SIMULATING THE TIME-VARYING PARAMETERS OF ROBOTS IN PERFORMING THE COMPLEX SEQUENTIAL TASKS

M. Iqbal[1], R. A. M. Qureshi[2], G. Abbas[3*],

[1]Department of Industrial and Manufacturing Engineering, University of Engineering &Technology Lahore, Pakistan
[2]Department of Physics, Riphah International University Lahore, Pakistan
[3]Riphah College of Veterinary Sciences, Riphah International University Lahore, Pakistan
*Corresponding Authors: muzmmaliqbal346@gmail.com

ABSTRACT: To improve the performance in complex sequential chores, simulating time-varying parameters in robotic systems is essential. This review paper explores an advanced computational framework for modeling and analyzing the dynamic parameters of robots like position, torque, velocity, and force during sequential operations. Time-varying factors play a key role in defining the accuracy and efficiency of robotic tasks, specifically in environments where tasks are multi-stage, subject to changing conditions, and non-repetitive. The recommended simulation model incorporates important techniques including kinematic modeling, adaptive control algorithms, and nonlinear dynamic equations of motion to deliver real-time apprises on robot performance under joint friction, external disturbances, and varying loads. A multi-parameter time-series approach is utilized to simulate the unceasing interaction between robotic systems and their working environments. The model put on finite element analysis to simulate machine-driven deformation, and stress confirming the consistency of the robot's structure during task execution. The review also includes reinforcement learning to let robots self-improve in real-time as tasks progress, adjusting to unexpected variables like terrain changes, task priorities, and fluctuating payloads. The dynamic task scheduling is controlled by Markov decision processes which allow well-organized switching between tasks whilst minimizing resource consumption and downtime. An inverse dynamics approach is engaged to compute actuator forces and joint torques essential for the execution of the wanted movements, allowing for real-time adjustments in speed and trajectory. To enhance the simulation's fidelity, sensor fusion procedures are applied, joining data from multiple/compound sensors e.g., gyroscopes, force/torque sensors, and cameras, etc. to deal with widespread feedback on the robot's interface with its environment. This feedback is managed by Kalman filters to alleviate noise and offer correct apprises to the control system. Experiments results conducted on mobile robots and industrial robots performing tasks such as object manipulation, navigation, and assembly line operations through dynamic environments demonstrate that the simulated robots can adjust to time-varying factors with greater accuracy and less error margins, helping improved operational robustness and task efficiency. The usage of trajectory optimization algorithms in the simulation has shown a noteworthy decrease in wear on robot joints and energy consumption by smoothing out motion paths and preventing sudden changes in movement. Therefore, this review focuses on the presentation of a healthy simulation framework that efficiently adapts and models to the time-varying parameters of robots during complex sequential tasks.

Keywords: Simulating; time-varying parameters; robots; complex sequential tasks; accuracy; efficiency

INTRODUCTION

Accurate modeling of time-varying parameters like joint torques, accelerations, and velocities is crucial for creating accurate robotic simulations that faithfully mimic real-world behavior (Spong *et al*., 2006). These parameters impact the robot's accuracy, efficiency, and security during task performance, letting developers envisage and mitigate probable issues before deployment (Liu and Chen, 2019). Control systems as well as optimization methods improve the practicality and functionality of robotic models, confirming robots can become accustomed to dynamic environments with higher accuracy and safety. Machine-driven wear and tear are serious factors in the long-standing performance and trustworthiness of robots. Techniques like predictive maintenance and adaptive control help outspread the robot's working lifespan and increase efficiency and safety in long-term deployments. Time-varying parameters also affect a robot's adaptability to unpredictable and complex task sequences, augmenting the reliability and flexibility in complex and unpredictable applications.

Managing sequential dependencies between tasks requires tactics i.e. hierarchical task planning, pre- and post-condition modeling, reinforcement learning, task dependency graphs, dynamic task error recovery mechanisms, and scheduling. These methods confirm reliable and efficient action in complex, changeable environments. Advanced control approaches, like adaptive control, Model Predictive Control (MPC), robust control, state estimation, dynamic modeling, and feed-forward control maintain constancy despite parameter variations. Control algorithms that adapt to dynamic changes in system parameters are essential for real-time robotic systems. Robots use sensor feedback, adaptive control, control systems, machine learning algorithms, and fault detection systems to sense and react to deviancies in time-varying parameters. These mechanisms ensure efficiency, safety, and precision during task completion. Robotic simulations account for unpredictable or extreme differences in time-varying parameters, and time-varying parameter models can scale through diverse platforms and tasks. Simulation authentication for robot time-varying parameters includes physical model testing, real-world data assessment, scenario testing, sensitivity analysis, machine learning, benchmarking, and real-time feedback. The accomplishment of a robot's performance in handling serial tasks with time-varying parameters can be evaluated through metrics like accuracy, task completion time, energy consumption, adaptability, and safety. Joint torque is the rotational force exerted on a robot's joints during movement, influenced by factors like configuration, payload, and frictional forces. Robotic simulations use dynamic models, such as Newtonian mechanics or the Lagrangian approach, to accurately model joint torques. Real-time adjustments to joint torque are required, often using PID control systems, to ensure safe operational limits in safety-critical applications like manufacturing and healthcare. Accelerations and velocities are key parameters in robot motion, used in simulations to accurately model their speed and rate of change. These parameters are crucial for high-speed robotic applications, such as pick-and-place tasks in industrial settings. Realistic simulations ensure constant monitoring and adjustment of these parameters to maintain stability and precision. Excessive acceleration can lead to mechanical wear and tear, making it essential to model limits to prevent damage or operational inefficiency. Robotic simulations face challenges in continuously adjusting time-varying parameters during task execution. Real-time feedback and control systems, like MPC, optimize joint torques, velocities, and accelerations. These methods minimize energy consumption, improve task efficiency, and extend the robot's lifespan. By integrating real-time adaptation and optimization, developers can fine-tune parameters for safer and more efficient robot operation.

Methods for Accounting for Mechanical Wear and Tear: To maintain the efficiency and longevity of machinery/robots methods for accounting for mechanical wear and tear play a critical role. One of the common methods used is depreciation accounting**,** where the cost of equipment is gradually written off over its useful life, reflecting the predictable wear and tear. Techniques like accelerated depreciation **or** straight-line depreciation methods, i.e**.** double-declining balance, can be applied depending on how speedily the equipment is estimated to damage. Moreover, condition-based monitoring is assisted by regularly judging the machinery's efficacy/performance and/or condition to envisage whether mechanical wear or repair substitutes are desired. Predictive maintenance influences sensors to forecast and data analytics, leading to timely interferences. This arrangement of proactive maintenance strategies and financial accounting warrants that wear and tear are systematically mitigated and tracked (See Table 1).

