Automating the Tower Crane: Integrating the Development and Simulation of Path Planning and Trajectory Tracking of Tower Crane in ROS Framework

Muhammad Muddassir[1], Mohamed A. A. Abdelkareem[2], Tarek Zayed[1] and Zoubeir Lafhaj[3]

[1]Department of Building and Real Estate, The Hong Kong Polytechnic University, Hong Kong
[2]Department of Mechanical Engineering, The Hong Kong Polytechnic University, Hong Kong
[3]Laboratoire de Mécanique Multiphysique Multiéchelle, LaMcube, UMR 9013, Centrale Lille, CNRS, Université de Lille, F-59000 Lille, France

muhammad.muddassir@polyu.edu.hk, mohamed.a.ali@mu.edu.eg, tarek.zayed@polyu.edu.hk, zoubeir.lafhaj@centralelille.fr

Abstract -

In modern construction sites, tower cranes play a crucial role but often rely on multiple human operators. Despite the advancements of the Construction 4.0 era, a comprehensive framework for automated tower crane operations is currently lacking. This study proposes a framework that integrates a real-scaled construction site and tower crane into a physics-based simulation in ROS (robot operating system) framework to enable collision-free motion planning and control. Specifically, we develop time-varying linear quadratic regulators (LQR) for trolley and jib control while employing a proportion-integrated-derivative (PID) method for hoisting control. Additionally, we utilise 5th-order quintic spline trajectories to plan the desired pose of the payload, reducing acceleration discontinuities. The framework's effectiveness is validated through simulations of a real-scaled tower crane and construction site equipped with LiDAR sensors. The results demonstrate that higher-order trajectories effectively minimise oscillations in unactuated systems. Our scalable framework holds promise for real-scale operations in the field of tower crane automation.

Keywords -

Tower Cranes; Automation in Construction Sites; Time-varying LQR; Robot Operating System (ROS); Underactuated Systems

1 Introduction

Tower cranes (TCs) are indispensable in modern construction sites, serving vital purposes such as lifting and handling heavy construction materials, precise placement of concrete, installation of essential equipment and machinery, and efficient site logistics during construction processes. In addition, TCs are valuable in various high-hazard environments, such as shipyards, nuclear plants, and factory floors, where they facilitate the safe transportation of high-risk materials. Efficient operation of TCs in construction and logistics sites minimises operation time per load and ensures high safety standards for workers and operators. In this sense, TCs automation and control can enhance efficiency, improve safety, and optimise productivity in construction and logistic sites where multiple TCs are also considered. Automating the TC's operations can minimise human error, precise load control, real-time monitoring, streamline operations, and reduce risks.

In practice, most industrial TCs are manually controlled and still rely on various human operators (TC operator, signalperson and lifting supervisor) despite the availability of advanced sensing and monitoring systems. The on-site working environment poses many uncertainties and variations, challenging efficient TC operations. Failure in TC operation control can lead to material damage and pose risks to on-site workers, with an uncertain amount of time waste and operation cost [1, 2]. Generally, gripping the load, lifting, transporting from one location to another, lowering, and releasing are the main TC operations [3]. The most time-consuming operation is load transport between two locations, and it requires a skilled operator to minimise the swings and collision of the load during transportation. Automation of TC operations has been a research interest in the construction industry to improve efficiency, mitigate risks, and decrease dependence on human operators.

Numerous studies have delved into various aspects of automated tower crane control [4, 5, 6, 7, 8, 9]. An adaptive control method for tower cranes was developed by [10, 11, 12, 13] to improve outdoor efficiency by addressing parameter uncertainties and disturbances. The robust control methods for controlling jib, trolley, and hoisting of TC were developed by [8, 14]. Similarly, the energy-based methods to develop a regulator controller for achieving the desired state while minimising the swings in hanging payload can be found in [13, 15, 16] In a separate study, He et al. [17] proposed an anti-collision system for tower cranes that considers the lifting of heavy objects, specifically focusing on the dangerous area concept of cylindrical heavy objects to simulate real working situations. For monitoring and planning the collision-free lifting operations, [18, 19, 20, 21, 22] proposed various path planning methods using CAD and BIM information, specifically focusing on time efficient and obstacle avoidance while transporting or lifting from blind spot of crane operators.

The application of computer-aided design and modelling in tower crane operations has facilitated convenient modelling, simulation, and control via physics-based simulations. This approach enhances tower cranes' understanding and motion analysis in full-scale construction sites. Kang and Miranda [23] developed a physics-based dynamic model of TCs to improve the understanding of crane dynamics and develop computer-aided training methods for crane operators to reduce accidents and enhance overall safety.

