Proceedings of the International Conference on Industrial Engineering and Operations Management

Publisher: IEOM Society International, USA DOI: 10.46254/NA11.20260409
Published: June 9, 2026

A ROS 2-Based Modular Framework for Omnidirectional Mobile Manipulator Design: System Architecture and Coupled Kinematic Validation

Nayan Banik
Department of Computer Science
University of Wisconsin-Milwaukee (UWM)
Milwaukee, WI, USA
nbanik@uwm.edu

Jawhar Ghommam
Associate Professor
Department of Electrical and Computer Engineering
Sultan Qaboos University
Muscat, Oman
jawher@squ.edu.om

Mohammad H Rahman
Professor
Department of Mechanical Engineering
University of Wisconsin-Milwaukee (UWM)
Milwaukee, WI, USA
rahmanmh@uwm.edu

Abstract

Industry 4.0 manufacturing environments demand flexible automation systems capable of adapting to evolving production layouts and diverse operational requirements. Mobile manipulators that integrate omnidirectional bases with multi-degree-of-freedom serial arms address this demand; however, the coordinated control of simultaneous base locomotion and arm manipulation remains a significant engineering challenge. This paper presents MecaXarm, a modular ROS 2-based software framework for an omnidirectional mobile manipulator comprising a four-wheel mecanum-drive base and a six-degree-of-freedom industrial arm. The framework incorporates a modular five-package software architecture for rapid prototyping, a web-based human-machine interface for real-time operator monitoring, and a coupled kinematic control scheme based on a nine-degree-of-freedom extended Jacobian formulation using Resolved Motion Rate Control. The coupled controller is validated through Gazebo simulation via simultaneous execution of base lemniscate and end-effector Lissajous trajectories, achieving sub-millimeter mean end-effector tracking accuracy (0.18 mm) with the base locked and low-millimeter accuracy (2.16 mm mean) during coupled base–arm operation. A dual controller-manager architecture facilitates seamless transition from simulation to real hardware by isolating heterogeneous communication protocols. The results demonstrate that commercial off-the-shelf components integrated through open-source ROS 2 tools can deliver a validated mobile manipulator framework suitable for flexible manufacturing applications consistent with Industry 4.0 principles.

Keywords
Mobile Manipulator, Mecanum Wheels, ROS 2, Kinematic Control, Flexible Manufacturing

1. Introduction
The transition toward Industry 4.0 has increased the demand for manufacturing systems that are flexible, reconfigurable, and capable of supporting frequent changes in products, layouts, and operational requirements (Soori et al. 2024). Mobile manipulators are increasingly relevant in this context because they combine the workspace mobility of autonomous mobile platforms with the dexterity of robotic manipulators. Recent reviews identify mobile manipulators as important enablers of flexibility, adaptability, and collaboration in Industry 4.0 manufacturing environments, while also emphasizing that their deployment requires careful consideration of safety, human-robot interaction, cybersecurity, and system integration (Ghodsian et al. 2023).

Conventional fixed-base industrial robots provide high repeatability and throughput, but their operational workspace is restricted by their mounting location. As a result, adapting fixed robotic workcells to new production layouts often requires substantial re-engineering, additional fixtures, and downtime. Mobile manipulators address this limitation by allowing the manipulator to be repositioned across multiple workstations (Aggogeri and Pellegrini 2024; Ghodsian et al. 2023). NIST has also identified mobile manipulators as relevant to large-scale manufacturing processes, including aerospace, energy, transportation, and conformal additive manufacturing, where autonomous repositioning can enable continuous cooperation between the mobile base and arm (Aboul-Enein et al. 2024).

Among mobile platform designs, mecanum-wheeled bases are particularly suitable for constrained industrial environments because they provide holonomic planar motion. A mecanum-drive base can translate laterally, move longitudinally, and rotate in place without requiring complex maneuvering (Banik et al. 2025). When such a base is integrated with a multi-degree-of-freedom robotic arm, the combined system becomes kinematically redundant. This redundancy can extend the reachable workspace and support coordinated base-arm motion, but it also introduces additional sources of uncertainty and error. Recent work on industrial mobile manipulator accuracy emphasizes that mobile manipulators remain challenged by precision limitations, error sources across both the base and manipulator, and the need for improved control and compensation methods (Popov et al. 2024).

Despite the potential of mobile manipulators, the development of an integrated platform remains a significant systems engineering challenge. A complete solution must combine heterogeneous hardware interfaces, simulation models, controller configurations, robot descriptions, sensor integration, and operator interfaces. Recent mobile manipulation work continues to identify subsystem integration as a central difficulty because navigation, mapping, perception, grasping, control, and user interaction must operate as a coordinated workflow rather than as isolated modules (Adil et al. 2025).

ROS 2 provides a mature middleware foundation for modular robotic software. Macenski et al. (2022) describe ROS 2 as a redesign of ROS intended to address modern robotic requirements such as modularity, scalability, reliability, sensing, planning, mobility, and autonomy. The ROS 2 Jazzy Jalisco release is especially relevant to this work because it officially supports Ubuntu 24.04 as a Tier 1 platform, matching the software environment used in the MecaXarm framework (ROS 2 Docs 2026).

This paper presents MecaXarm, a modular ROS 2-based framework for the design, simulation, and deployment of an omnidirectional mobile manipulator. The system integrates a custom four-wheel mecanum-drive base with a UFACTORY xArm6 six-degree-of-freedom manipulator, depth-sensing devices, a web-based human-machine interface, and a coupled kinematic controller. The main objective of this work is to demonstrate that a modular ROS 2 software architecture can support rapid prototyping, simulation-based validation, and hardware-oriented deployment of a mobile manipulator suitable for flexible manufacturing environments.

The specific contributions of this paper are as follows:

a) A modular ROS 2 software architecture consisting of five custom packages for robot description, control, simulation, real-hardware bringup, and human-machine interaction.
b) A coupled nine-degree-of-freedom kinematic formulation that represents the mecanum base and six-degree-of-freedom manipulator as a unified mobile manipulation system.

c) A Resolved Motion Rate Control strategy with Closed-Loop Inverse Kinematics for end-effector trajectory tracking during arm-only and coupled base-arm operation.
d) A simulation-based validation study in Gazebo Harmonic using Lissajous end-effector trajectories and lemniscate base trajectories to quantify tracking accuracy.
e) A dual controller-manager deployment architecture that separates the EtherCAT-based mobile base interface from the TCP/IP-based manipulator interface to improve real-hardware communication reliability.