Degradation Models: One common approach to account for wear and tear is through degradation models, which simulate the progressive deterioration of mechanical components. These models incorporate the effects of fatigue, friction, and material degradation into the simulation. For instance, fatigue-based models use historical data on component failure rates to predict the lifespan and efficiency loss of parts such as motors and gears (Jardine *et al*., 2006). These models allow engineers to estimate how much wear a robot's joints or actuators can withstand before performance begins to deteriorate significantly.

Friction and Stiction Models: As mechanical components degrade, frictional forces within joints and between moving parts tend to increase, which directly impacts the accuracy of a robot's movements. Friction models, such as the Coulomb and viscous friction models, simulate the increase in resistance over time due to wear. Similarly, stiction (static friction) models account for the higher resistance encountered when starting motion from rest. These friction models are integrated into the robot's dynamic equations to adjust parameters like joint torque and velocity to reflect the additional energy required to overcome increased friction (Canudas-de-Wit *et al*., 1995).

Component Health Monitoring: Some systems incorporate sensors to monitor the health of critical components, such as encoders that track the wear on bearings or strain gauges that measure load and stress on joints. This real-time data is used to update the robot's dynamic model and adapt to changing performance capabilities. For instance, as the wear increases, control algorithms can adjust joint torques or velocities to maintain consistent performance (Mobley, 2002). By integrating component health data into simulations, parameter variations due to wear can be accurately predicted and managed over time (See Table 1).

Table 1: Different methods for accounting for mechanical wear and tear in robots

Method	References	Advantages	Description	Limitations
Preventive Maintenance	Hongyan *et al*., 2023	Decreases sudden downtime and breakdowns.	Scheduled maintenance activities based on time or usage to avoid failures.	This may lead to over-maintenance and increased costs.
Lubricant Analysis	Acar *et al*., 2020	Detect initial signs of wear without disassembling components.	Examines lubricant for contamination by wear particles to judge the state of mechanical components.	Needs recurrent sampling and analysis; equipment-specific.
Condition-Based Monitoring (CBM)	Uhlmann *et al*., 2020	Minimizes useless repairs; lessens downtime.	Checking real-time data to determine equipment situation and maintenance requirements.	Requires sensors and continuous monitoring systems.
Thermal Imaging	Lintvedto., 2023	Non-invasive and offers quick results.	Uses infrared cameras to sense abnormal heat arrays that show friction and wear in mechanical components.	Needs specialized equipment and knowledge for interpretation.
Vibration Analysis	Jianlong *et al*., 2022	Effective for an initial finding of mechanical problems.	Measures vibrations in components to detect signs of wear, misalignment, or imbalance.	Requires expert analysis and can be sensitive to noise.
Predictive Maintenance (PdM)	Pookkuttath *et al*., 2022	Reduces unplanned downtimes and optimizes resource usage.	Utilizes data analytics and machine learning to predict when failures may occur, allowing for optimized maintenance.	Requires extensive data collection and analysis capabilities.
Life Cycle Assessment (LCA)	Stuhlenmiller *et al*., 2021	Offers widespread long-term insights into robot longevity.	Evaluate the environmental impact and wear over the entire life cycle of the robot, from manufacturing to disposal.	Complex to implement and may need adjustments over time.
Digital Twin	Yao *et al*., 2023	Let real-time checking and forecasting for proactive maintenance.	An effective replica of the robot that simulates its wear and tear over time, guessing failures and maintenance needs.	Complexity and high implementation cost.

Integration into Parameter Variation: Adaptive control techniques are used to integrate wear and tear into a robot's parameter variation model. These techniques regulate control parameters built on real-time feedback and system dynamics, reimbursing for decreased efficiency. Predicting maintenance algorithms utilize sensor data to envisage component failure, leading to maintenance scheduling before significant performance breakdown or loss. Probabilistic models, like the Bayesian framework, capture ambiguity in wear development, permitting more accurate predictions of when wear will significantly affect the robot's operations.

Task Complexity and Sequential Planning: Time-varying parameters such as joint torques, accelerations, velocities, and external forces play a vital role in defining a robot's capability to adjust to unpredictable and complex task sequences. These parameters dynamically change during a robot's operation, particularly when it interrelates with changing environments or performs multi-step tasks (See Table 2). How well a robot can grip these time-varying situations directly affects its adaptability, overall performance, and precision in real-time, especially in tasks that need a high degree of flexibility and responsiveness. Robotics are equipped with various parameters to ensure their performance in various tasks. Joint torque, a crucial parameter, determines the force applied by each joint to perform tasks. In unpredictable environments, the robot must adjust its torque to maintain stability and accuracy. Adaptive controllers like Model Reference Adaptive Control (MRAC) can modify torque in real time based on task demands, enabling the robot to dynamically handle unpredictable tasks. Velocity is another critical parameter, affecting the robot's ability to adapt. It must vary its speed in response to task complexity or environmental conditions. Time-varying velocity control allows robots to switch between high-speed and high-precision modes based on the task sequence. Failure to adapt velocity can result in inefficient operation or errors in delicate procedures, negatively impacting the robot's overall task adaptability. Acceleration also plays a role, directly impacting a robot's ability to maintain balance and stability while performing complex tasks. Time-varying control of acceleration ensures the robot adapts fluidly to the unpredictable aspects of the task, maintaining safety and accuracy.

Role in Handling Unpredictable Task Sequences: Time-varying parameters in robots empower them to handle random task sequences, letting them adjust dynamically to innovative information. These considerations are essential in dynamic environments, specifically in human-robot collaboration, where the robot's capability to adjust its behavior in answer to changeable inputs is needed for safe and effectual cooperation. The capability to cope with successive needs between tasks is critical in robotics and further fields where composite, multi-step operations are completed. Careful planning, as well as performance, are compulsory to confirm each task is completed in the correct order and any variations in one task are accounted for in subsequent ones. Several strategies are employed to optimize efficiency and reliability, often used in tandem.

Hierarchical Task Planning: Hierarchical task planning is a common strategy for breaking down tasks into sub-tasks with dependencies mapped in a top-down structure. This method breaks down high-level goals into manageable actions, organized according to their dependencies. For example, in assembly-line robots, each stage depends on the previous one's completion. This helps robots or systems execute steps correctly, manage dependencies, and trace and adjust sequences based on outcomes from earlier tasks.