In this study, a physics-based simulation module for a tower crane situated in a full-scale construction site was developed utilising the ROS framework. Subsequently, a time-variant Linear Quadratic Regulator (LQR) control system was proposed to govern the trolley and jib movements, while a Proportional-Integral-Derivative (PID) method was employed for controlling the hoisting action. 5th-order quintic spline trajectories were utilised to facilitate planning the payload's desired pose and reduce acceleration discontinuities. While previous studies have explored various aspects of tower crane automation, our work presents a comprehensive framework that addresses collision-free motion planning, trajectory planning, and control algorithms.

2 Why ROS?

Robot Operating System (ROS) is a popular open-source framework for developing robotic systems. It provides a collection of libraries and tools to help developers create complex robotic systems. ROS has a wide range of capabilities, including:

- **Hardware Abstraction**: ROS provides a hardware abstraction layer that allows one to write code that can run on different types of robots with minimum modifications.

- **Message Parsing**: ROS provides a message-parsing system that allows different parts of a robot system to communicate. This makes it easy to develop modular systems where various components can be developed independently. This capability is scalable to multi-agent configurations, where each agent can communicate with the master or other agent.

- **Simulation**: It provides a simulation environment called Gazebo [24] that allows one to test their code in a virtual environment before deploying it on a real robot. Gazebo can simulate the dynamics of a robot, including the effects of gravity, friction, and other physical forces like wind and magnetic field. Gazebo supports various sensors, including cameras, lidars, sonars, and IMU. This makes it possible to integrate sensor data into the control algorithms of a robot for offline development, testing, and validation.

- **Scalability**: ROS is highly scalable, which means it can be used to develop robotic systems of different sizes and configurations. This makes it a versatile platform for developing automated systems for various applications on a small scale and then porting it to a more extensive system.

- **Flexibility**: ROS is a flexible platform that allows developers to create custom modules and libraries that can be integrated into similar robotic systems. This makes developing complex robotic systems that can perform various tasks uncomplicated.

Despite its many capabilities, tower crane integration in the ROS framework has some challenges. For example, the tower crane system's complexity, size, and scale differ from most systems for what ROS tools are available. Furthermore, the dynamics of the tower crane contain flexible cable, unactuated swing in radial and tangential directions, and a complex pulley mechanism. Despite this, the integration of tower cranes in ROS, presented here, shows the potential of robotising the tower crane system just like other robotic systems. Finally, ensuring the safety of the crane and its surroundings is of utmost importance, and this requires careful consideration, evaluation and validation of the control algorithms and sensor configurations used in the system.

3 Dynamics Model of a Tower Crane

A tower crane (more specifically, a hammerhead TC) is a five-degree-of-freedom (5-DOF) nonlinear dynamic system, as shown in Figure 1. 3-DOF are actuated, namely jib, trolley and hoisting cable. The generalised coordinates of the jib, trolley and hoisting cable will be γ, x and l, respectively. Whereas 2 of them are unactuated: radial and tangential swings. ϕ and θ denote generalised coordinates of radial and tangential swings, respectively. The jib is the rotating arm housed over a tower (known as a mast) and powered by an electric motor with a gear mechanism. The trolley is housed on the jib and can travel in both inward or outward radial directions. A cable-driven mechanism powered by the second electric motor pushes or pulls the trolley. A pulley system powered by the third electric motor enables a tower crane for lifting/hoisting motion.

In this paper, we have adopted the dynamical system reported by [25]. They used the Lagrangian approach to study the dynamics of a tower crane. The position vector of trolley position and payload position is given by:

$$\mathbf{x}_P = [x - l\cos(\theta)\sin(\phi),\ l\sin(\theta),\ -l\cos(\theta)\cos(\phi)]^T \tag{1}$$

$$\mathbf{x}_T = [x,\ 0,\ 0]^T \tag{2}$$

$\mathbf{x}_P$ is the position vector of the payload and $\mathbf{x}_T$ the position vector of the trolley. The linear velocity of the trolley and payload can be calculated as:

$$\dot{\mathbf{x}} = \frac{d\mathbf{x}}{dt} + \dot{\gamma} \times \mathbf{x} \tag{3}$$

Now, the kinetic and potential energies can be written as:

$$T = \frac{1}{2}m\langle \dot{\mathbf{x}}_P, \dot{\mathbf{x}}_P\rangle + \frac{1}{2}M\langle \dot{\mathbf{x}}_T, \dot{\mathbf{x}}_T\rangle + \frac{1}{2}J_o\dot{\gamma}^2 \tag{4}$$

$$V = -mgl\cos(\theta)\cos(\phi) \tag{5}$$

Here, m and M are the mass of the payload and trolley, respectively, whereas g is the gravitational constant. The generalised displacement vector is $\mathbf{q} = [x,\ \phi,\ \gamma,\ \theta,\ l]^T$ and generalised forces correspond to $\mathbf{q}$ is $\mathbf{F} = [F_x,\ 0,\ T_\gamma,\ 0,\ F_l]^T$.

