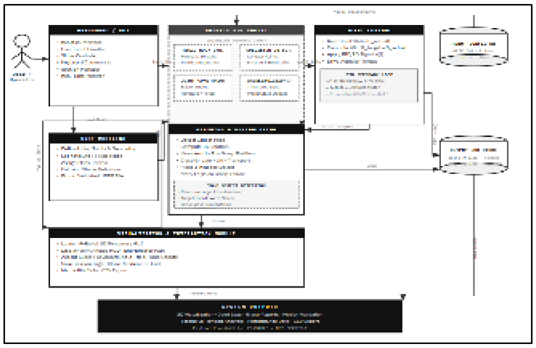


Fig 1.1 System Architecture diagram of robotic manipulator

Each layer operates independently while exchanging structured data with adjacent components, enabling modular upgrades and performance optimization.

A. SystemOverview

The architecture follows a sequential data-processing pipeline. A user provides payload input, which is processed through kinematic and dynamic modules. The simulation engine computes joint torques in real time, and a decision module evaluates load feasibility based on actuator constraints.

The overall objective of the architecture is to transform a simple payload input into a dynamically valid

B. User Interaction Layer

This layer acts as the entry point to the system. It is responsible for:

- Accepting payload mass input
- Initiating simulation runs
- Displaying feasibility results
- Triggering incremental load testing

The interface abstracts all complex computations from the user, ensuring simplicity while maintaining technical depth internally.

C. Robotic Modeling Layer

The robotic arm is modeled using a URDF-based structure. Each link and joint is defined with precise physical properties including:

- Mass
- Inertia matrix
- Joint axis
- Motion limits
- Maximum torque capacity

This layer ensures that the robotic manipulator behaves as a true multibody rigid system rather than a simplified geometric structure.

The modeling data is imported into the physics engine for simulation execution.

D. Kinematic Processing

The kinematic layer handles motion configuration prior to dynamic evaluation. It performs:

1. Forward Kinematics – Computes end-effector pose from joint parameters.
2. Inverse Kinematics – Determines required joint angles to reach a target lifting position.

This ensures that torque evaluation occurs under realistic joint configurations. Since torque requirements vary depending on arm posture, this layer is critical for accurate load analysis.

E. Multibody Simulation Layer

The core computational engine of the system is implemented using PyBullet.

This layer enables:

- Gravity-enabled simulation
- Rigid-body dynamics computation
- Joint constraint enforcement
- Collision detection
- Motor torque feedback

During simulation execution, the physics engine computes:

- Gravitational torque components
- Inertial forces
- Coriolis and centrifugal effects
- Reaction forces between connected links

The manipulator operates under realistic environmental conditions, ensuring high-fidelity dynamic behavior.

F. Torque Monitoring and Data Logging Layer

This layer continuously extracts torque values from each joint motor during simulation runtime. Key responsibilities include:

- Recording real-time torque values
- Identifying peak torque per joint
- Logging data into CSV format
- Storing joint position and velocity information

This structured logging enables post-simulation performance analysis and graphical visualization of torque trends.

G. Load Validation and Decision Layer

The final layer performs payload feasibility assessment. The logic follows a constraint-checking mechanism:

- Compare computed torque values with predefined actuator torque limits.
- < If all joints operate within limits → Load Feasible>
- <If any joint exceeds threshold →Load not Feasible>

This automated validation process provides immediate decision output without manual calculation.

H. End-to-End Data Flow

The overall system flow proceeds as follows:

1.User inputs payload mass.
2. Kinematic module computes valid joint configuration.
3.Payload is attached to the end-effector.
4.Physics simulation executes under gravity.
5.Joint torques are computed in real time.
6. Peak torque values are extracted.
7. Validation module compares torque with actuator limits. 8. Feasibility result is displayed.

This structured pipeline ensures traceability, reproducibility, and reliability.

I. Architectural Advantages

The proposed architecture offers several key benefits:

- Modularity: Each layer can be independently modified or upgraded.
- Scalability: Additional joints or robotic configurations can be integrated easily.
- Safety: Eliminates risk of mechanical damage during early testing.
- Cost Efficiency: Reduces dependency on physical prototypes.
- Repeatability: Simulations can be executed multiple times under identical conditions.

