AMBF-Based Simulation of RAVEN-II Surgical Robot with Realistic Control and Feedback API

Boyang Li, Haonan Peng, Natalie Chalfant, Yun-Hsuan Su, Blake Hannaford, *Fellow, IEEE*

***Abstract*—Realistic and safe testing environments are essential for the advancement of surgical robotics. However, developing such environments is challenging due to hardware limitations, high costs, and safety concerns. In this work, we present an integrated simulation framework combining the detailed kinematic realism of the Asynchronous Multi-Body Framework (AMBF) simulator with the standardized, real-time control capabilities of the Collaborative Robotics Toolkit (CRTK) API. This integration creates an easily accessible environment that replicates the RAVEN-II surgical robot at a control frequency of 1000 Hz. By overcoming critical limitations of existing simulators, our solution provides researchers with enhanced tools for exploring innovative robotic control strategies and easier sim-to-real transfer, accelerating the development of safer and more effective surgical robotics.**

***Index Terms*—surgical robotics, RAVEN-II, robotic simulation, AMBF, CRTK, real-time control**

I. Introduction

The rise of surgical robotics has significantly enhanced precision and outcomes in medical procedures [1]. However, testing novel control strategies directly on robotic hardware poses significant practical challenges, including potential hardware damage and high operational costs [2] [3]. Simulation-based methods offer a promising alternative by allowing preliminary testing in risk-free environments. Yet, many existing simulators lack standardized control interfaces, and do not provide sufficient real-time feedback, limiting their effectiveness and reliability. Conversely, although Collaborative Robotics Toolkit (CRTK) [4] provides a control standard, integrating it with high-fidelity simulation environments has been non-trivial.

Recognizing these gaps, we designed a new simulation framework that integrates AMBF's sophisticated modeling capabilities [5] with the widely-adopted CRTK standard. This work provides comprehensive real-time feedback, realistic kinematic behavior, and realistic control interfaces to enhance the feasibility and impact of RAVEN-II [6] robotic surgery research.

II. Methods

A. System and Control Framework

The overall architecture of our RAVEN-II simulation framework integrates multiple functional modules to emulate the behavior, control structure, and data flow of the real robot in a virtual environment, as shown in Fig. 1. At its core, the system orchestrates real-time bidirectional communication between control inputs and the AMBF RAVEN-II simulation using the CRTK command protocol and custom feedback layers such as `raven_state`.

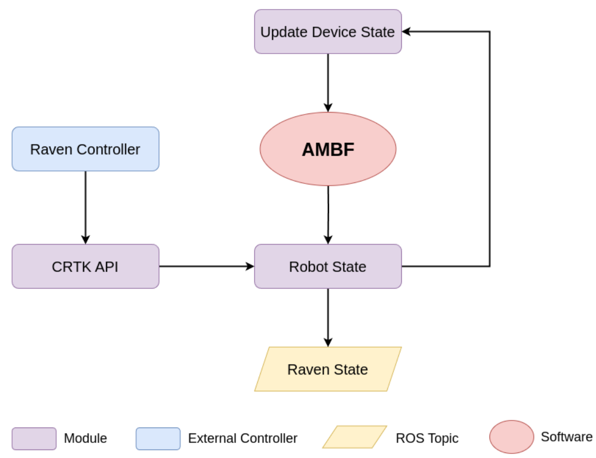

Fig. 1. System architecture of the simulated RAVEN-II, highlighting CRTK-driven control flow and high-frequency state feedback.

The CRTK API layer manages standardized command and feedback topics, providing a familiar interface for researchers accustomed to the real robot's control scheme [4]. Commands such as joint or Cartesian pose targets are published using CRTK topics by Raven Controller, or any controller works on real-world RAVEN-II, which are then received by the Robot State module.

The Robot State acts as a centralized hub that collects and manages the internal kinematic status of the robot. It receives data from AMBF (e.g., joint angles) and also receives control commands via the CRTK API. The Update Device State process ensures that the control loop remains synchronized by continuously pushing updates from Robot State to the AMBF simulator at 1000 Hz, reflecting the latest actuation instructions.

Meanwhile, the Raven State module derives its information exclusively from the Robot State. It repackages both original measurements and computed derivatives (e.g., motor velocities, encoder values, end-effector transforms) into a structured ROS message format consistent with the real-world `raven_state` specification. This message, published at 1000 Hz, allows external modules and researchers to monitor the robot's behavior with high temporal resolution.

B. Raven State Data Integration

In this work, we implemented the `raven_state` message, a comprehensive robot state data structure specific to

Boyang Li, Haonan Peng, and Blake Hannaford are with the University of Washington, Seattle, WA 98195, USA. `bli24, penghn, blake@uw.edu`

Natalie Chalfant and Yun-Hsuan Su are with Mount Holyoke College, South Hadley, MA 01075, USA. `chalf22n, msu@mtholyoke.edu`