Lagrangian is the difference of kinetic and potential energies $\mathcal{L} = T - V$ and the Lagrangian equation will be:

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}_j}\right) - \frac{\mathcal{L}}{\dot{q}_j} = F_j, \quad j = 1, 2, 3, 4, 5 \tag{6}$$

This will provide the equation of motion (EOM) similar to the one reported in [25]. The derived equations are complex and highly coupled. To devise the controllers from these equations, we have linearised the equation of motion and divided the system into two subsystems for designing two LQR controllers. During the linearization of the system, we assumed small swing angles, considered the parameters of each subsystem as time-varying and ignored the derivative of coupled variables in each subsystem, resulting in 2 simplified subsystems. The controller $\mathbf{K}_T$ to position the trolley and minimise the radial swing angle will be derived from Subsystem 1 ($\mathcal{S}_1$). Where $\mathcal{S}_1$ is defined as:

$$\ddot{x} + m_t\phi = \bar{F}_x \tag{7}$$

$$l\ddot{\phi} + g\phi - \ddot{x} = 0 \tag{8}$$

Here, $m_t = m/M$ and $\bar{F}_x = F_x/M$. The controller $\mathbf{K}_\gamma$ to rotate the jib and minimise the tangential swing angle will be derived from Subsystem 2 ($\mathcal{S}_2$), which is defined as:

$$(1 + M_r x^2)\ddot{\gamma} - m_r g x\theta = \bar{T}_\gamma \tag{9}$$

$$l\ddot{\theta} + g\theta - x\ddot{\gamma} = 0 \tag{10}$$

Here, $m_r = m/J_o$, $M_r = M/J_o$ and $\bar{T}_\gamma = T_\gamma/J_o$. The subsystems $\mathcal{S}_1$ and $\mathcal{S}_2$ will be utilised in the next section to derive the time-varying LQR controller for the trolley and jib positioning.

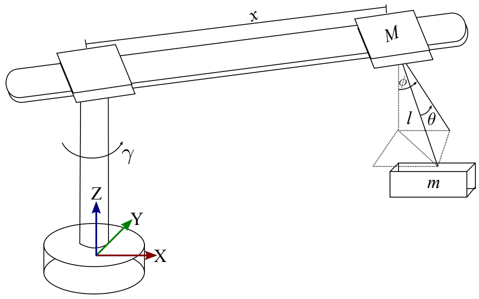

Figure 1. Dynamics of a tower crane with the 5 degrees of freedom.

4 Tower Crane Control Methods

4.1 Problem Statement

Here, we aim to design a control strategy that can bring the tower crane system to the desired states χ_d, $\mathbf{\Gamma}_d$ and $\mathbf{L}_d$ in finite time whilst respecting the hardware limitations such as F_x, T_γ and F_l. Two LQR controllers, namely K_χ and K_γ and a PID controller, K_l are designed to bring the states χ, $\mathbf{\Gamma}$ and $\mathbf{L}$ to χ_d, $\mathbf{\Gamma}_d$ and $\mathbf{L}_d$, respectively.

4.2 Controller Design

Figure 2 shows the architecture of the proposed control method. The real-scaled tower crane in the physics simulation (Gazebo [24] acts as a plant and provides all states of the system. Three controllers K_Γ, K_χ, and $K_\mathbf{L}$ are input with current states and the desired states, and they output forces F_x, F_l and torque T_γ required to apply on the trolley, payload, and jib for regulating the desired states. All outputs from the controller pass from the saturation function to ensure input to the tower crane system couldn't exceed the physical limits of the mechanical and electrical systems. In the proposed control method, the gain matrices for slew and trolley control are recomputed in each control cycle as the parameter of the subsystems $\mathcal{S}_1$ and $\mathcal{S}_2$ are varying with respect to the time. This kind of LQR implementation is also known as time-varying LQR.

4.3 Trolley Controller

A full-state feedback LQR controller is designed to control the trolley motion of the tower crane using the $\mathcal{S}_2$. The state space model of the $\mathcal{S}_2$ can be written as, $\dot{\chi} = \mathbf{A}_\chi\chi + \mathbf{B}_\chi F_x$ and the state vector will be

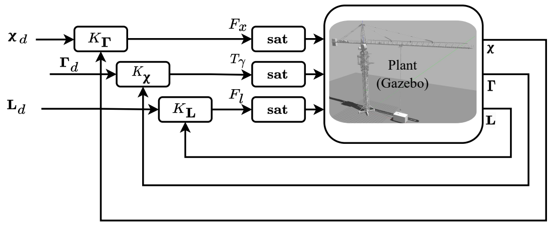

Figure 2. Flow diagram of the control methods.