J. Architectural Flexibility for Future Expansion

The current architecture can be extended to support:

- Advanced PID or computed torque control
- Collision-aware grasping mechanisms
- Energy consumption analysis
- Reinforcement learning-based load optimization
- Digital twin integration

The modular design ensures seamless integration of such advanced components.

K. Technology Stack

The system is implemented using the following technologies:

- Programming Language: Python for simulation control, kinematic computation, and torque analysis.
- Physics Engine: PyBullet for multibody dynamics simulation, gravity modeling, and real-time torque computation.
- Robot Modeling Format: URDF (Unified Robot Description Format) for defining link mass, inertia, joint limits, and collision properties.
- Numerical Computation: NumPy for matrix operations, kinematic calculations, and dynamic parameter processing.
- Data Logging: CSV module for recording joint torque values, payload variations, and simulation timestamps.
- Visualization: Matplotlib for plotting torque vs payload graphs and performance analysis charts
- Development Environment: Windows-based system with local simulation execution.

- Simulation Mode: Real-time physics-based rigid-body simulation environment.

V. METHODOLOGY

The methodology of the proposed system is focused on physics-based multibody dynamic modeling, real-time torque computation, and structured payload feasibility validation. The research follows a simulation-driven experimental design that integrates kinematic configuration, dynamic evaluation, and incremental load testing within a unified framework.

The methodology is divided into five major stages:

1.Robotic Modeling and Parameter Definition
2 Kinematic Configuration
3.Dynamic Simulation and Torque Computation
4.Payload Increment Testing
5.Data Logging and Validation

A. Robotic Modeling and Parameter Definition

The robotic manipulator is modeled as a serial multibody chain consisting of rigid links connected through revolute joints. Each link is defined with:

- Mass $,m\text{-}i.$
- Center of mass $,c\text{-}i.$
- Inertia tensor $,I\text{-}i.$
- Geometric constraints

Joint parameters include:

- Axis of rotation
- Joint limits
- Maximum torque capacity.

The robot description is implemented using URDF and imported into PyBullet for dynamic simulation.

Accurate mass and inertia modeling is critical because torque estimation depends directly on these parameters.

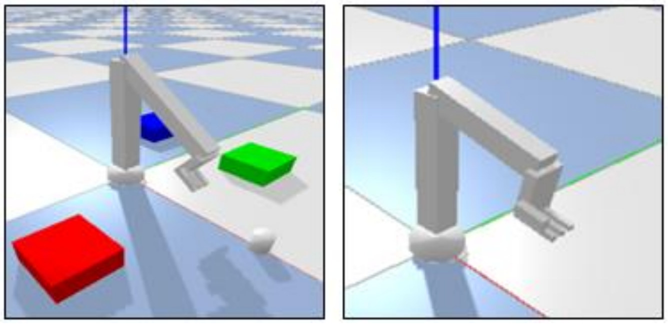


Fig 1.2 Robotic manipulator modelling using URDF

B. Kinematic Configuration

Before performing dynamic evaluation, the manipulator must reach a stable lifting configuration. The following kinematic procedures are applied:

1) Forward Kinematics Forward kinematics computes the end-effector position and orientation as:

$$T=,T\text{-}1.,T\text{-}2.,T\text{-}3.\ldots,T\text{-}n.$$

Where $,T\text{-}i.$represents the homogeneous transformation matrix of each joint.
This determines the spatial pose of the end-effector for a given set of joint angles.

2) Inverse Kinematics Inverse kinematics determines joint variables required to reach a desired target position:

$$q=,f\text{-}{-}1.(x,y,z)$$

Where:
- q= Joint angle vector
- $,z.$= Desired end-effector position

This ensures torque evaluation occurs at a realistic operational configuration.