The remainder of this paper is organized as follows. Section 2 reviews related work on mobile manipulators, mecanum-wheel kinematics, whole-body kinematic control, ROS 2-based control frameworks, and Industry 4.0 human-machine interfaces. Section 3 presents the hardware and software architecture of the proposed system. Section 4 describes the coupled kinematic model and RMRC controller. Section 5 presents the simulation experiments, numerical results, and validation analysis. Section 6 discusses industrial implications, limitations, and future work. Section 7 concludes the paper.

2. Literature Review

2.1 Mobile Manipulators in Manufacturing

Mobile manipulators have been widely studied as an enabling technology for flexible automation because they combine mobility and manipulation within a single robotic system. Mobile manipulation is a unified problem involving coordinated locomotion and manipulation, and subsequent platforms such as the KUKA youBot demonstrated the feasibility of integrating an omnidirectional base with a serial manipulator for research and education (Ancona 2017). More recent reviews show that mobile manipulators are increasingly relevant to Industry 4.0 because they support adaptability, connectivity, and collaborative manufacturing, while also raising practical concerns related to safety, human-robot interaction, perception, data analysis, virtualization, and cybersecurity (Ghodsian et al. 2023). Within manufacturing environments, mobile manipulators are especially relevant when production layouts must be reconfigured or when tasks are distributed across multiple stations. Instead of installing multiple fixed robots, a mobile manipulator can travel between work areas and perform different manipulation tasks. NIST identifies continuous mobile manipulation as valuable for large, curved, and complex workpieces, but also notes that such flexibility introduces performance uncertainty that can affect pose repeatability and accuracy. Recent standardization activity also shows that mobile manipulator performance measurement is becoming an important industrial concern (Aboul-Enein et al. 2024).

2.2 Mecanum Wheel Kinematics

Mecanum wheels enable omnidirectional motion through angled passive rollers mounted around the circumference of each wheel. The roller geometry allows a four-wheel mecanum platform to generate longitudinal, lateral, and rotational motion by coordinating individual wheel speeds. This capability makes mecanum-drive bases attractive for mobile manipulation in constrained environments, where lateral motion and in-place rotation reduce the need for large turning maneuvers (Hernández et al. 2024; Banik et al. 2025). However, mecanum-wheel platforms also introduce implementation challenges. Wheel slip, roller-ground contact variation, friction uncertainty, and floor-surface dependency can cause deviations between commanded and actual base motion. These effects are particularly important for mobile manipulators because base pose error propagates directly into end-effector pose error. Recent reviews on mobile manipulator accuracy identify the mobile base, manipulator, calibration, localization, and control strategy as coupled contributors to overall pose error, reinforcing the need for validation metrics that quantify end-effector accuracy during simultaneous base-arm motion (Galati et al. 2022). Recent studies on mecanum-wheeled robots have highlighted the impact of wheel-ground uncertainty, payload variations, and disturbances on trajectory tracking performance. Banik et al. (2026) demonstrated improved robustness under such conditions using a model-based iterative learning control approach, emphasizing the importance of considering base-level tracking errors when evaluating mobile manipulators.

2.3 Whole-Body Kinematic Control

Whole-body kinematic control provides a framework for coordinating mobile-base and manipulator motion. Whitney (1969) introduced Resolved Motion Rate Control, which maps desired Cartesian end-effector velocities into joint velocities through the manipulator Jacobian. Later work extended Jacobian-based approaches to mobile manipulators by treating the base and manipulator as a unified kinematic system. In this formulation, the extended Jacobian combines the contribution of the mobile base with the contribution of the arm joints. For mobile manipulators, the extended Jacobian is especially useful because it provides a direct relationship between base velocity, arm joint velocity, and end-effector twist. Since the MecaXarm system has nine degrees of freedom and a six-dimensional end-effector task space, it is kinematically redundant. The pseudo-inverse of the extended Jacobian can be used to obtain a minimum-norm velocity solution, while damped least-squares regularization improves numerical stability near singular configurations. Recent studies continue to identify coordinated control and precision enhancement as key research needs for industrial mobile manipulators, particularly when high end-effector accuracy is required during base motion (Haviland and Corke 2023).

2.4 ROS 2 and ros2_control Framework

ROS 2 was developed to address limitations of ROS 1 and to better support modern robotic systems requiring modularity, scalability, reliability, sensing, planning, mobility, and autonomy (Macenski et al. 2022). This makes ROS 2 appropriate for mobile manipulation systems, where multiple subsystems must communicate through shared software architecture. The MecaXarm framework is implemented using ROS 2 Jazzy Jalisco, the tenth ROS 2 release, which supports Ubuntu 24.04 as a Tier 1 platform. This is relevant because the proposed framework targets ROS 2 Jazzy on Ubuntu 24.04, which provides a stable and officially supported software foundation for simulation and deployment (ROS 2 Docs 2026). The ros2_control framework provides a standardized hardware abstraction layer for robot control. Its official documentation describes ros2_control as a framework for real-time control of robots in ROS 2, with the goal of simplifying new hardware integration and addressing drawbacks of earlier ros_control approaches (ROS 2 Control Docs 2026). The framework also includes commonly used controllers such as forward command controllers, joint trajectory controllers, and differential drive controllers, supporting modular controller development and reuse. For simulation, gz_ros2_control connects Gazebo with the ros2_control interface. The official Jazzy documentation states that gz_ros2_control is released for ROS 2 Jazzy on Ubuntu and that, for ROS 2 Jazzy and Rolling, the Gazebo version provided through the Gazebo vendor packages is Harmonic. This supports the use of Gazebo Harmonic as the simulation environment in the proposed framework.

2.5 Gazebo-Based Simulation and Virtual Commissioning

Simulation is important for mobile manipulation because it allows controller logic, robot descriptions, sensor bridges, and trajectory-generation methods to be validated before physical deployment. Gazebo Harmonic supports ROS 2 interaction through bridge mechanisms that exchange messages between ROS 2 and Gazebo Transport, enabling joint states, transforms, sensor streams, and commands to move between the simulation and ROS 2 ecosystem (Gazebo Docs 2026). This simulation-to-hardware continuity is closely related to virtual commissioning. Recent work on digital twins for robot systems in manufacturing states that digital twins can help address challenges associated with designing, testing, commissioning, and operational reconfiguration of robot systems. This supports the design choice in MecaXarm to maintain common robot descriptions and controller abstractions across Gazebo simulation and real-hardware configurations (Mesa et al. 2025).