$\chi = [x, \dot{x}, \phi, \dot{\phi}]$. Where:

$$\mathbf{A}_\chi = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -m_r g & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{g}{l(m_r+1)} & 0 \end{pmatrix}, \tag{11}$$

$$\mathbf{B}_\chi = \begin{pmatrix} 0 \\ 1 \\ 0 \\ -1/l \end{pmatrix} \tag{12}$$

The input force on the trolley F_x will be computed as:

$$F_x = -G_\chi(\chi - \chi_d) \tag{13}$$

Here, G_χ is the gain matrix and χ_d the desired states. G_χ is computed using the LQR method, where an optimisation function is minimised to compute the optimal gain matrix G_χ. The optimisation function for the LQR method is defined as:

$$\min \int \left(\chi^T \mathbf{Q}_\chi \chi + \mathbf{R}_\chi F_x^2 \right) \tag{14}$$

Where $\mathbf{Q}_\chi$ is a weight matrix to penalise the system states in order to tune the control performance and $\mathbf{R}_\chi$ the weight for input.

4.4 Slew Controller

For deriving the slew motion control, the full-state feedback system will be $\dot{\mathbf{\Gamma}} = \mathbf{A}_\Gamma \mathbf{\Gamma} + \mathbf{B}_\Gamma F_\gamma$ and the state vector will be $\mathbf{\Gamma} = [\gamma, \dot{\gamma}, \theta, \dot{\theta}]$. $\mathbf{A}_\Gamma$ and $\mathbf{B}_\Gamma$ are:

$$\mathbf{A}_\Gamma = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{m_r g x}{1+M_r x^2} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{g}{l}\left(\frac{1+x^2 m_r}{1+x^2 M_r}\right) & 0 \end{pmatrix}, \tag{15}$$

$$\mathbf{B}_\Gamma = \begin{pmatrix} 0 \\ \frac{1}{1+M_r x^2} \\ 0 \\ -\frac{x}{L(1+M_r x^2} \end{pmatrix} \tag{16}$$

Where the slew controller generates the input torque T_γ for the mast and jib joint. The control signal to regulate the slew motion, the T_γ is:

$$T_\gamma = -G_\Gamma(\mathbf{\Gamma} - \mathbf{\Gamma}_d) \tag{17}$$

Here $\mathbf{\Gamma}_d$ is the desired state for the jib and tangential swings. The optimal gain matrix G_Γ for the slew controller is computed using the LQR method, as computed for G_χ.

4.5 Hoisting Controller

The hoisting controller is responsible for regulating the height of the payload during the transporting or lifting operation. The hoisting motion is not directly responsible for inducing motion in unactuated joints (although variation in cable length can change the natural frequency of hanging load. But this coupling is not considered here). A standard PID controller is implemented to regulate the height of the payload.

$$F_l = G_p e_l + G_i \int_0^t e_l \mathrm{d}\tau + G_d \frac{\mathrm{d}e_l}{\mathrm{d}t} \tag{18}$$

Here, the G_p, G_i, G_d are the proportional, integral and derivative gain of the hoisting controller. These values were manually tuned to obtain optimal performance. e_l denotes the error between the current and desired length of the cable $e_l = l - l_d$

5 Trajectory Generation

The MoveIt provides the collision-free path from the current state of the TC to the desired state. It ensures that the planned path avoids the surrounding obstructions and minimises the mechanical effort of a complete path. When followed, the planned path is a function of an interpolated set of joint positions, ensuring collision-free plan execution. When each set of joint positions in the planned path is assigned a timestamp, it is called trajectory, and the controllers are required to bring the state of the system to the planned state at a particular timestamp. Here, the trajectory is obtained from a quintic spline, where each segment between interpolated path points is a polynomial function of order 5. This means that the trajectory is differentiable till the fifth order. Thus, the resulting velocities and accelerations of the trajectory will be a continuous function which leads to smoother transition of payload from stationary state to moving state and vice versa, as shown in Figure 5(g) and 5(h), and Figure 6(g) and 6(h).