Pre- and Post-Condition Modeling: Pre- and post-condition modeling is a technique that defines the conditions required before and after a task, ensuring that outcomes match prerequisites for the next task. This approach helps dynamically assess task readiness, preventing premature execution and potential failure. It ensures logical and accurate handling of dependencies, allowing early detection of errors or inconsistencies. For example, in industrial robotics, successful positioning of a part requires pre-condition satisfaction for task B to begin.

Task Dependency Graphs: A task dependency graph makes it simple to identify essential paths and possible bottlenecks by providing a visual depiction of the sequential links between jobs. It is especially helpful in intricate systems like driverless cars, where there are several interdependent jobs, such as city navigation. This organized representation guarantees effective functioning and aids in the management of related duties.

Reinforcement Learning for Sequential Tasks: Reinforcement learning (RL) is a strategy that helps robots and systems learn the optimal sequence of actions through trial and error. It adapts to changing environments and handles dependencies dynamically. RL is particularly effective when dependencies between tasks are not explicitly known or highly variable. For example, in robotic manipulation, RL optimizes the robot's performance by continuously learning from task outcomes, ensuring successful task completion and optimal performance.

Dynamic Task Scheduling: Dynamic task scheduling is a real-time method that adjusts task execution based on the system's state and environment, ensuring system efficiency. It's crucial for tasks with varying durations or unexpected delays. In human-robot collaboration, helps robots adjust task sequences based on worker pace or material availability, ensuring flexibility and responsiveness to complex task dependencies.

Error Recovery Mechanisms: Error recovery mechanisms are crucial for managing sequential task dependencies. They allow a system to retry, adjust subsequent tasks, or recover from errors without disrupting the overall sequence. This mitigates risks associated with task dependencies and prevents cascading failures. For instance, in surgical robots, error recovery mechanisms compensate for deviations in outcomes by adjusting subsequent actions to compensate for the variation.

Stability and Control: Ensuring system stability when time-varying parameters, such as mass or center of gravity (CoG), change during a task is a crucial aspect of robotic control and dynamic systems. These parameters can shift due to various factors like load variations, environmental interactions, or internal changes within the system itself. Managing such changes requires advanced control strategies to prevent instability and ensure the robot can continue to operate safely and effectively.

Here are several approaches used to ensure stability in the presence of time-varying parameters:

Adaptive Control: Adaptive control is a real-time method used to handle system parameters like mass and CoG in industrial robots. It continuously monitors performance and adjusts control laws to compensate for changes. Adaptive controllers adjust torque and force inputs based on changes in mass and CoG, maintaining stable operation. This method is particularly useful in robotic manipulators where mass distribution can shift.

Table 2: Role of Time-Varying Parameters in Handling Unpredictable Task Sequences.

Time-Varying Parameter	Researchers	Explanation	Role in Handling Unpredictable Task Sequences
Speed Adaptation	Galvan-Perez *et al*. (2023)	Robots can slow down for complex, delicate tasks and speed up for simpler, urgent tasks to optimize performance.	Adjusts speed dynamically to respond to changes in task difficulty or urgency.
Force Control	MacDonald *et al*. (2024)	Robots adapt their grip force when handling fragile or complex items, increasing safety and precision.	Modifies applied force based on the object's properties and task requirements.
Learning Rate in ML Algorithms	Wang, *et al*. (2015)	Time-varying learning rates in machine learning algorithms allow robots to learn faster or slower, based on the task dynamics.	Changes the learning rate to enable quicker adaptation to task variability.
Sensor Fusion Weighting	Zhang & Wei, (2013)	When critical sensors (e.g., vision or touch) become more important in a task, robots give higher weight to their inputs.	Dynamically adjusts the importance of sensory input based on environmental changes.
Task Prioritization	Somani *et al*. (2016)	Robots can reorder tasks dynamically based on emerging needs or changes in task sequences, ensuring critical tasks are prioritized.	Changes task priority in real-time, depending on the task or environment.
Energy Management	Nonoyama *et al*. (2022)	Robots can conserve energy during low-intensity tasks and increase output during high-demand tasks to ensure efficiency.	Regulates energy use to balance consumption and performance in varying tasks.
Control Gains	Yue *et al*., (2024)	Modifying control gains dynamically helps robots remain stable when faced with unexpected disturbances or changes in tasks.	Adjusts control parameters to maintain stability during unpredictable conditions.

Robust Control: Robust control is a method that ensures stability despite uncertainties or variations in system parameters. It handles a predefined range of parameter variations without real-time updates, ensuring stability even when parameters like mass or CoG change. This technique is commonly applied in systems with predictable time-varying parameters, such as autonomous vehicles or drones navigating dynamic environments.

MPC: MPC is a strategy that maintains system stability in complex, multi-task operations. It predicts future system behavior based on a dynamic model and adjusts control actions accordingly. MPC is particularly useful in robotic arms, where parameter variations are frequent. By continuously updating its model, MPC calculates optimal control actions to stabilize the robot's movement amidst these changes.

State Estimation Techniques: Accurate state estimation is crucial for maintaining stability in time-varying parameters. Techniques like the Kalman filter or extended Kalman filter (EKF) estimate internal states like mass and CoG, allowing real-time control strategies. For legged robots, state estimation helps predict and adjust posture to maintain balance and stability during movement, ensuring stability.

Feedforward Control: Feedforward control is a method that anticipates changes in time-varying parameters and adjusts the system's control actions accordingly. It uses predictive models to proactively adjust control inputs, allowing for system stabilization before instability occurs. For example, in robotic arms, feed-forward control can adjust motor torques in anticipation of mass changes, preventing instability or oscillations.

Dynamic Modeling and Parameter Identification: Accurate dynamic models, including mass, CoG, joint torques, and external forces, are crucial for robot stability. Continuous parameter identification methods update the robot's internal model in real time, allowing the control system to anticipate and correct potential instability. For surgical robots, continuous monitoring and model updating ensure precise, stable performance by adapting to time-varying forces and dynamics.

Fuzzy Logic and Artificial Neural Networks: Fuzzy logic and artificial neural networks (ANNs) are effective in handling non-linear and uncertain time-varying parameters in robotic systems. Fuzzy logic controllers use approximate reasoning, while ANNs learn from experience and adjust control actions based on complex relationships between system states and parameters. These approaches ensure stability by learning and adapting over time.

Energy-based Stability Criteria: In some systems, ensuring stability relies on maintaining a consistent energy balance. Techniques like the Lyapunov stability criterion are used to guarantee that the total energy of the system remains bounded despite variations in time-varying parameters. By designing controllers that ensure the system's energy dissipates or remains within safe limits, these methods provide a mathematical guarantee of stability, even as parameters like mass or CoG fluctuate (Haddad *et al*., 2006).