2.6 Industry 4.0, Digital Twins, and Human-Machine Interfaces

Industry 4.0 emphasizes digital connectivity, real-time monitoring, human-centered automation, and integration of cyber-physical systems into manufacturing environments. For mobile manipulators, a human-machine interface must support both system-level awareness and task-level control. Operators need to monitor robot state, view sensor feedback, control the base and manipulator when necessary, and diagnose controller status. Recent work on human-centric digital twins argues that human-centered manufacturing increasingly relies on digital twins and human-machine collaboration, especially from the perspective of Industry 5.0 and human-centered applications (Krupas et al. 2024). This supports the inclusion of a digital-twin-style visualization and web-based HMI in MecaXarm, where the operator can observe the robot state, controller status, and telemetry through a browser-based dashboard. Mobile manipulators are not always expected to operate under full autonomy. A 2025 review on variable autonomy in mobile manipulators emphasizes that many systems require human-robot teaming to operate safely and reliably under uncertainty, and identifies cognitive workload, communication delay, shared control, and whole-body variable autonomy as continuing challenges (Contreras et al. 2025). This further motivates the web-based operator interface and teleoperation functions included in the proposed framework.

3. System Design and Methods

The MecaXarm system architecture is designed to support modularity, hardware abstraction, and operator accessibility. Modularity allows each subsystem to be developed and maintained independently. Hardware abstraction enables the same robot model and controller structure to be used in both simulation and real-hardware configurations.

Operator accessibility is addressed through a web-based HMI that provides real-time visualization, monitoring, and teleoperation without requiring specialized software installation.

3.1 Hardware Platform

The MecaXarm platform consists of a custom four-wheel mecanum-drive base integrated with a UFACTORY xArm6 six-degree-of-freedom manipulator. The mobile base provides holonomic planar motion, while the robotic arm provides dexterous manipulation within the workspace surrounding the platform. The system also includes sensing components for perception, localization, and operator visualization. Table 1 summarizes the main hardware specifications, and Figure 1 shows the assembled mobile manipulator. This hardware configuration reflects the broader direction of industrial mobile manipulation, where the mobility of an autonomous platform is combined with the precision and dexterity of a robotic arm. Recent mobile-manipulator reviews emphasize that this integration can improve manufacturing flexibility, but also requires attention to safety, human-robot interaction, connectivity, and system-level integration (Ghodsian et al. 2023).

Table 1. MecaXarm hardware specifications

Component	Specification
Base platform	Custom mecanum drive, 0.81 × 0.38 m chassis
Base mass	44 kg (including structure, motors, electronics)
Wheels	4× mecanum wheels, R = 0.09 m, roller angle 45°
Base DOF	3 (planar: x, y, yaw) — holonomic
Geometric param (lx + ly)	0.56 m
Arm	UFACTORY xArm6, 6-DOF collaborative arm
Arm payload / reach / weight	5 kg / 700 mm / 12.2 kg
Arm repeatability	±0.1 mm
Gripper	xArm parallel-jaw gripper
Base camera	Slamtec Aurora S (stereo RGBD + IMU)
EE camera	Stereolabs ZED 2i (stereo RGBD)
Arm mount offset (body frame)	[0.155, 0.0, 0.440] m from base_footprint

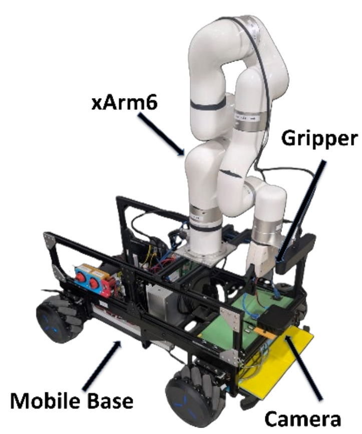

Figure 1. MecaXarm system with its main components

3.2 Modular ROS 2 Software Framework

The software framework is implemented in ROS 2 Jazzy and organized into five custom packages. This package-level separation improves maintainability and supports rapid prototyping because changes in one subsystem do not require modification of the entire codebase. The package inventory is summarized in Table 2, and the dependency relationships are shown in Figure 2. The use of ROS 2 is motivated by its modular architecture and its redesign for modern robotic systems. The mecaxarm_description package contains the URDF/XACRO robot model, mesh files, and ros2_control configuration files. The mecaxarm_control package contains custom control plugins and base-control components. The mecaxarm_gazebo package provides simulation launch files, worlds, and sensor bridges for Gazebo Harmonic. The mecaxarm_real package contains launch files and configuration files for physical hardware deployment. The mecaxarm_hmi package implements the web-based operator dashboard and its ROS bridge interface. The robot model is generated using XACRO to support reusable and configurable robot descriptions. Separate composition files are used for simulation and real-hardware deployment. Both configurations share the same geometry, joint structure, coordinate frames, and visual models, but they use different hardware-interface plugins. This design follows the ros2_control principle of separating controller logic from hardware-specific communication, allowing simulation and real-hardware configurations to share a common controller structure (ROS 2 Control Docs 2026).

Table 2. ROS 2 package inventory

Package	Purpose
mecaxarm_description	URDF/XACRO models, STL meshes, ros2_control YAML configs
mecaxarm_control	Custom C++ ros2_control plugins for torque-level base control
mecaxarm_gazebo	Gazebo Harmonic launch files, sensor bridges, worlds
mecaxarm_real	Real-hardware dual controller-manager launch, sensor bringup
mecaxarm_hmi	Web-based operator dashboard with ROS bridge nodes