6 Results

We have implemented the proposed controller on a real-scaled version of a tower crane in a physics-based simulation, namely, Gazebo [24]. Gazebo is a highly integrated

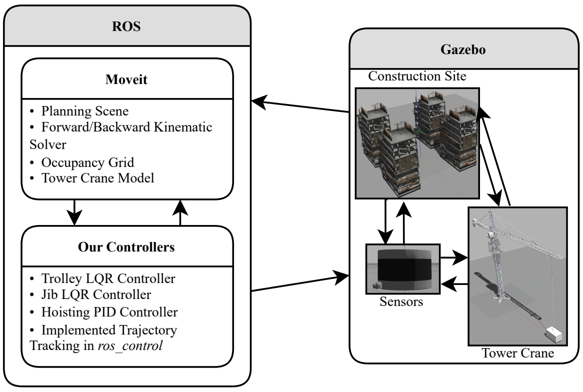

Figure 3. Flowchart of various modules used in the methodology.

simulation environment with the ROS framework and supports the ROS's message parsing mechanisms. This enables the development and validation of robot controllers and motion planning on a near-to-real system before porting the software to the robots in the real world. Figure 4 shows the planning screen for the physics-based simulated tower crane. Figure 4(a) illustrates the planning scene of MoveIt [26]. The tower crane in orange is the desired state, whereas grey shows the current state of the tower crane. The coloured voxel displays the occupancy grid, which is a way to notify the path planner about obstructions in the space during the planning paths.

Figure 4(b) shows the real-scaled tower crane in Gazebo, which can incorporate all the inertial and collision properties of a real tower crane, as well as the friction and damping values between each joint. It was assumed that the cable was non-stretchable and that the tower crane structure was non-deformable (or rigid) during motion. The hanging payload was free to oscillate in the radial and tangential direction of the jib. This was achieved by assuming that a tower crane is a 5-DOF robot (5 joints) whereas 3 were the actuated joints and two the unactuated joints.

The proposed framework and controller were validated in two scenarios: transporting and hoisting the payload from a blind spot (the part of a site which is not visible to a crane operator). In Gazebo, a construction site was designed with four under-construction buildings, an array of LiDAR sensors and a tower crane with payload. 3 LiDAR sensors were placed around the tower crane mast and 1 in front t of the construction site. Then, their point cloud data was fused to acquire the occupancy grid, as shown in Figure 4(a). MoveIt! (motion planner) can consider the occupancy grid as the obstruction in the space and plan the path accordingly in order to avoid any collision between the payload, tower crane and construction site during motion.

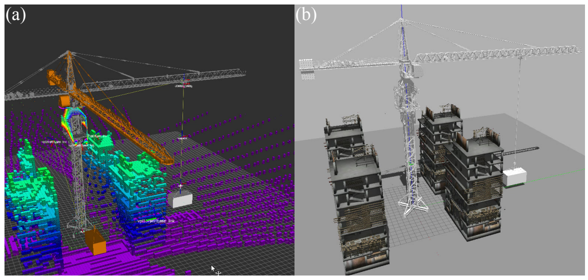

Figure 4. (a) Planning scene in RViz. (b) Simulated World in Gazebo

6.1 Transporting the Payload from Blindspot

Figure 5 shows the jib, trolley and hook trajectory tracking with input forces, input torque and swing angles. MoveIt computed the collision-free path, and then the trajectory was estimated using a quintic spline (5th order spline). Quintic spline is differentiable till the fifth order, thus minimising discontinuities in the derivative of acceleration (jerk). This induces lesser oscillation in hook and load at the start of motion. Figure 5(a) and 5(c) show that the trolley and jib (blue line) follows the trajectory (orange line) while inducing smaller radial and tangential swings, as shown in Figure 5(g) and 5(h). Figure 5(h) shows the radial swing during trajectory tracking and after 50s (see Figure 5(a)) when the desired trajectory of the trolley requires higher velocities, resulting in higher radial swings. However, the controller reduced the swings while bringing the trolley to the desired state.

6.2 Hoisting the Payload from Blindspot

Figure 6 shows the jib, trolley and hook tracking of trajectory with input forces, torque and swing angles during the second scenario. MoveIt computed the collision-free path; then, the trajectory was estimated using a quintic spline (5th-order spline). Figure 6(a), (c) and (e) show that the trolley, jib and hook (blue line) follow the trajectory (orange line) while inducing smaller radial and tangential swings, as shown in Figure 6(g) and (h). In this scenario, the desired distance of the trolley was relatively shorter than the jib and hook; thus, minute radial swings can be observed from Figure 6(g). The jib's desired state and current state are relatively closer in this scenario than the previous one. Thus, the jib controller inputs the highest allowable torque to the jib at the start and end of the planned motion in order to efficiently track the planned trajectory. Still, the tangential oscillation was considerably lower. As the planned trajectory is differentiable to the fifth order, the accelerations were continuous, resulting in smooth transitions of states from rest to motion and vice versa. The video demonstration of the proposed methods can be