4- Optimization and Performance: Optimizing time-varying parameters in robots, such as joint torques, velocities, and accelerations, is crucial for minimizing task completion time while ensuring energy efficiency and precision. Achieving this balance involves a combination of real-time control algorithms, advanced optimization techniques, and physical models of the robot's dynamics. Here's how these time-varying parameters are optimized:

Trajectory Optimization: One of the most effective methods for minimizing task completion time and optimizing energy efficiency is trajectory optimization. This involves determining the optimal path that the robot should follow to complete a task while accounting for the robot's kinematic and dynamic constraints. The time-optimal Trajectory Planning method minimizes the time required to move from one point to another while respecting the robot’s dynamic limits, such as maximum velocity and acceleration. Time-optimal trajectories push the robot's joints and actuators to their limits, reducing overall task time but still adhering to safety constraints (Bry & Roy, 2011). The energy-efficient Trajectory Planning approach minimizes energy consumption during the task. Energy-efficient trajectories avoid high-speed or abrupt movements, which consume more power. Optimizing energy can involve selecting paths that reduce joint torques or resistive forces (Knöchelmann *et al*., 2020). In industrial robots, optimal joint trajectories are calculated by minimizing a cost function that combines time, energy, and torque constraints. The goal is to maintain precise movements while completing tasks as quickly and efficiently as possible.

MPC: MPC is a powerful real-time control algorithm that optimizes time-varying parameters dynamically. MPC uses a predictive model of the robot's dynamics to calculate the optimal control inputs at each time step, minimizing task time and energy consumption. MPC continuously predicts the future states of the robot based on its current time-varying parameters (such as position and velocity) and adjusts the control inputs to minimize task completion time. MPC can incorporate energy consumption as part of the optimization problem by adding energy-related terms to the cost function. For example, it can prioritize minimizing the integral of power consumption over the tasking horizon (Qin & Badgwell, 2003) i.e. in mobile robots, MPC can be used to optimize paths and speed while considering battery consumption, minimizing both task time and energy usage Braunl, (2012).

Multi-Objective Optimization: Multi-objective optimization techniques address the trade-off between minimizing task completion time, energy efficiency, and precision by optimizing multiple criteria simultaneously. These methods use Pareto frontiers to explore the best balance between conflicting objectives. Multi-objective optimization generates a set of solutions (Pareto-optimal solutions), where improving one objective (e.g., task completion time) results in sacrificing another (e.g., energy consumption or precision). This allows the operator or control system to choose the most appropriate solution based on current operational requirements (Deb *et al*., 2002). Genetic algorithms are often used in multi-objective optimization, as they can explore a large search space and provide a set of optimal solutions by mimicking biological evolution (Srinivas & Deb, 1994), i.e. in robotic surgery, multi-objective optimization is used to ensure precision in tool movements while minimizing energy consumption and time spent during a procedure.

Adaptive Control and Learning-Based Methods: Adaptive control systems continuously adjust the robot's parameters based on real-time feedback to optimize performance for specific tasks. These systems can account for changes in the robot's dynamics, environment, or task complexity. Adaptive controllers adjust joint torques and velocities in real-time, learning from the task's execution to reduce energy consumption. These systems can optimize movements by reducing unnecessary accelerations and decelerations. Reinforcement learning algorithms allow robots to learn optimal strategies for minimizing task completion time and energy use through trial and error. These algorithms adjust control parameters based on feedback from the environment, allowing robots to improve efficiency over time (Kober *et al*., 2013), i.e. in manufacturing, adaptive controllers can dynamically adjust a robotic arm’s speed and force to minimize energy use without sacrificing precision (**Xu, 2007)**.

Dynamic Programming: Dynamic programming is an optimization method that divides a complex task into smaller sub-problems and solves each one optimally (Xu, 2007**)**. It’s particularly useful for optimizing time-varying parameters over a sequence of movements or tasks. Bellman’s Principle of Optimality principle is applied in dynamic programming, where the overall task is optimized by finding the best decisions at each step. The robot’s joint angles, velocities, and accelerations are optimized to reduce task time and energy use step-by-step (Bellman, 1957). Dynamic programming can be used to solve optimal control problems where the robot’s entire movement is planned to minimize energy consumption and time across all steps of the task, i.e. in robotic pick-and-place tasks, dynamic programming can be used to optimize joint angles and velocities to ensure the fastest possible movement with minimal energy use.

Fuzzy Logic and Heuristics: Fuzzy logic and heuristic methods can be used to fine-tune time-varying parameters when dealing with uncertain or imprecise conditions. These methods are particularly useful when it's difficult to model the system exactly or when human-like reasoning is required. Fuzzy logic controllers make decisions based on imprecise or fuzzy inputs (e.g., approximate positions or velocities). These controllers can adjust the robot's parameters to balance between speed, energy efficiency, and precision, especially in unstructured environments (Zadeh, 1965). Heuristic algorithms, such as particle swarm optimization, are often used to find near-optimal solutions quickly. These algorithms are effective in high-dimensional problems, such as optimizing time-varying parameters across multiple joints and tasks, i.e. in healthcare robotics, fuzzy logic may be used to control a robot's joint angles and force to maintain both speed and safety when interacting with patients.

Task-Specific Optimizations: In some cases, robots perform highly specialized tasks that require fine-tuning of specific time-varying parameters. For example, a welding robot needs to adjust its path, speed, and heat application for precise and efficient welds. In tasks such as robotic painting or welding, optimizing the robot's path and velocity in real time ensures both precision and energy efficiency. The system continuously adjusts the speed and angle of the tool to achieve the best coverage with the least energy expenditure, i.e. in robotic painting, optimizing the speed and path of the arm minimizes paint waste, energy consumption, and task completion time.

Learning and Adaptability: Machine learning techniques can be used to predict and optimize time-varying parameters in complex tasks by leveraging vast amounts of real-time data generated by robots. Reinforcement learning (RL) is a key approach, where robots learn optimal actions through trial and error, receiving feedback in the form of rewards or penalties based on their performance. This allows the robot to adjust parameters dynamically to changing conditions or unexpected challenges. Supervised learning uses historical data from past tasks to train models that predict how time-varying parameters will evolve under different conditions. Deep learning algorithms analyze large datasets to identify patterns in the robot's performance, allowing for real-time optimization of parameters. Transfer learning allows robots to apply knowledge gained from one task to optimize performance in related tasks, reducing training time. By integrating these machine learning techniques, robots can continuously optimize their performance, reduce errors, and adapt their behavior in complex, dynamic environments. Imitation learning allows robots to learn from expert demonstrations, adjusting their parameters based on successful task completions and adapting to new tasks that share similarities with past experiences. Transfer learning further extends this by enabling robots to transfer knowledge from one task to another, allowing them to adjust their parameters more efficiently in new but related tasks.