C. Dynamic Simulation and Torque Computation

The manipulator dynamics are governed by the standard rigid-body dynamic equation:

$$\tau=M(q),q.+C(q\ ,q.),q.+G(q)$$

Where:
- τ= Joint torque vector
- $M(q)$= Mass matrix
- $C(q,q.)$= Coriolis and centrifugal effects
- $G(q)$= Gravitational torque component

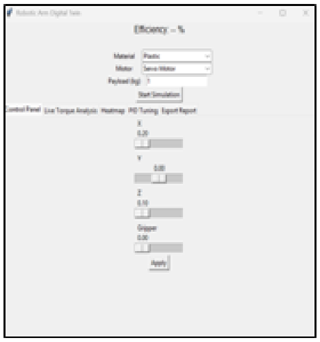
Dynamic Evaluation Procedure
1. Payload mass $,m\text{-}p.$is attached to the end-effector.
2. Gravity is enabled in simulation.

3. Joint motors execute controlled motion.
4. The physics engine computes real-time torque for each joint.
5. Peak torque values are extracted.

Torque increases proportionally with:

- Payload mass
- Distance from base joint
- Angular acceleration

In serial manipulators, proximal joints (base and shoulder) typically experience the highest torque due to cumulative load effects.



Efficiency: -- %

Material: Plastic

Motor: Servo Motor

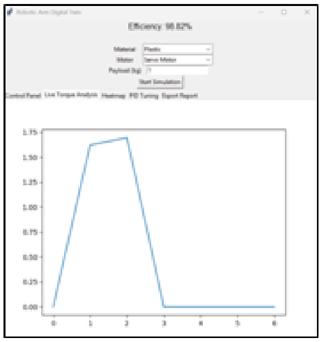
Payload (kg): 1

X 0.20

Y 0.00

Z 0.10

Gripper 0.00



Efficiency: 98.82%

Material: Plastic

Motor: Servo Motor

Payload (kg): 1

Fig 1.3.1 Joint torque values control slider

Fig 1.3.2 Live torque analysis of manipulator

D. Payload Increment Testing

To determine the maximum load capacity of the robotic arm, an incremental payload testing strategy is applied. Algorithmic Steps:

1. Initialize payload mass $m_{p} = m_{initial}$.
2. Run dynamic simulation
3. Record peak torque values
4. Compare with actuator torque limits
5. <If within limits→Increase payload>
6. <If exceeded→Stop and record maximum feasible load>

This iterative approach determines the safe operating boundary of the manipulator.

E. Torque Limit Validation

Each joint has a predefined maximum allowable torque τ_{max}.

The feasibility condition is:

$$Feasible\ if\ \tau_{i} \leq \tau_{max,i}\ \forall i$$

If any joint exceeds its threshold:

$$\tau_{i} > \tau_{max,i}$$

The system classifies the load as infeasible.

F. Data Logging and Performance Analysis

During simulation, the following parameters are logged:

- Joint position q
- Joint velocity q.
- Computed torque τ
- Payload mass
- Simulation time

Data is stored in structured CSV format for:

- Torque vs time plotting
- Payload vs torque trend analysis
- Peak torque identification

Graphical analysis enables validation of dynamic behavior and identification of limiting joints.

G. Experimental Validation Strategy

The simulation experiments are conducted under controlled conditions:

- Fixed gravitational acceleration
- Identical joint configuration
- Gradually increasing payload
- Constant motor control parameters

Performance metrics evaluated include:

- Maximum feasible payload
- Peak torque distribution
- Joint stress distribution
- Stability under load

H. Methodological Advantages

The proposed methodology provides:

- Physics-based dynamic validation
- Real-time torque monitoring
- Automated payload classification
- Reduced dependency on hardware testing
- Repeatable experimental conditions

Compared to traditional static torque estimation, this methodology accounts for inertial and gravitational effects, leading to more accurate and realistic results.

VI. RESULTS AND EVALUATION

This section presents the experimental evaluation of the proposed multibody dynamic simulation framework for robotic load feasibility assessment. The objective of the evaluation is to determine the maximum safe payload capacity of the robotic manipulator based on real-time torque monitoring under gravitational conditions.

A. Experimental Setup

The robotic arm was modeled using URDF and simulated in a gravity-enabled environment using PyBullet.