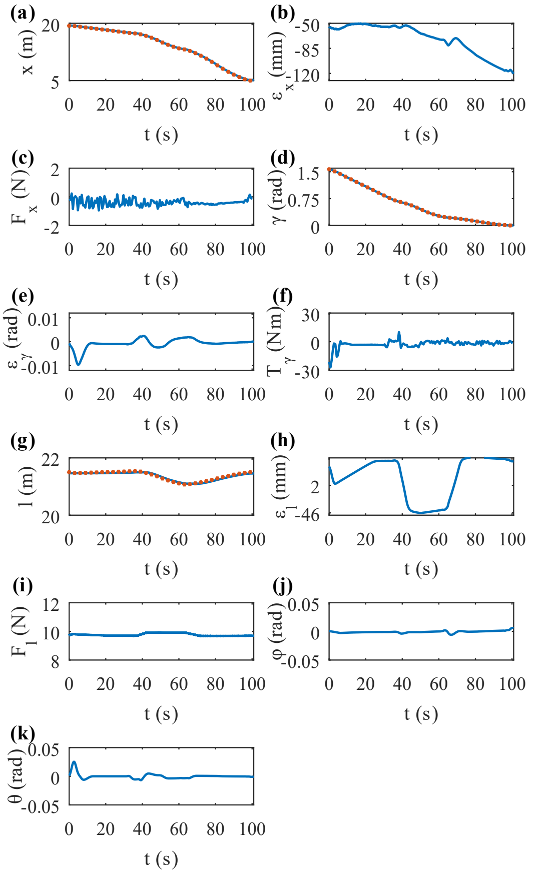

Figure 5. Transporting Trajectory Tracking. In graphs (a), (c) and (e), the dotted red line shows the planned (desired) trajectory and the blue lines in all graphs show the current state of the TC and input quantities. ε_x, ε_γ and ε_l are the difference between the desired and current state of trolly, jib and hook.

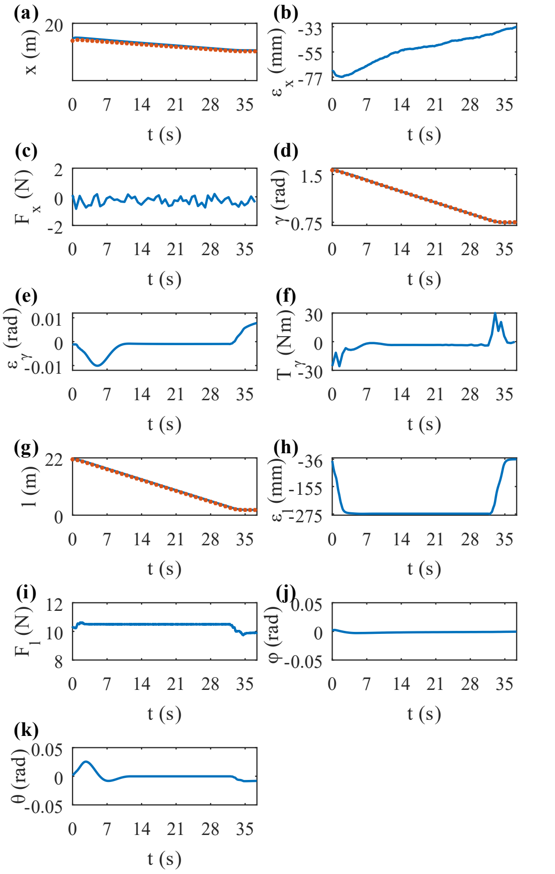

Figure 6. Hoisting Trajectory Tracking. In graphs (a), (c) and (e), the dotted red line shows the planned (desired) trajectory and the blue lines in all graphs show the current state of the TC and input quantities. ε_x, ε_γ and ε_l are the difference between the desired and current state of trolly, jib and hook.

on `https://github.com/muddassir93/ISARC2024_Demo/raw/main/Demo.mp4`.

7 Discussion

This study aimed to assess the potential of implementing the framework for robotic applications in the context of tower cranes (TCs) and construction sites. While no direct comparison was made with other TC automation techniques, the research emphasises leveraging ROS's path/trajectory planning algorithms and programming frameworks for TC automation. The successful performance in path planning and trajectory execution showcased in the video supports the viability of our proposed framework.

The assumption of non-stretchable cable and rigid tower cranes is ideal, but these elements could experience stretch or deformation. This assumption simplifies the mathematical modelling and analysis of the system. The proposed framework remains valid for several reasons. The cable stretch and tower crane deformation are relatively smaller compared to the scale of the TC structure and can often be neglected. Control strategies and algorithms are robust and adaptive, enabling them to compensate for these minor deviations. Future research will incorporate cable stretch and tower crane deformation into the model. This can be achieved through advanced mathematical models or sensor-based feedback control strategies. This will enhance the accuracy and reliability of the proposed framework.

Gazebo does not directly support flexible or non-rigid structures. The flexibility of TC's structure and rope can be represented as a link of multiple prismatic (linear) and revolute joints, allowing for near-to-real dynamic behaviour but significantly increasing the computation load on physics simulation. Future research will focus on quantifying or estimating the degree of error in both approaches. Regarding payload orientation, using double cables for the hook (reeving) and spreaders for longer payloads naturally dampens twisting oscillations, leading to the assumption that the payload orientation follows the jib's orientation.