Error Handling and Recovery: When task errors or interruptions occur, robots adjust their time-varying parameters, such as joint torques, velocities, and accelerations, using a combination of real-time feedback control, adaptive control algorithms, and error recovery strategies. These mechanisms allow the robot to adapt and compensate for deviations from the desired task trajectory. Robots often use feedback control loops, such as Proportional-Integral-Derivative (PID) controllers, to continuously monitor task execution. If an error or interruption is detected, the controller adjusts parameters like velocity or torque to correct the error in real-time. For instance, if a robot's arm encounters unexpected resistance while moving, the feedback system will modify the applied torque to maintain stability and prevent damage (Franklin & Powell, 2014). In more complex environments, adaptive control is used to handle changes in system dynamics caused by errors or task interruptions. Adaptive control systems can automatically tune the robot's parameters by estimating unknown variables, such as changes in the robot's mass or center of gravity, during the task. This helps the robot to adapt on the fly and continue the task despite disturbances (Ioannou & Sun, 2012). MPC systems use a dynamic model of the robot to predict future states based on current conditions. When errors or interruptions occur, MPC recalculates the optimal control inputs to adjust the time-varying parameters and guide the robot back to the desired trajectory, minimizing delays or further errors (Mayne, 2014). Machine Learning-Based Adjustment: In advanced systems, machine learning algorithms may be employed to predict the occurrence of errors and adjust parameters accordingly. For example, reinforcement learning algorithms can learn from past task executions, enabling the robot to autonomously adjust parameters when similar errors occur in the future (Kober *et al*., 2013). In this way, the robot gradually improves its resilience to interruptions and errors. In all these methods, ensuring the system's stability while adjusting parameters is critical. This involves balancing speed and precision while also considering safety factors, particularly in tasks where human-robot collaboration is involved.

Communication and Data Fusion: Multiple sensors in systems provide feedback, requiring data fusion from various sources. Techniques like Kalman filtering combine data from sensors to provide more accurate estimates. Sensor data is crucial for adjusting parameters during task execution, ensuring optimal performance and safety in dynamic environments. Sensors continuously monitor internal and external conditions, allowing real-time adjustments to key parameters, enhancing precision, adapting to changes, and maintaining system stability. Sensors like encoders, accelerometers, and gyroscopes provide continuous real-time data about the robot's positions, velocities, and forces. This data allows the control system to adjust parameters to ensure that the robot follows the desired trajectory and meets performance goals. For example, joint torque sensors help regulate the force applied to specific components, ensuring smooth motion even in unpredictable environments (Mayne, 2014). Sensor data enables the detection of errors or deviations from expected behaviors during task execution. For example, if the robot encounters unexpected resistance, force sensors detect this, and the control system adjusts torque to prevent mechanical damage or task failure. Control algorithms like PID (Proportional-Integral-Derivative) or MPC use sensor data to minimize errors and correct deviations (Chinnappa, 2023). In complex environments, adaptive control strategies based on sensor feedback allow robots to adjust to time-varying parameters in real time. For instance, when lifting objects of varying weight, force sensors measure the load, and the robot adapts its grip strength accordingly to avoid slippage or damage (Craig, 2018). Adaptive control algorithms can modulate the joint velocities and torques dynamically to maintain stability and accuracy in performing tasks (Rudomanenko *et al*., 2021). In human-robot collaboration (HRC) settings, proximity sensors, vision systems, and force sensors play a key role in preventing accidents. Sensor data ensures that the robot maintains safe distances from humans or other obstacles. If sensors detect an impending collision, the robot can instantaneously reduce speed, adjust force, or stop altogether to ensure safety (Navarro *et al*., 2014; Göger *et al*., 2010; Ajoudani *et al*., 2017; Villani *et al*., 2018). Long-term use of robotic systems leads to wear and tear in mechanical components. Sensor data can detect changes in performance due to this wear, such as increased friction in joints or decreased accuracy in motion. By monitoring these deviations, control systems adjust time-varying parameters like torque to compensate for degradation, extending the robot's operational life (Isermann, 2006).

Robustness and Scalability: The robustness of simulations in addressing unpredictable time-varying parameters depends on the quality and comprehensiveness of models, algorithms, and scenarios used. Modern robotic simulations, particularly those involving joint torques, velocities, and accelerations, use advanced methods to account for uncertainties and adapt to unforeseen changes. In simulations where extreme variations in parameters can occur, probabilistic or stochastic models are often employed. These models account for random variations in forces, torques, or environmental factors. By introducing randomness into the simulations, the system's behavior under rare or unpredictable conditions is tested more thoroughly. Monte Carlo simulations, for instance, use a large number of random samples to estimate the effects of unpredictable variations (Tedrake Russ, 2023). Robust simulations use adaptive control strategies that modify the robot's behavior in real-time as variations occur. Algorithms such as MPC or Reinforcement Learning-based controllers adjust the robot's movements and parameters dynamically, allowing the robot to react to unexpected conditions like sudden load changes or shifts in the center of gravity. These adaptive algorithms help the robot handle extreme variations with greater precision and stability (Chen and `11`Astolfi, 2021).

In some scenarios, simulations are designed to stress-test robots by subjecting them to extreme variations in parameters beyond normal operating conditions. These include simulating sudden and unpredictable shifts in joint forces, temperature extremes, or abrupt mechanical changes due to wear and tear. This allows engineers to identify weak points in the system and improve the robot's tolerance for such extreme conditions. Environmental stress testing is crucial for environments like space exploration or hazardous industrial tasks (Milecki and Nowak. 2023). Simulations that integrate real-time sensor data can adapt to extreme changes more effectively. Sensor data allows the simulation to adjust key parameters—such as velocity or acceleration—on the fly. The inclusion of realistic sensor noise or delays in these simulations ensures that the models reflect real-world challenges. In more advanced systems, simulations with real-time feedback loops from sensors can detect outliers in time-varying parameters and compensate in real time, increasing the robustness of the model (Rudomanenko *et al*., 2021). In the event of extreme variations causing subsystem failures (e.g., motor breakdowns or joint malfunctions), fault-tolerant control strategies are built into the simulations. These systems allow the robot to continue functioning by isolating the failure and compensating with operational components. Redundant systems or software-based fault detection algorithms are often included to ensure that even in extreme cases, the system can maintain some level of performance (Isermann, 2006). Simulations are used to test the robustness of robots' time-varying parameter control, exposing them to sudden changes in torque or speed without losing stability. These simulations expose the robot to various tasks and platforms, ensuring it can adjust appropriately without losing stability. The scalability of these models is a complex challenge but can be achieved with appropriate modeling strategies and adaptation mechanisms. Scaling these models requires accounting for differences in the robot's physical structure, task complexity, and environmental factors.