The following experimental conditions were maintained

- Fixed joint configuration during lifting
- Standard gravitational acceleration
- Incremental payload testing
- Predefined maximum torque limits for each joint
- Real-time torque logging during simulation

Payload mass was increased step-by-step to observe corresponding torque variations in each joint.

B. Torque Distribution Analysis

During simulation, torque values were recorded for all revolute joints. It was observed that:

- The base and shoulder joints experienced the highest torque.
- Distal joints showed comparatively lower torque values.
- Torque increased proportionally with payload mass.
- This behavior aligns with multibody dynamics principles, where proximal joints support cumulative loads from subsequent links and the attached payload.

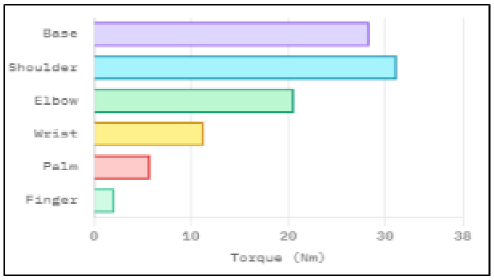


Fig 1.4 Joint torque distribution

C. Payload vs Torque Performance

Table I presents a representative torque variation with increasing payload.

Table 1.1. payload vs maximum joint torque

Payload (kg)	Max Torque (Nm)	Feasibility
0.5	12.4	Feasible
1.0	18.7	Feasible
1.5	24.9	Feasible
2.0	31.2	Feasible
2.5	38.5	Not Feasible

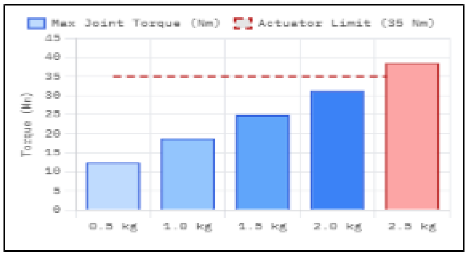


Fig 1.5 Payload vs Maximum Joint torque

From the results, it was observed that torque exceeded actuator threshold at 2.5 kg payload, indicating that the maximum safe payload capacity of the manipulator lies between 2.0 kg and 2.5 kg under the tested configuration.

D. Critical Joint Identification

The simulation identified the shoulder joint as the limiting factor for payload capacity. Once this joint reached its maximum allowable torque, the system classified the load as infeasible. This confirms that load capacity is typically constrained by the proximal joint due to higher moment arms and accumulated link weight

.

E. Graphical Evaluation

Torque vs Payload plots demonstrated a near-linear relationship between increasing payload and required torque. Peak torque values were extracted from CSV logs and visualized using Matplotlib for analysis.

The graphical results confirm:

- Stable dynamic behavior under feasible loads
- Rapid torque escalation beyond threshold
- Clear overload detection capability

F. Validation of Proposed Approach

The experimental results validate that multibody physics simulation can reliably predict actuator overload conditions before physical deployment. The system successfully identifies maximum safe payload thresholds using torque-based decision logic.

The integration of kinematic modeling and real-time torque monitoring enhances predictive reliability and supports safe robotic system design.

VII. DISCUSSION

The experimental results demonstrate that multibody dynamic simulation provides an effective and reliable method for evaluating the load feasibility of robotic manipulators. By integrating URDF-based modeling, kinematic configuration, and real-time torque monitoring within the PyBullet environment, the proposed system successfully identifies actuator overload conditions before physical deployment.

A major finding from the experimental analysisis that torque demand increases proportionally with payload mass. This linear relationship aligns with classical rigid-body dynamics principles, where torque is directly influenced by the applied load and its distance from the joint axis. As expected in serial manipulators, proximal joints—particularly the shoulder joint—experience significantly higher torque due to cumulative link weight and external payload forces. This confirms theoretical predictions derived from the Newton–Euler dynamic formulation.