This framework is designed to be modular and scalable, allowing for easy customisation and adaptation to different crane models. The control algorithms and strategies can be adjusted based on specific crane characteristics, operational requirements, and dynamic models. However, scaling and customising the system may pose challenges, requiring extensive testing, validation, and additional sensors or hardware components. To tackle these challenges, we propose comprehensive testing and validation processes, compatibility with various sensors and hardware components, and developing advanced control strategies and algorithms to enhance applicability across the construction industry.

While the Robot Operating System (ROS) is widely used, alternative robotics frameworks such as PyRobot, Orca, Yet Another Robot Platform (YARP), Mobile Robot Programming Toolkit (MRPT), Robotics Library (RL), and Dartsim/Dart also offer valuable tools. However, these alternatives often specialise in specific areas or need more active development and community adaptation. For instance, PyRobot is an excellent choice for research or education, while Orca, despite its suitability for embedded systems, has not been developed in recent years. YARP primarily provides robust communication channels among various robot components. MRPT is tailored towards mobile robots, and RL exclusively offers a C++ API. These specialisations and limitations should be considered when selecting a robotics framework for a particular application.

8 Conclusion

This paper presents a framework for planning the collision-free path, jerk-free trajectory and control of the jib, trolley and hoisting mechanism to follow a planned path using sensory input based on ROS and MoveIt. The developed framework is validated on a simulated real-scaled tower crane in Gazebo. Unlike several other path-planning and controlling approaches, we aim to validate our methods on a real-scaled tower crane, which poses different challenges. The developed framework is scaleable to a real system as far as the desired data types are provided to ROS framework, for example, occupancy grid of surroundings and all TC states and their first derivatives in real-time. In the future, we will scale our framework to a scaled-down version of a tower crane while incorporating more realistic parameters like joint frictions, TC body deformation and cable flexibility.

References

[1] C. Huang, C.K. Wong, and C.M. Tam. Optimization of tower crane and material supply locations in a high-rise building site by mixed-integer linear programming. *Automation in Construction*, 20(5):571–580, August 2011. doi:10.1016/j.autcon.2010.11.023.

[2] C. M. Tam and Thomas K. L. Tong. GA-ANN model for optimizing the locations of tower crane and supply points for high-rise public housing construction. *Construction Management and Economics*, 21(3):257–266, April 2003. doi:10.1080/0144619032000049665.

[3] Yehiel Rosenfeld and Aviad Shapira. Automation of existing tower cranes: economic and technological feasibility. *Automation in Construction*, 7(4):285–298, May 1998. doi:10.1016/S0926-5805(98)00049-1.

[4] Songbo Hu, Yihai Fang, and Yu Bai. Automation and optimization in crane lift planning: A critical review. *Advanced Engineering Informatics*, 49:101346, August 2021. doi:10.1016/j.aei.2021.101346.

[5] Menghua Zhang and Xingjian Jing. Adaptive Neural Network Tracking Control for Double-Pendulum Tower Crane Systems With Nonideal Inputs. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 52(4):2514–2530, April 2022. doi:10.1109/TSMC.2020.3048722.

[6] G. Rigatos, M. Abbaszadeh, and J. Pomares. Nonlinear optimal control for the 4-DOF underactuated robotic tower crane. *Autonomous Intelligent Systems*, 2(1):21, August 2022. doi:10.1007/s43684-022-00040-4.

[7] Šandor Ileš, Jadranko Matuško, and Mircea Lazar. Piece-wise ellipsoidal set-based model predictive control of linear parameter varying systems with application to a tower crane. *Asian Journal of Control*, 23(3):1324–1339, May 2021. doi:10.1002/asjc.2515.

[8] Lobna T Aboserre and Ayman A El-Badawy. Robust integral sliding mode control of tower cranes. *Journal of Vibration and Control*, 27(9-10):1171–1183, May 2021. doi:10.1177/1077546320938183.

[9] Zheng Tian, Lili Yu, Huimin Ouyang, and Guangming Zhang. Transportation and swing reduction for double-pendulum tower cranes using partial enhanced-coupling nonlinear controller with initial saturation. *ISA Transactions*, 112:122–136, June 2021. doi:10.1016/j.isatra.2020.11.028.

[10] Tong Yang, Ning Sun, He Chen, and Yongchun Fang. Neural Network-Based Adaptive Anti-swing Control of an Underactuated Ship-Mounted Crane With Roll Motions and Input Dead Zones. *IEEE Transactions on Neural Networks and Learning Systems*, 31(3):901–914, March 2020. doi:10.1109/TNNLS.2019.2910580.