Validation and Testing: Validating simulations is crucial for ensuring robots' time-varying parameters, such as joint torques, velocities, accelerations, and dynamic properties, behave as expected in real-world tasks. This process involves comparison with real-world data, performance evaluation, and systematic testing under varying conditions. Robotic simulations are validated by comparing simulated behavior with real-world experimental data. This involves running the robot through tasks in both real and simulated environments and collecting data on critical parameters like joint angles, velocities, and forces. The simulation is adjusted to minimize discrepancies between simulated and actual values. High-fidelity sensors on the physical robot capture real-time data, comparing predicted outputs to actual behavior. Hardware-in-the-loop (HIL) simulations use physical models to validate a robot's dynamic properties in real-time scenarios. HIL tests a portion of the robot's hardware, such as actuators or sensors, while simulated parts are tested. This hybrid approach identifies any mismatches between simulation assumptions and physical behavior due to factors like mechanical wear or environmental changes. Simulation validation involves sensitivity analysis on time-varying parameters, such as mass, friction, or stiffness, to ensure robustness and prevent minor variations in real-world parameters from drastically altering the robot's behavior, ensuring that the simulation remains robust and accurate. Simulations are conducted in real-world scenarios to accurately predict a robot's behavior across various tasks and conditions. These tests validate the robustness of time-varying parameters like joint acceleration or torque, making the simulation more reliable and generalizable to various real-world applications. This ensures the robot's ability to adapt to changing conditions and handle different tasks.

Machine learning techniques enhance the accuracy of simulation models by adjusting robot parameters based on past experiences or real-world task execution. Over time, these models become more accurate, narrowing the gap between simulated predictions and real-world performance, thus validating simulations. Sensor feedback during task execution is crucial for validating simulations. High-fidelity sensors monitor robot parameters like torque and velocity, providing real-time data for comparison. This ensures deviations between simulated and actual parameters are detected and corrected, improving simulation accuracy, especially in dynamic environments. Simulation validation in industries like healthcare and manufacturing involves benchmarking robot performance against safety and operational standards. For example, FDA regulations require high accuracy in replicating human-like movements, making meeting these benchmarks a critical part of the validation process.

Metrics critical for ensuring the robots operating efficiency: Evaluating the success of a robot's performance with time-varying parameters in sequential tasks relies on a diverse set of metrics that consider efficiency, precision, safety, adaptability, and task completion. These metrics are critical for ensuring that robots operate effectively in dynamic environments where parameters such as joint torques, velocities, and accelerations can vary over time. One of the primary metrics is the time taken to complete a sequence of tasks. The efficiency of the robot is measured by how quickly it can complete tasks without errors, considering the variability in parameters like speed and acceleration. Shorter completion times are generally preferred, but this must be balanced with accuracy and safety considerations (Matteo *et al*., 2021). Energy efficiency is vital in systems that require long operational periods or autonomous functionality. Time-varying parameters like torque and speed directly impact the energy required. The goal is to minimize energy consumption while maintaining task performance, particularly in mobile and battery-powered robots. Measuring the energy consumed during task execution helps identify areas where parameter optimization can reduce power usage (Navarro *et al*., 2014; Göger *et al*., 2010; Ajoudani *et al*., 2017; Villani *et al*., 2018)).

Accuracy is crucial for tasks that require high levels of precision, such as manufacturing or surgery. Metrics like the root mean square error (RMSE) between the robot's actual trajectory and the desired path are used to evaluate how well the robot adapts to time-varying conditions. High accuracy indicates that the robot is effectively managing its parameters to follow precise paths (Matteo, 2023). The success rate reflects the proportion of tasks completed without failure. In complex sequential tasks, success may depend on how well the robot adjusts its time-varying parameters across all stages. A higher success rate implies that the robot is robustly managing variable parameters and executing tasks without breakdowns or errors (Pinto *et al*., 2021). Safety is critical in human-robot interaction. Force and torque sensors monitor whether the robot is operating within safe limits. For instance, excessive joint torque during task execution can indicate unsafe operations, especially in environments where robots interact with humans. Safety metrics ensure that time-varying parameters do not lead to hazardous conditions (Feng *et al*., 2018). The robot's ability to adapt to changes in its environment is key to success in dynamic tasks. This adaptability is measured through metrics like the robot's response time to external disturbances or environmental changes. For example, how quickly a robot adjusts its torque or speed when unexpected changes occur in the task sequence (Nottensteiner, 2023). Smoothness metrics, such as minimizing jerk (the rate of change of acceleration), are used to evaluate the fluidity of the robot’s motion. In tasks involving delicate operations, smooth transitions between task stages are essential to avoid errors or damage. Lower jerk values indicate more stable and controlled movements, which are crucial for sequential tasks (Khatib *et al*., 2019). Force and torque metrics are critical for assessing how the robot interacts with objects and its environment, particularly in tasks involving physical manipulation. Monitoring these values helps ensure that the robot does not apply excessive force, leading to potential task failure or hardware damage. Adjusting force and torque in real-time based on task needs helps in efficient task handling (Pires *et al*., 2021). Error rate is another important metric, measuring how frequently the robot deviates from its intended path or objective due to time-varying parameter mismatches. Additionally, the system’s ability to detect and correct these errors efficiently—without manual intervention—ensures smooth task execution and increased reliability (Siciliano & Khatib, 2019).

Conclusion: This review discusses a simulation framework for modeling time-varying parameters in robots as these do complex sequential jobs. By incorporating dynamic equations, machine learning, adaptive control, and sensor feedback, the simulation lets robots to alter in real-time in response to changing conditions/environments like terrain, varying loads, and task significances, thus can considerably increase the efficiency of robots in real-world applications, contributing a flexible way out for managing multi-stage dynamic tasks in industries like logistics and manufacturing.

REFERENCES

Acar, Nazli, José M. Franco, Erik Kuhn, David E. P. Gonçalves, and Jorge H. O. Seabra (2020). "Tribological Investigation on the Friction and Wear Behaviors of Biogenic Lubricating Greases in Steel–Steel Contact" *Applied Sciences* 10, no. 4: 1477. https://doi.org/10.3390/app10041477.