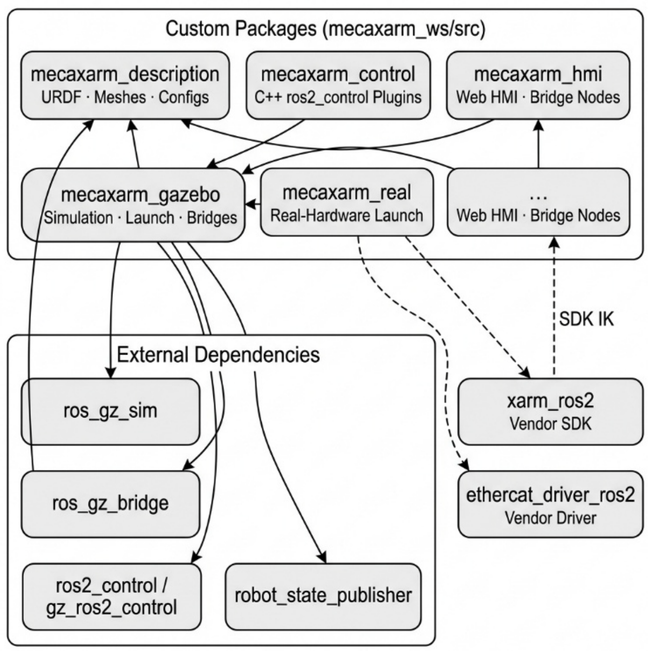

Figure 2. ROS 2 package dependency architecture of the MecaXarm framework

3.3 Dual Controller-Manager Architecture

Real-hardware deployment introduces a communication challenge because the mobile base and robotic arm rely on different communication protocols and update rates. The mecanum base uses an EtherCAT fieldbus with CiA402 motor drives, while the xArm6 communicates through a TCP/IP interface. In preliminary integration, running both hardware interfaces inside a single ros2_control node caused communication interference during high-frequency updates. To address this issue, the real-hardware configuration uses a dual controller-manager architecture. One controller manager is dedicated to the mecanum base and operates at the base-control update rate. A second controller manager is dedicated to the manipulator and gripper. This separation isolates the EtherCAT communication loop from the arm's TCP/IP communication loop, reducing the risk of bus starvation and improving subsystem reliability. This design is consistent with the broader systems-integration challenge reported in recent mobile manipulation literature, where heterogeneous subsystems such as mapping, navigation, perception, grasping, and motion control must be integrated into a seamless workflow. Table 3 details the configuration.

Table 3. Dual controller-manager configuration

Controller Manager	Hardware	Protocol	Rate	Key Controllers
Base CM	4 mecanum wheels	EtherCAT (CiA402)	200 Hz	mecanum_drive_controller, wheel_effort_controller
Arm CM	6 arm joints + gripper	TCP/IP (xArm SDK)	100 Hz	xarm6_traj_controller, xarm_gripper_traj_controller

3.4 Web-Based Human-Machine Interface

The MecaXarm human-machine interface is implemented as a web-based dashboard connected to ROS 2 through rosbridge_server and WebSocket communication. The HMI provides real-time monitoring, teleoperation, and diagnostics through three main widget groups: monitoring widgets for digital twin visualization, base status, wheel telemetry, and manipulator status; control widgets for base teleoperation, joint jogging, Cartesian jogging, and gripper commands; and system widgets for controller-manager status, console logs, and camera-feed integration. These functions allow operators to monitor system behavior, issue manual commands, and diagnose faults from a standard browser. The design aligns with recent work on digital twins, human-machine collaboration, and variable autonomy, which emphasizes the need for real-time visualization, operator involvement, and diagnostic support in uncertain mobile manipulation environments (Krupas et al. 2024; Contreras et al. 2025).

Table 4. HMI dashboard widget categories

Category	**Widgets**	**Function**
Monitoring	Digital Twin, Base Status, Wheel Telemetry, Arm Status	Real-time 3D visualization and telemetry
Control	Teleop D-Pad, Joint Jog, Cartesian Jog, Gripper Control	Direct operator commands
System	Controller Manager, Console Log, Camera Feed	Lifecycle management and diagnostics

4. Coupled Kinematic Model and RMRC Controller

This section presents the nine-DOF coupled kinematic model that treats the mecanum base and the serial arm as a single kinematic chain and derives the RMRC controller used for trajectory tracking.

4.1 Generalized Coordinates and Extended Jacobian

The MecaXarm has nine degrees of freedom: three from the holonomic base (planar translation and yaw) and six from the arm joints. The generalized coordinate vector is defined as:

$$q = [x_W \quad y_W \quad \psi \quad q_1 \quad q_2 \quad q_3 \quad q_4 \quad q_5 \quad q_6]^\top \in \mathbb{R}^9$$

here (x_W, y_W) is the base footprint position in the world frame, ψ is the base heading (yaw), and q_1 through q_6 are the arm joint angles. The end-effector spatial velocity (twist) in the world frame is related to the generalized velocity through the extended Jacobian:

$$\xi_{ee} = J_{ext}(q)\,\dot{q}_{ext} \in \mathbb{R}^6$$

where $J_{ext} \in \mathbb{R}^{6\times 9}$ is constructed by horizontal concatenation of the base Jacobian and the arm Jacobian:

$$J_{ext} = [J_{base} \quad J_{arm}]$$

Base Jacobian. When the base moves with world-frame velocity $[\dot{x}_W, \dot{y}_W]$ and yaw rate $\dot{\psi}$ while the arm joints are held fixed, the end-effector velocity is:

$$v_{ee} = v_{base} + \omega_{base} \times r$$

where $r = p_{ee} - p_{base}$ is the moment arm from the base footprint to the end-effector. Since $\omega_{base} = [0,\ 0,\ \dot{\psi}]^\top$, the base Jacobian takes the form:

$$J_{base} = \begin{bmatrix} 1 & 0 & -r_y \\ 0 & 1 & r_x \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

where r_x and r_y are components of the moment arm r in the world frame.

Arm Jacobian. The arm Jacobian $J_{arm} \in \mathbb{R}^{6\times6}$ is computed using the KDL (Kinematics and Dynamics Library) chain solver from the URDF model and rotated to the world frame using the base heading:

$$J_{arm}^{W} = \text{blkdiag}(R_{WB},\ R_{WB})\ J_{arm}^{KDL}$$

where $\boldsymbol{R_{WB}} = \text{Rot}_z(\psi)$ is the rotation matrix from the base body frame to the world frame.

4.2 CLIK Algorithm

The Resolved Motion Rate Control (RMRC) controller uses Closed-Loop Inverse Kinematics (CLIK) to compute joint velocity commands from task-space errors. The algorithm operates at each control cycle (50 Hz) as follows.