The results further indicate that load feasibility is highly dependent on joint configuration. Even for identical payload values, torque requirements vary depending on arm posture and end-effector position. This highlights the importance of integrating both kinematic and dynamic modeling rather than relying solely on static torque approximation. Static calculations often neglect inertial and gravitational interactions that occur during motion, leading to underestimation of required torque.

Another significant result of this study is thevalidation of simulation-based testing as a cost-effective alternative to hardware experimentation. Traditional payload testing requires physical prototypes and carries risks such as motor overheating, gear damage, or structural stress. The proposed simulation framework eliminates these risks while providing repeatable and controlled testing conditions. Moreover, incremental load testing enables precise identification of maximum safe payload thresholds.

However, certain limitations must be acknowledged. Simulation accuracy depends heavily on correct modeling of link mass, inertia tensors, and actuator torque limits. Any inaccuracies in the parameters may lead to deviation from real-world behavior. Additionally, factors such as joint friction, motor efficiency, gear backlash, and material flexibility were not deeply modeled and may influence real hardware performance.

Despite these limitations, the proposed approach establishes a strong foundation for predictive robotic performance evaluation. The modular system architecture allows integration of advanced control strategies, energy consumption analysis, and hardware-in-the-loop validation in future extensions.

Overall, the discussion confirms that physics-driven multibody simulation is a reliable and scalable solution for robotic load feasibility assessment. The framework bridges the gap between theoretical dynamic modeling and practical engineering validation, enabling safer and more efficient robotic system design.

VIII. PERFORMANCE MEASURE

The proposed simulation framework demonstrated:

- Accurate torque estimation under gravity
- Automated overload detection
- Reduced testing time compared to physical experiments
- Fully repeatable experimental conditions

Compared to traditional hardware-based testing, the simulation approach significantly reduces development cost and risk of mechanical damage.

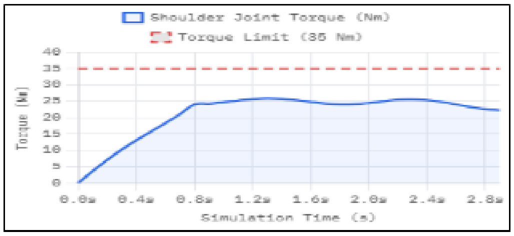
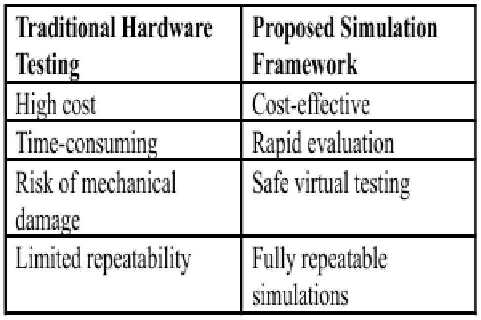


Fig 1.6 Torque vs Simulation time



Traditional Hardware Testing	Proposed Simulation Framework
High cost	Cost-effective
Time-consuming	Rapid evaluation
Risk of mechanical damage	Safe virtual testing
Limited repeatability	Fully repeatable simulations

Table 1.2. Advantages Over Traditional Methods

IX. CONCLUSION

This paper presented a multibody dynamic simulation framework for evaluating the load feasibility and torque requirements of a robotic manipulator using a physics-based simulation approach. The proposed system integrates robotic modeling using the URDF, kinematic analysis, real-time torque monitoring, and automated payload feasibility evaluation within the PyBullet environment.

The developed framework enables accurate analysis of robotic manipulator behavior under varying payload conditions by simulating realistic physical effects such as gravity, inertia, and joint constraints. Through dynamic simulation, joint torque values were continuously monitored while the manipulator executed lifting operations. The experimental observations show that torque requirements increase proportionally with payload mass and that proximal joints, particularly the base and shoulder joints, experience the highest mechanical load due to cumulative mass effects along the kinematic chain.

By comparing the computed torque values with predefined actuator limits, the system automatically determines whether a given payload can be safely manipulated without exceeding joint capacity. This automated feasibility evaluation provides immediate feedback regarding safe operating conditions for the robotic arm. In addition, incremental payload testing allows systematic identification of maximum safe load capacity and highlights the joints that act as critical constraints in the lifting process.