[11] Pham Van Trieu, Hoang Manh Cuong, Hoang Quoc Dong, Nguyen Huu Tuan, and Le Anh Tuan. Adaptive fractional-order fast terminal sliding mode with fault-tolerant control for underactuated mechanical systems: Application to tower cranes. *Automation in Construction*, 123:103533, March 2021. doi:10.1016/j.autcon.2020.103533.

[12] Yiming Wu, Ning Sun, He Chen, and Yongchun Fang. Adaptive Output Feedback Control for 5-DOF Varying-Cable-Length Tower Cranes With Cargo Mass Estimation. *IEEE Transactions on Industrial Informatics*, 17(4):2453–2464, April 2021. doi:10.1109/TII.2020.3006179.

[13] Huimin Ouyang, Zheng Tian, Lili Yu, and Guangming Zhang. Adaptive tracking controller design for double-pendulum tower cranes. *Mechanism and Machine Theory*, 153:103980, November 2020. doi:10.1016/j.mechmachtheory.2020.103980.

[14] Anh Tuan Le and Soon-Geul Lee. 3D cooperative control of tower cranes using robust adaptive techniques. *Journal of the Franklin Institute*, 354(18):8333–8357, December 2017. doi:10.1016/j.jfranklin.2017.10.026.

[15] Menghua Zhang, Yongfeng Zhang, Bing Ji, Changhui Ma, and Xingong Cheng. Modeling and energy-based sway reduction control for tower crane systems with double-pendulum and spherical-pendulum effects. *Measurement and Control*, 53(1-2):141–150, January 2020. doi:10.1177/0020294019877492.

[16] Huai-Tao Shi, Jian-Qi Huang, Xiaotian Bai, Xiang Huang, and Jie Sun. Nonlinear Anti-swing Control of Underactuated Tower Crane Based on Improved Energy Function. *International Journal of Control, Automation and Systems*, 19(12):3967–3982, December 2021. doi:10.1007/s12555-020-0292-1.

[17] Dongfeng He, Junping He, Zhibin Li, Bo Huang, and Hongliang Cheng. An anti-collision system for tower cranes considering lifting heavy objects. *Journal of Physics: Conference Series*, 2365 (1):012019, November 2022. doi:10.1088/1742-6596/2365/1/012019.

[18] Malak Al Hattab, Emile Zankoul, and Farook R. Hamzeh. Near-Real-Time Optimization of Overlapping Tower Crane Operations: A Model and Case Study. *Journal of Computing in Civil Engineering*, 31(4):05017001, July 2017. doi:10.1061/(ASCE)CP.1943-5487.0000666.

[19] Jiahao Tian, Sang Luo, Xinming Wang, Jing Hu, and Jun Yin. Crane Lifting Optimization and Construction Monitoring in Steel Bridge Construction Project Based on BIM and UAV. *Advances in Civil Engineering*, 2021:1–15, July 2021. doi:10.1155/2021/5512229.

[20] Souravik Dutta, Yiyu Cai, Lihui Huang, and Jianmin Zheng. Automatic re-planning of lifting paths for robotized tower cranes in dynamic BIM environments. *Automation in Construction*, 110:102998, February 2020. doi:10.1016/j.autcon.2019.102998.

[21] Wenyu Zhang, Hong Zhang, and Lu Yu. Collaborative Planning for Stacking and Installation of Prefabricated Building Components Regarding Crane-Collision Avoidance. *Journal of Construction Engineering and Management*, 149(6):04023029, June 2023. doi:10.1061/JCEMD4.COENG-12955.

[22] Leon C. Price, Jingdao Chen, Jisoo Park, and Yong K. Cho. Multisensor-driven real-time crane monitoring system for blind lift operations: Lessons learned from a case study. *Automation in Construction*, 124:103552, April 2021. doi:10.1016/j.autcon.2021.103552.

[23] Shih-Chung Jessy Kang. Physics based model for simulating the dynamics of tower cranes. 2004. URL `https://api.semanticscholar.org/CorpusID:210718033`.

[24] N. Koenig and A. Howard. Design and use paradigms for gazebo, an open-source multi-robot simulator. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, volume 3, pages 2149–2154, Sendai, Japan, 2004. IEEE. ISBN 9780780384637. doi:10.1109/IROS.2004.1389727.

[25] Hanafy M Omar and Ali H Nayfeh. Gain Scheduling Feedback Control of Tower Cranes with Friction Compensation. *Journal of Vibration and Control*, 10(2):269–289, February 2004. doi:10.1177/1077546304035610.

[26] Sachin Chitta, Ioan Sucan, and Steve Cousins. Moveit! [ros topics]. *IEEE Robotics & Automation Magazine*, 19(1):18–19, 2012.