Step 1: *Task-space error:* Given the desired end-effector pose $(\boldsymbol{p_d}, \boldsymbol{R_d})$ and the current pose $(\boldsymbol{p_{ee}}, \boldsymbol{R_{ee}})$ computed via forward kinematics:

$$e_{pos} = p_d - p_{ee} \in \mathbb{R}^3$$
$$e_{ori} = \text{angle-axis}(R_d\, R_{ee}^{\top}) \in \mathbb{R}^3$$
$$e_{task} = \begin{bmatrix} e_{pos} \\ e_{ori} \end{bmatrix} \in \mathbb{R}^6$$

Step 2: *Velocity command:* The task-space velocity command combines feedforward (desired velocity from the trajectory generator) with proportional correction:

$$\dot{\boldsymbol{x}}_{cmd} = \dot{\boldsymbol{x}}_{ff} + K\, \boldsymbol{e_{task}}$$

where $\boldsymbol{K}$ is the CLIK proportional gain ($\boldsymbol{K} = \mathbf{10.0}$ in our implementation).

Step 3: *Damped pseudo-inverse*: To avoid numerical instability near kinematic singularities, we use the Levenberg-Marquardt (damped least-squares) pseudo-inverse:

$$\boldsymbol{J}^{\dagger} = \boldsymbol{J}^{\top}(\boldsymbol{J}\boldsymbol{J}^{\top} + \lambda^2 \boldsymbol{I})^{-1}$$

where $\boldsymbol{\lambda} = \mathbf{0.01}$ is the damping factor.

Step 4: *Joint velocity mapping*: The full 9-DOF joint velocity command is:

$$\dot{q}_{cmd} = J^{\dagger}\, \dot{x}_{cmd}$$

The first three components $[\dot{\boldsymbol{x}}_{\boldsymbol{W}}, \dot{\boldsymbol{y}}_{\boldsymbol{W}}, \dot{\boldsymbol{\psi}}]$ are transformed to body-frame velocities and published as base velocity commands. The remaining six components are arm joint velocities.

Step 5: *Integration and drift correction*: Rather than integrating from noisy joint feedback each cycle, the controller maintains a tracked joint state q_clik and applies alpha-blend drift correction:

$$q_{cmd} = (1 - \alpha)(q_{clik} + \dot{q}_{arm}\,\Delta t) + \alpha\, q_{measured}$$

where $\boldsymbol{\alpha} = \mathbf{0.05}$ provides a 5% correction toward the measured state per cycle, preventing open-loop drift while preserving smooth tracking. Table 5 summarizes the RMRC controller parameters.

Table 5. RMRC controller parameters

Parameter	Symbol	Value	Description
Control rate	f	50 Hz	Control loop frequency
CLIK gain	K	10.0	Proportional correction gain
Damping factor	λ	0.01	Pseudo-inverse regularization
Alpha blend	α	0.05	Drift correction factor
Max joint velocity	-	1.5 rad/s	Joint velocity saturation
Max base velocity	-	0.5 / 0.5 / 0.5	vx / vy / ω limits (m/s, rad/s)

4.3 Coupled Trajectory Generation

To validate the coupled kinematic model, we define simultaneous trajectories for the base and arm end-effector.

Base trajectory — Lemniscate (figure-8): The base follows a lemniscate of Bernoulli in the world x-y plane:

$$x_{base}(t) = \frac{A\sin(\omega t)}{1 + \cos^2(\omega t)}, \qquad y_{base}(t) = \frac{A\sin(\omega t)\cos(\omega t)}{1 + \cos^2(\omega t)}$$

with amplitude $\boldsymbol{A} = \mathbf{0.5}$ m and angular frequency $\boldsymbol{\omega} = \mathbf{0.2}$ rad/s.

Arm trajectory — 3D Lissajous: The arm end-effector follows a 3D Lissajous curve relative to its initial position:

$$\boldsymbol{p_{ee}}(t) = \boldsymbol{p_0} + \rho(t)\begin{bmatrix} A_x \sin(2\pi f_x t) \\ A_y \sin(2\pi f_y t) \\ A_z \sin(2\pi f_z t) \end{bmatrix}$$

with amplitudes $[\boldsymbol{A_x}, \boldsymbol{A_y}, \boldsymbol{A_z}] = [\mathbf{100}, \mathbf{100}, \mathbf{50}]$ mm, frequencies $[\boldsymbol{f_x}, \boldsymbol{f_y}, \boldsymbol{f_z}] = [\mathbf{0.4}, \mathbf{0.5}, \mathbf{0.3}]$ Hz (ratio 4:5:3, full pattern period 10 s), ramp function $\boldsymbol{\rho(t)} = \mathbf{min}(\boldsymbol{t/T_{ramp}},\ \mathbf{1})$, and $\boldsymbol{T_{ramp}} = \mathbf{5}$ s to ensure smooth startup without velocity discontinuities.
The simultaneous execution of these trajectories tests the full nine-DOF coupled kinematic model: the base navigates a complex planar path while the arm independently traces a three-dimensional pattern, with the extended Jacobian distributing the required motion across all nine DOFs.

4.4 Redundancy Resolution
With nine DOFs and a six-dimensional task space, the system possesses three redundant DOFs. The RMRC controller naturally exploits this redundancy through the pseudo-inverse, which produces the minimum-norm joint velocity solution. Additionally, the null-space projector $N = I - J^{\dagger}J \in \mathbb{R}^{9\times 9}$ is available for secondary objectives such as joint centering or preferred heading maintenance. In the current implementation, the redundancy is resolved implicitly through the velocity limits applied to the base and arm DOFs independently, which shape the velocity distribution between locomotion and manipulation.

5. Simulation Validation
5.1 Simulation Environment
The RMRC controller was validated in Gazebo Harmonic integrated with ROS 2 Jazzy Jalisco on Ubuntu 24.04 LTS. The simulation used the gz_ros2_control plugin as the hardware interface layer, allowing the controller to access the same position, velocity, and effort interfaces used for real-hardware deployment. The simulated robot model was configured with the same URDF geometry, inertial properties, and joint limits as the physical MecaXarm platform to maintain consistency between simulation and hardware configurations. The Gazebo simulation used the default DART physics engine with a physics update rate of 1000 Hz. The RMRC control loop operated at 50 Hz, while the arm Joint Trajectory Controller received position commands with a 40 ms time-from-start horizon. The arm-only experiment was executed for more than 70 seconds, and the coupled base-arm experiment was executed for more than 30 seconds. This setup provided a controlled simulation environment for evaluating both isolated manipulator tracking and simultaneous base-arm coordination.

5.2 Experiment 1: Arm-Only Tracking (Base Locked)
In the first experiment, the mobile base is locked (all base velocity limits set to zero) and only the six arm DOFs are active. The end-effector tracks a 3D Lissajous trajectory with amplitudes [60, 50, 20] mm and frequencies [0.15, 0.25, 0.1] Hz (ratio 6:10:4). The trajectory is centered at the initial forward-kinematic position of the end-effector.