Ajoudani, A., Zanchettin, A. M., Ivaldi, S., Albu-Schäffer, A., Kosuge, K., and Khatib, O. (2017). Progress and prospects of the human-robot collaboration. Auton. Robots 42, 957–975. doi: 10.1007/s10514-017-9677-2

Bellman, R. (1957). Dynamic Programming. Princeton University Press.

Braunl, T. (2012). *Embedded Robotics—Mobile Robot Design and Applications with Embedded Systems (Russian)*. (3rd ed.) Springer.

Bry, A., & Roy, N. (2011). Rapidly-exploring random belief trees for motion planning under uncertainty. In IEEE International Conference on Robotics and Automation.

Canudas-de-Wit, C., Olsson, H., Åström, K. J., & Lischinsky, P. (1995). A new model for control of systems with friction. IEEE Transactions on Automatic Control, 40(3), 419-425.

Chen, K. and A. Astolfi, 2021). Adaptive Control for Systems With Time-Varying Parameters," in *IEEE Transactions on Automatic Control*, vol. 66, no. 5, pp. 1986-2001, doi: 10.1109/TAC.2020.3046141.

Cheng, L., Liu, Y., & Zhang, H. (2019). Dynamic energy management in robotic systems for unpredictable workloads. IEEE Access, 7, 95268-95280. https://doi.org/10.1109/ACCESS.2019.2928551.

Chinnappa, U. R (2023)AN APPROACH FOR RISK MITIGATION AND SAFETY DURING HUMAN-ROBOT COLLABORATION, MASTERS THESIS of Mechanical Engineering, Faculty of Engineering Technology University of Twente P.O. Box 217 7500 AE Enschede The Netherlands.

Craig, J. J. (2018). Introduction to robotics: Mechanics and control. Pearson.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, 6(2), 182-197.

Feng, Y., Li, M., & Gao, H. (2018). "Safety metrics in human-robot collaboration: A time-varying approach." Journal of Field Robotics, 35(6), 1238-1251.

Franklin, G. F., Powell, J. D., & Emami-Naeini, A. (2014). *Feedback Control of Dynamic Systems* (7th ed.). Pearson.

Galvan-Perez, Daniel, Francisco Beltran-Carbajal, Ivan Rivas-Cambero, Hugo Yañez-Badillo, Antonio Favela-Contreras, and Ruben Tapia-Olvera. 2023. "Motion-Tracking Control of Mobile Manipulation Robotic Systems Using Artificial Neural Networks for Manufacturing Applications" *Mathematics* 11, no. 16: 3489. https://doi.org/10.3390/math11163489.

Göger, D., Blankertz, M., and Wörn, H. (2010). A tactile proximity sensor. IEEE Sensors 589–594.

Haddad, W.M., Chellaboina, Vs. and Nersesov S.G. (2006) Impulsive and Hybrid Dynamical Systems, Stability, Dissipativity and Control. Princeton University Press, Princeton.https://doi.org/10.1515/9781400865246

Hogan, N. (1985). "Impedance control: An approach to manipulation." ASME Journal of Dynamic Systems, Measurement, and Control, 107(1), 1-24. https://doi.org/10.1115/1.3140702,

Hongyan Dui, Huiting Xu, Luyu Zhang, Jia Wang (2023). Cost-based preventive maintenance of industrial robot system. Reliability Engineering & System Safety,240:109595,10.1016/j.ress.2023.109595.

Ioannou, P., & Sun, J. (2012). Robust Adaptive Control. Dover Publications.

Isermann, R. (2006). Fault-diagnosis systems: an introduction from fault detection to fault tolerance. Springer Science & Business Media.

Jardine, A. K. S., Lin, D., & Banjevic, D. (2006). A review on machinery diagnostics and prognostics implementing condition-based maintenance. Mechanical Systems and Signal Processing, 20(7), 1483-1510.

Jianlong, L., D. Wang, X. Wu, K. Xu, and X. Liu (2022). "Vibration Prediction of the Robotic Arm Based on Elastic Joint Dynamics Modeling" *Sensors* 22, no. 16: 6170. https://doi.org/10.3390/s2216617.

Khatib, O., Kumar, V., & Siciliano, B. (2019). Control in Robotics: A Guide to Handling Dynamic Systems. MIT Press.

Knöchelmann, E., Steinke, D., Greenyer, J., Spindeldreier, S., Ortmaier, T. (2020). Trajectory Optimization Methods for Robotic Cells Considering Energy Efficiency and Collisions. In: Schüppstuhl, T., Tracht, K., Henrich, D. (eds) Annals of Scientific Society for Assembly, Handling and Industrial Robotics. Springer Vieweg, Berlin, Heidelberg. https://doi.org/10.1007/978-3-662-61755-7_21

Kober, J., Bagnell, J. A., & Peters, J. (2013). "Reinforcement learning in robotics: A survey." The International Journal of Robotics Research, 32(11), 1238-1274. https://doi.org/10.1177/0278364913495721

Kumar, A., & Patel, R. (2022). Life cycle assessment of robotics: Impacts of wear and tear on sustainability. Journal of Cleaner Production, 337, 131234. https://doi.org/10.1016/j.jclepro.2022.131234.

Laffranchi, M., N. G. Tsagarakis and D. G. Caldwell, "Safe human robot interaction via energy regulation control," *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, St. Louis, MO, USA, 2009, pp. 35-41, doi: 10.1109/IROS.2009.5354803.

Lintvedt, Mona Naomi (2023). Thermal imaging in robotics as a privacy-enhancing or privacy-invasive measure? Misconceptions of privacy when using thermal cameras in robots. Published in Digital Society 2, 33 (2023) https://link.springer.com/article/10.1007/s44206-023-00060-4, University of Oslo Faculty of Law Research Paper No. 2023-06, Available at SSRN: https://ssrn.com/abstract=4569269.

Liu, S. and Chen, G. (2019). Fundamentals of Robot Dynamics and Control. In Dynamics and Control of Robotic Manipulators with Contact and Friction (eds S. Liu and G. Chen). https://doi.org/10.1002/9781119422518.ch

MacDonald, Ian, and Rickey Dubay. 2024. "Development of an Adaptive Force Control Strategy for Soft Robotic Gripping" *Applied Sciences* 14, no. 16: 7354. https://doi.org/10.3390/app14167354.

Matteo Russo, Luca Raimondi, Xin Dong, Dragos Axinte, James Kell (2021).Task-oriented optimal dimensional synthesis of robotic manipulators with limited mobility, Robotics and Computer-Integrated Manufacturing. 69:102096,https://doi.org/10.1016/j.rcim.2020.102096.