Compared with traditional hardware-based testing approaches, the proposed simulation-driven framework offers a cost-effective, flexible, and risk-free method for early-stage validation of robotic manipulator performance. The use of physics-based simulation enables engineers to analyze mechanical behavior, evaluate design feasibility, and optimize robotic configurations before physical implementation. Furthermore, the integration of object detection using the YOLO provides a foundation for intelligent perception-driven manipulation tasks within a digital twin simulation environment.

Although simulation accuracy depends on the correct modeling of physical parameters such as link mass, inertia properties, and actuator specifications, the results demonstrate that multibody dynamic analysis provides reliable insights into torque distribution and payload feasibility in robotic systems. The proposed framework effectively bridges the gap between theoretical dynamic modeling and practical robotic system evaluation.

Future work will be focused on extending the framework by incorporating advanced actuator models, friction and gear transmission dynamics,energy consumption analysis, and hardware-in-the-loop validation. Additional improvements may include integration with real robotic hardware, adaptive control algorithms, and enhanced digital twin synchronization for real-time monitoring and predictive maintenance.

Overall, the proposed approach establishes a scalable and physics-driven solution for robotic torque analysis and intelligent robotic system design.

REFERENCES

[1] G. Boschetti and T. Sinico, “Designing Digital Twins of Robots Using Simscape Multibody,” Robotics, vol. 13, no. 62, 2024.
[2] Y. Cen et al., “Digital Twin-Empowered Robotic Arm Control: An Integrated PPO and Fuzzy PID Approach,” Mathematics, vol. 13, no. 216, 2025.
[3] Q. Qin et al., “Robot Digital Twin Systems in Manufacturing: Technologies, Applications, Trends and Challenges,” Robotics and Computer-Integrated Manufacturing, vol. 97, 2026.
[4] R. Huang et al., “SyncTwin: Fast Digital Twin Construction and Synchronization for Safe Robotic Grasping,” 2026.
[5] “Digital Twin Design of a 5 Degrees of Freedom Industrial Robotic Arm for Engineering Education Purposes,” Measurement: Sensors, vol. 38, 2025.
[6] F. Tao, Q. Qi, A. Liu, and A. Kusiak, “Data-Driven Smart Manufacturing,” Journal of Manufacturing Systems, vol. 48, pp. 157–169, 2019.
[7] Tao and M. Zhang, “Digital Twin Shop-Floor: A New Shop-Floor Paradigm Toward Smart Manufacturing,” IEEE Access, vol. 7, pp. 7425–7435, 2019.
[8] Q. Qi and F. Tao, “Digital Twin and Big Data Towards Smart Manufacturing and Industry 4.0: 360 Degree Comparison,” IEEE Access, vol. 6, pp. 3585–3593, 2019
[9] S. Leng, Q. Liu, and J. Ye, “Digital Twin-Driven Smart Manufacturing: Connotation, Reference Model, Applications and Research Issues,” Robotics and Computer-Integrated Manufacturing, vol. 61, 2020.
[10] X. Lu, C. Liu, K. I.-K. Wang, H. Huang, and X. Xu, “Digital Twin-Based Smart Manufacturing: Connotation, Reference Model, Applications and Research Issues,” Robotics and Computer-Integrated Manufacturing, vol. 58, pp. 272–289, 2019.
[11] Y. Koren and X. Gu, “Reconfigurable Manufacturing Systems: Principles, Design, and Future Trends,” Journal of Manufacturing Systems, vol. 51, pp. 1–12, 2019.

[12] M. Uhlemann, C. Schock, C. Lehmann, S. Freiberger, and R. Steinhilper, “The Digital Twin: Realizing the Cyber-Physical Production System for Industry 4.0,” Procedia CIRP, vol. 61, pp. 335–340, 2019.

[13] J. Lee, H. Davari, J. Singh, and V. Pandhare, “Industrial Artificial Intelligence for Industry 4.0-Based Manufacturing Systems,” Manufacturing Letters, vol. 18, pp. 20–23, 2019.