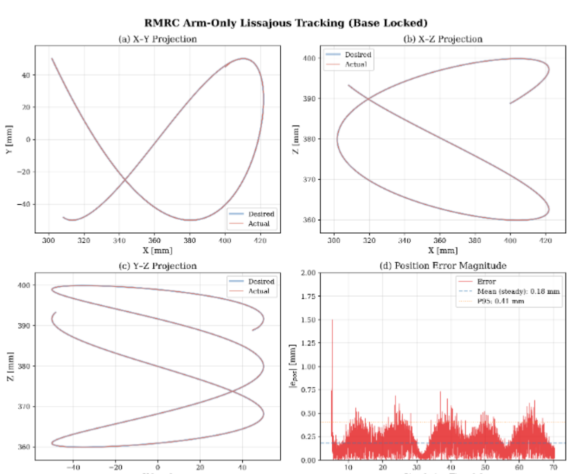

Figure 3. RMRC arm-only Lissajous tracking results. (a) X–Y projection showing desired vs. actual EE path, (b) X–Z projection, (c) Y–Z projection, (d) Position error magnitude vs. simulation time with mean steady-state error annotated.

The arm-only experiment achieved sub-millimeter tracking accuracy, with a mean steady-state end-effector error of 0.18 mm and a maximum error of 0.73 mm (see Figure 3). The 95th percentile error was 0.41 mm, indicating stable tracking over the full 71.4 s simulation. No joint-velocity saturation events or joint-limit violations were observed. These results show that the selected CLIK parameters provided accurate trajectory tracking when the base was stationary.

5.3 Experiment 2: Coupled Base + Arm Tracking

In the second experiment, both the mobile base and the arm are active simultaneously. The base follows a lemniscate trajectory with amplitude A = 0.5 m and frequency ω = 0.2 rad/s, while the arm end-effector independently tracks a 3D Lissajous trajectory with amplitudes [100, 100, 50] mm and frequencies [0.4, 0.5, 0.3] Hz. Base velocity limits are set to [0.5, 0.5, 0.5] m/s (vx, vy, ω). The results are shown in Figure 4 and Figure 5.

This experiment is the primary validation of the coupled kinematic model. The extended Jacobian must correctly distribute the desired end-effector velocity across all nine DOFs, accounting for the coupling between base motion and end-effector position through the moment arm r.

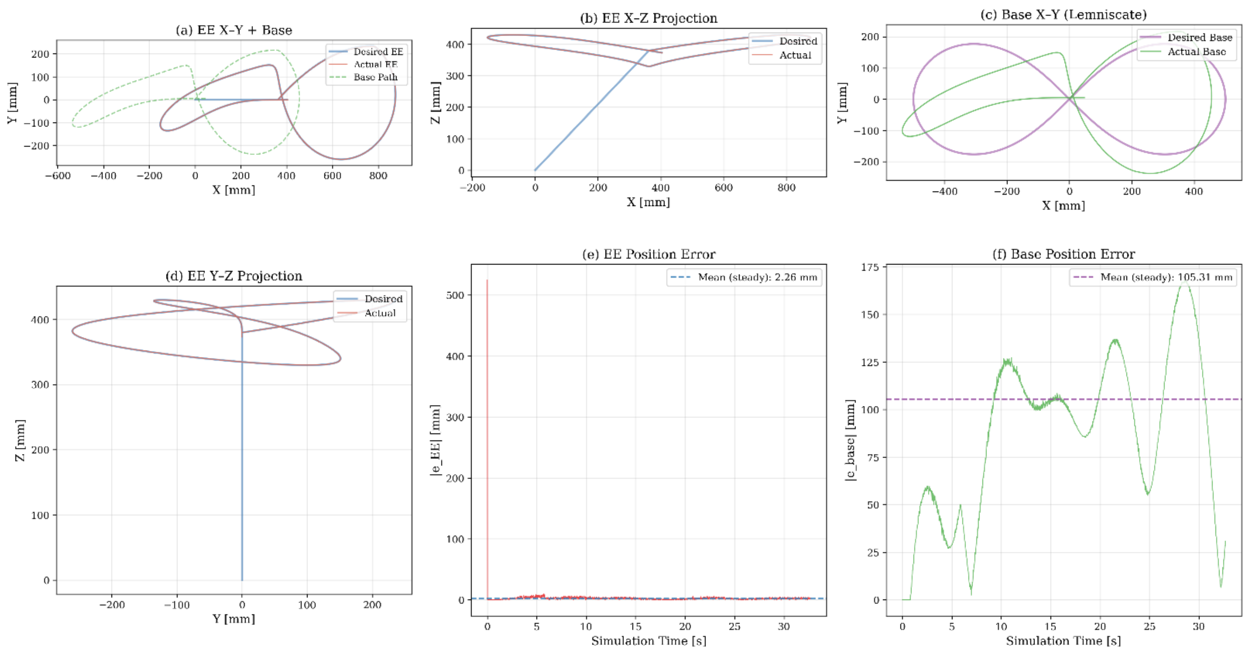

Figure 4. RMRC coupled base+arm tracking results. (a) EE X–Y projection with desired and actual paths plus base trajectory, (b) EE X–Z projection, (c) Base X–Y lemniscate (desired vs. actual), (d) EE Y–Z projection, (e) EE position error vs. time, (f) Base position error vs. time.

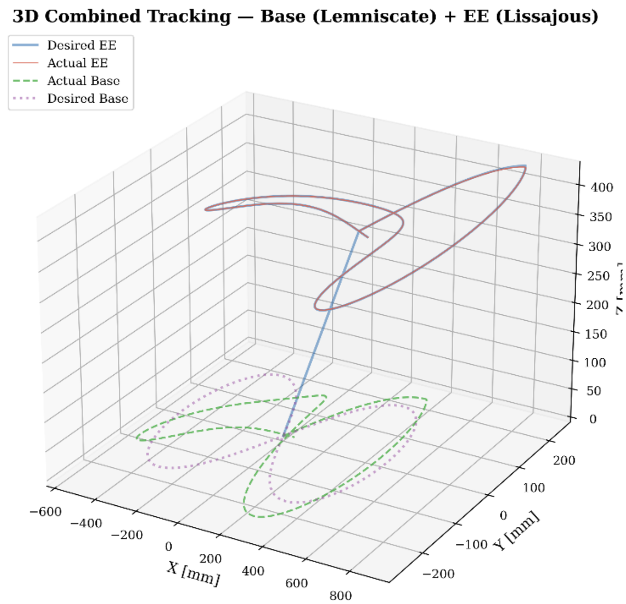

Figure 5. 3D visualization of coupled tracking. The base lemniscate trajectory is shown on the ground plane (Z = 0) and the end-effector Lissajous curve is shown in 3D space. Blue: desired trajectories, red: actual trajectories.