Matteo, R 2023. "Measuring Performance: Metrics for Manipulator Design, Control, and Optimization" *Robotics* 12, no. 1: 4. https://doi.org/10.3390/robotics12010004.

Mayne, D. Q. (2014). Model predictive control: Recent developments and future promise. *Automatica*, 50(12), 2967-2986. https://doi.org/10.1016/j.automatica.2014.10.128.

Milecki, Andrzej, and Patryk Nowak. 2023. "Review of Fault-Tolerant Control Systems Used in Robotic Manipulators" *Applied Sciences* 13, no. 4: 2675. https://doi.org/10.3390/app13042675

Mobley, R. K. (2002). An introduction to predictive maintenance. Butterworth-Heinemann. Amsterdam London New York Oxford Paris Tokyo Boston San Diego San Francisco Singapore Sydney.

Murray, R. M., Li, Z., & Sastry, S. S. (2017). A mathematical introduction to robotic manipulation. CRC press.

Murray, R. M., Li, Z., & Sastry, S. S. (2017). A mathematical introduction to robotic manipulation. CRC press.

Navarro, S. E., Schonert, M., Hein, B., and WWörn, H. (2014). “6D proximity servoing for preshaping and haptic exploration using capacitive tactile proximity sensors,” in IEEE/RSJ Int. Conf. on Robots and Intelligent Systems, IROS.

Nonoyama, K.; Liu, Z.; Fujiwara, T.; Alam, M.M.; Nishi, T. Energy-Efficient Robot Configuration and Motion Planning Using Genetic Algorithm and

Particle Swarm Optimization. *Energies* 2022, *15*, 2074. https://doi.org/10.3390/en15062074.

Nottensteiner, K (2023). Adaptive Task Execution for Flexible Robotic Assembly Systems Using Intrinsic Tactile Sensing, An approved dissertation of Doctor of Engineering (Dr.-Ing.).Technische Universität München TUM School of Computation, Information and Technology.

Pinto, M.L., Wehrmeister, M.A. & de Oliveira, A.S. Real-Time Performance Evaluation for Robotics. J Intell Robot Syst 101, 37 (2021). https://doi.org/10.1007/s10846-020-01301-1

Pookkuttath, Sathian, Mohan Rajesh Elara, Vinu Sivanantham, and Balakrishnan Ramalingam. 2022. "AI-Enabled Predictive Maintenance Framework for Autonomous Mobile Cleaning Robots" *Sensors* 22, no. 1: 13. https://doi.org/10.3390/s22010013.

Qin, S. J., & Badgwell, T. A. (2003). A survey of industrial model predictive control technology. Control Engineering Practice, 11(7), 733-764.

Rudomanenko, A.P., Chernyadev, N.A., Vorotnikov, S.A. (2021). Adaptive Control System for Industrial Robotic Manipulator. In: Yuschenko, A. (eds) Modern Problems of Robotics. MPoR 2020. Communications in Computer and Information Science, vol 1426. Springer, Cham. https://doi.org/10.1007/978-3-030-88458-1_12.

Siciliano, B., & Khatib, O. (2016). Springer handbook of robotics. Springer.

Somani, N., M. Rickert, A. Gaschler, C. Cai, A. Perzylo and A. Knoll (*2016).*Task level robot programming using prioritized non-linear inequality constraints, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Daejeon, Korea (South), 2016, pp. 430-437, doi: 10.1109/IROS.2016.7759090.

Spong, M. W., Hutchinson, S., & Vidyasagar, M. (2006). Robot modeling and control. Wiley.

Srinivas, N., & Deb, K. (1994). Multiobjective optimization using nondominated sorting in genetic algorithms. Evolutionary Computation, 2(3), 221-248.

Stuhlenmiller, Florian, Steffi Weyand, Jens Jungblut, Liselotte Schebek, Debora Clever, and Stephan Rinderknecht. 2021. "Impact of Cycle Time and Payload of an Industrial Robot on Resource Efficiency" *Robotics* 10, no. 1: 33. https://doi.org/10.3390/robotics10010033

Tedrake, Russ (2023).Underactuated Robotics, Algorithms for Walking, Running, Swimming, Flying, and Manipulation. Course Notes for MIT 6.832https://underactuated.csail.mit.edu

Uhlmann, E., J. Polte, C. Geisert. (2020). Condition Monitoring Concept for Industrial Robots. 17th IMEKO TC 10 and EUROLAB Virtual Conference “Global Trends in Testing, Diagnostics & Inspection for 2030” October 20-22, 2020. pp253-257.

Villani, V., Pini, F., Leali, F., and Secchi, C. (2018). Survey on human-robot collaboration in industrial settings: safety, intuitive interfaces and applications. Mechatronics 55, 248–266. doi: 10.1016/j.mechatronics.2018.02.009.

Wang, C., Y. Li, S. S. Ge and T. H. Lee (2015). Optimal Critic Learning for Robot Control in Time-Varying Environments," in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 10, pp. 2301-2310, doi: 10.1109/TNNLS.2014.2378812.

X. Xu (2007). Impulsive and Hybrid Dynamical Systems: Stability, Dissipativity, and Control - s[Book review; W. M. Haddad; V. S. Chellaboina, and S. G. Nersesov]," in *IEEE Transactions on Automatic Control*, vol. 52, no. 11, pp. 2193-2194, Nov., doi: 10.1109/TAC.2007.906834.

Yao, J. F., Yang, Y., Wang, XC. *et al*. Systematic review of digital twin technology and applications. Vis. Comput. Ind. Biomed. Art 6, 10 (2023). https://doi.org/10.1186/s42492-023-00137-4.

Yoshikawa, T (2000). Force control of robot manipulators. *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (*Cat. No.00CH37065), San Francisco, CA, USA, pp. 220-226 vol.1, doi: 10.1109/ROBOT.2000.844062.

Yue, P., Xu, B. & M. Zhang (2024). An improve nonlinear robust control approach for robotic manipulators with PSO-based global optimization strategy. Sci Rep 14, 21447. https://doi.org/10.1038/s41598-024-72156-x.

Zadeh, L. A. (1965). Fuzzy sets. Information and Control, 8(3), 338-353.

Zhang, R., & Wei, J. Z. (2013). Multi-Sensor Information Fusion and its Application in Robots. Applied Mechanics and Materials, 443: 299-302. Trans Tech Publications, Ltd. https://doi.org/10.4028/www.scientific.net/AMM.443.299.