5.4 Consolidated Performance Summary

The consolidated results in Table 6 show that the RMRC controller achieved accurate end-effector tracking in both experimental conditions. In the arm-only case, the controller achieved sub-millimeter tracking accuracy with a mean end-effector error of 0.18 mm. In the coupled base-arm case, the mean end-effector error increased to 2.16 mm due to simultaneous base motion, larger trajectory amplitudes, and base-arm kinematic coupling. Nevertheless, the error remained within the low-millimeter range, confirming that the extended Jacobian formulation can coordinate the mobile base and manipulator within a unified kinematic control framework.

The coupled experiment also produced a mean base position error of 105.9 mm. This error is practically important because accurate base positioning is required when a mobile manipulator must align with workstations, fixtures, or constrained operating regions. The observed base error is mainly attributed to the kinematic-only base control, which does not compensate for wheel-ground interaction, mecanum roller slip, friction variation, actuator response, or localization uncertainty. However, despite this base-level deviation, the end-effector remained well regulated in the world frame, indicating that the CLIK controller partially compensated for base-motion errors at the task-space level. These results support the feasibility of the proposed framework while highlighting the need for future real-hardware validation, improved closed-loop base control, and dynamic compensation.

Table 6. Consolidated tracking performance across simulation experiments

Metric	Arm-Only Tracking	Coupled Base-Arm Tracking
Base motion	Stationary	Lemniscate, A=0.5 m, ω=0.2 rad/s
End-effector trajectory	3D Lissajous, 60/50/20 mm, 0.15/0.25/0.1 Hz	3D Lissajous, 100/100/50 mm, 0.4/0.5/0.3 Hz
Simulation duration	71.4 s	32.7 s
Mean EE position error	0.18 mm	2.16 mm
Maximum EE position error	0.73 mm	6.21 mm
95th percentile EE error	0.41 mm	Not evaluated
Mean base position error	Not applicable	105.9 mm
Maximum base position error	Not applicable	168.0 mm
Joint velocity saturation	None	None
Joint-limit violations	None	None

6. Discussion

6.1 Industrial Implications

The MecaXarm framework demonstrates several properties that are directly relevant to industrial engineering operations:

- *Operational flexibility:* The omnidirectional mecanum base enables the robot to navigate constrained factory environments and access multiple workstations without requiring dedicated pathways or layout modifications. This supports the trend toward flexible manufacturing systems where production lines must be rapidly reconfigured for different product variants.
- *Reduced integration cost:* The modular ROS 2 architecture allows individual subsystems (base, arm, sensors, HMI) to be developed, tested, and replaced independently. A manufacturing facility could swap the xArm6 for a different manipulator or replace the mecanum base with an alternative platform by modifying only the URDF and hardware interface plugin, without changing the control or monitoring software.
- *Virtual commissioning:* The identical software stack running in Gazebo and on real hardware supports virtual commissioning workflows, where new control strategies and trajectories can be validated offline before deployment. This reduces the time and risk associated with testing on physical equipment, a key consideration for production environments where downtime is costly.
- *Operator accessibility:* The web-based HMI eliminates the need for specialized software installations on operator workstations. Any device with a web browser can access the dashboard for monitoring, teleoperation, and diagnostics. The throttled subscription model (2 Hz frontend updates) ensures responsive performance even over constrained network connections.

6.2 Limitations and Future Work

Several limitations of the current work should be noted:

- ***Simulation-only validation:*** While the framework architecture supports real-hardware deployment (dual controller-manager, real-hardware launch files, conservative parameter configs), the kinematic tracking results presented in this paper are from simulation only. Physical experiments are required to confirm communication reliability, actuator response, localization accuracy, and controller performance under real floor-contact conditions. Real-hardware validation with the RMRC controller is ongoing and will be reported in future work.
- ***Kinematic-only control:*** The RMRC controller treats the system as a pure kinematic chain, ignoring dynamic effects such as inertia, Coriolis forces, and friction. These unmodeled effects are particularly relevant for mecanum-wheel platforms and are likely contributors to the 105.9 mm mean base position error observed during the coupled simulation. However, this is adequate for position-tracking tasks at moderate speeds but would be insufficient for contact-rich tasks (e.g., grinding, polishing) or high-speed manipulation where dynamic coupling becomes significant. The framework includes a Computed Torque Controller (CTC) based on the full nine-DOF Euler-Lagrange dynamic model, which will be the subject of a separate study.
- ***Lack of Comparative advantages:*** The current evaluation does not include a direct benchmark against alternative mobile-manipulator control strategies such as decoupled base-arm control, hierarchical whole-body control, model predictive control, or dynamically compensated control.

Future work will address these limitations through real-hardware experiments, external motion-capture or SLAM-based ground-truth validation, benchmarking against alternative control strategies, and implementation of dynamic compensation. Additional work will integrate Nav2-based autonomous navigation, perception-based manipulation, and redundancy-resolution objectives such as joint-limit avoidance, preferred base heading, and obstacle-aware motion generation. These extensions are necessary before making stronger claims about industrial deployment readiness.

7. Conclusion

This paper presented MecaXarm, a modular ROS 2-based framework for the design, simulation, and deployment of an omnidirectional mobile manipulator. The system integrates a four-wheel mecanum-drive base, a UFACTORY xArm6 six-DOF manipulator, depth-sensing components, ros2_control-based hardware abstraction, and a web-based human-machine interface. The proposed architecture organizes robot description, control, simulation, real-hardware bringup, and operator interaction into dedicated ROS 2 packages, supporting maintainable development and simulation-to-hardware transition. A coupled nine-DOF kinematic model was developed using an extended Jacobian that combines the motion contributions of the mobile base and manipulator joints. The model was implemented through a Resolved Motion Rate Control framework with Closed-Loop Inverse Kinematics and damped least-squares inversion. Gazebo Harmonic simulations validated the controller under both arm-only and coupled base-arm conditions. The arm-only case achieved sub-millimeter mean end-effector tracking error, while the coupled case maintained low-millimeter mean tracking error during simultaneous base lemniscate and end-effector Lissajous trajectory execution. The results confirm that the extended Jacobian formulation can coordinate mobile-base and manipulator motion within a unified kinematic control framework. The increase in tracking error during coupled operation quantifies the effect of base motion on end-effector accuracy and highlights the need for future dynamic compensation and physical validation. Overall, MecaXarm demonstrates that commercial robotic components, ROS 2 middleware, ros2_control abstraction, Gazebo simulation, and web-based operator interfaces can be integrated into a flexible mobile manipulation platform suitable for Industry 4.0 manufacturing applications. Future work will focus on real-hardware experiments, dynamic control, autonomous navigation, perception-based manipulation, advanced redundancy resolution, and variable-autonomy operator support.

References

Aboul-Enein, O., *Continuous Mobile Manipulator Performance Measurement Data*. U.S. Department of Commerce, National Institute of Standards and Technology, 2024.

Adil, A.A. et al. *A Multi-Robot Collaborative Manipulation Framework for Dynamic and Obstacle-Dense Environments: Integration of Deep Learning for Real-Time Task Execution*. *Frontiers in Robotics and AI*. 12:1585544, 2025.

Aggogeri, F. & Pellegrini, N. *6-DOFs Robot Placement Based on the Multi-Criteria Procedure for Industrial Applications*. *Robotics*. 13(10):153, 2024.

Ancona, R. *Redundancy Modelling and Resolution for Robotic Mobile Manipulators: A General Approach*. *Advanced Robotics*. 31(13):706–715, 2017.

Banik, N., Ghommam, J. & Rahman, M.H. *A Robust Model-Based Corrector Iterative Learning Control for Mecanum Robots: Comparative Analysis of Adaptive Architectures*, in *2026 3rd International Conference on Unmanned Vehicle Systems-Oman (UVS)*. IEEE, 2026.

Banik, N., Ghommam, J. & Rahman, M.H. *Finite-Time Pure Pursuit Guidance Control of a Four Mecanum Wheeled Mobile Robot with Active Disturbance Rejection*. *IEEE Access*, 2025.

Contreras, C.A. et al. *A Mini-Review on Mobile Manipulators with Variable Autonomy*. *Frontiers in Robotics and AI*. 12:1540476, 2025.

Gazebo Docs. *Gazebo Harmonic Documentation*, 2026, Available: https://gazebosim.org/docs/harmonic/getstarted/. Accessed May 20, 2026.

Galati, R., Mantriota, G. & Reina, G. *Adaptive Heading Correction for an Industrial Heavy-Duty Omnidirectional Robot*. *Scientific Reports*. 12(1):19608, 2022.

Ghodsian, N. et al. *Mobile Manipulators in Industry 4.0: A Review of Developments for Industrial Applications*. *Sensors*. 23(19):8026, 2023.

Haviland, J. & Corke, P. *Manipulator Differential Kinematics: Part I: Kinematics, Velocity, and Applications [Tutorial]*. *IEEE Robotics & Automation Magazine*. 31(4):149–158, 2023.

Hernández, J.C.O. & Rosas Almeida, D.I. *Kinematic Control in a Four-Wheeled Mecanum Mobile Robot for Trajectory Tracking*. *The Journal of Engineering*. 2024(9):e70006, 2024.

Krupas, M. et al. *Towards a Human-Centric Digital Twin for Human–Machine Collaboration: A Review on Enabling Technologies and Methods*. *Sensors*. 24(7):2232, 2024.

Macenski, S. et al. *Robot Operating System 2: Design, Architecture, and Uses in the Wild*. *Science Robotics*. 7(66):eabm6074, 2022.

Mesa-Montoya, C.A. et al. *Integration of a Digital Twin Framework for Trajectory Control of a 2RRR Planar Parallel Manipulator Using ROS/Gazebo and MATLAB*. *Future Internet*. 17(4):146, 2025.

Popov, V. et al. *Kinematic Modeling with Experimental Validation of a KUKA®–Kinova® Holonomic Mobile Manipulator*. *Electronics*. 13(8):1534, 2024.

ROS 2 Docs. *ROS 2 Jazzy Documentation*, 2026, Available: https://docs.ros.org/en/jazzy/index.html/. Accessed May 20, 2026.

ROS 2 Control Docs. *ROS 2 Control Jazzy Documentation*, 2026, Available: https://control.ros.org/jazzy/index.html/. Accessed May 20, 2026.

Soori, M. et al. *Intelligent Robotic Systems in Industry 4.0: A Review*. *Journal of Advanced Manufacturing Science and Technology*. 4(3):2024007, 2024.

Whitney, D.E. *Resolved Motion Rate Control of Manipulators and Human Prostheses*. *IEEE Transactions on Man-Machine Systems*. 10(2):47–53, 1969.

Biographies

Nayan Banik received his B.Sc. and M.Sc. degrees in Computer Science and Engineering from Comilla University, Bangladesh, in 2016 and 2018, respectively. He is currently a Ph.D. student in Computer Science at the University of Wisconsin–Milwaukee, USA, and is on study leave from his position as Assistant Professor at Comilla University. His research interests include generative AI for robotics, autonomous mobile robot control, and assistive and rehabilitation technologies.

Jawhar Ghommam received his B.Sc. degree in Computer and Control Engineering from INSAT, Tunisia, in 2003, his M.Sc. degree from the University of Montpellier, France, in 2004, and his Ph.D. degree in Control Engineering jointly from the National Engineering School of Sfax and the University of Orleans in 2008. He is currently an Associate Professor in the Department of Electrical and Computer Engineering at Sultan Qaboos University, Oman. His research interests include nonlinear and adaptive control, nonholonomic and underactuated systems, autonomous networked systems, cooperative control, consensus, and sensor networks.

Mohammad H. Rahman is a Professor in the Department of Mechanical Engineering at the University of Wisconsin–Milwaukee, USA, and Director of the BioRobotics Lab. He received his B.Sc. degree in Mechanical Engineering from Khulna University of Engineering & Technology, Bangladesh, in 2001, his M.Eng. degree in Bio-Robotics from Saga University, Japan, in 2005, and his Ph.D. degree in Engineering from École de technologie supérieure, Université du Québec, Canada, in 2012. His research interests include bio-robotics, wearable and collaborative robots, mobile robotics, intelligent systems, nonlinear control, and biological-signal-based control.