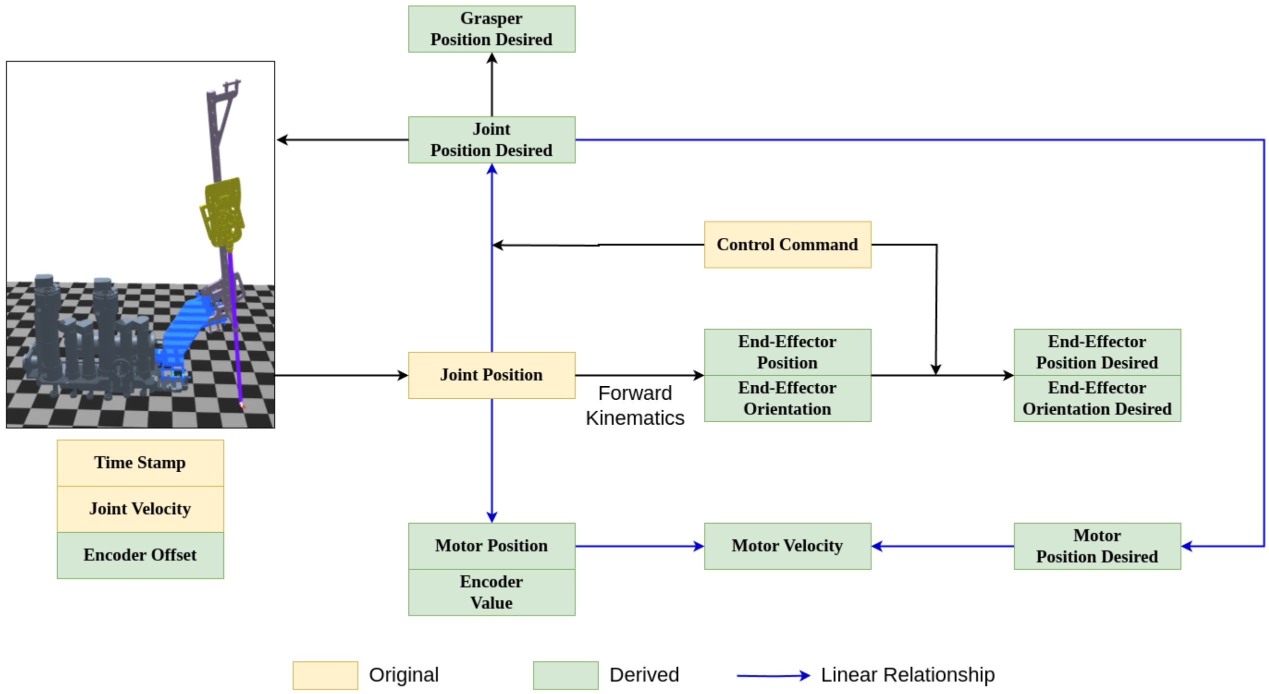

Fig. 2. Data flow and structural dependencies in the simulated `raven_state` message. This diagram replicates the relationship between core state variables in the real RAVEN-II system. Measurements (e.g., joint positions) are collected from the simulator and used to derive higher-level state features. Motor poses are linearly computed from joint angles, while end-effector pose and orientation are obtained through nonlinear forward kinematics. Control commands (e.g., joint or Cartesian deltas) are not formally included in the state but are shown to illustrate how they influence desired values. The complete state is published at 1000 Hz in standardized ROS message format, enabling real-time monitoring and data-driven experimentation.

the RAVEN-II surgical robot, within the AMBF simulation environment [7]. This implementation replicates the real-world RAVEN-II's data integration method, providing researchers with detailed state information essential for effective research and experimentation.

The AMBF `raven_state` message integrates multiple essential robot state parameters, including real-time joint poses, motor poses, encoder values, and end-effector states, as shown in Fig. 2. Specifically, the message incorporates both original measurements directly obtained from the simulator and derived metrics calculated from these raw measurements. For instance, motor positions are linearly derived from joint pose, whereas end-effector poses are computed from joint positions via forward kinematics, a nonlinear relationship.

The completed `raven_state` structure is encapsulated into standardized ROS messages and published at a consistent frequency of 1000 Hz. This high-frequency publication ensures robust synchronization with control commands and supports real-time monitoring and control strategies commonly utilized by researchers in the surgical robotics community [6]. Such detailed real-time feedback significantly facilitates advanced research tasks, including motion planning, and the development of sophisticated robotic manipulation techniques [7] [8].

C. Low-Level Motion Constraints and Interpolation Logic

To further replicate the control behavior of the physical RAVEN-II robot, our simulation framework incorporates low-level joint constraints and interpolation algorithms consistent with those used on the real system. Specifically, we enforce joint-specific range-of-motion limits across all seven actuated degrees of freedom. These constraints are modeled directly after hardware-level thresholds implemented in the original robot's firmware [9].

When control commands issued to the simulator exceed defined joint limits, the system issues a warning and prevents further motion beyond safe bounds. This mechanism provides realistic boundary handling and ensures that simulation results reflect the physical safety constraints of the actual RAVEN-II platform.

Additionally, to simulate smooth and realistic joint motion between desired poses, we implemented the same interpolation algorithm employed by the physical RAVEN-II. These components ensure that the motion logic of the simulated RAVEN-II remains faithful to its real-world counterpart, thereby enhancing the fidelity and reliability of simulation-based experiments.

III. Results

To assess the effectiveness of the proposed integration, we qualitatively compare the standard AMBF interface [5] with our extended CRTK-compliant architecture. This evaluation considers four key dimensions, as shown in TABLE I.

To further underscore the differences between our simulation and the original AMBF implementation, as summarized in TABLE I. Our approach clearly facilitates the sim-to-real transition. n particular, by integrating a richer data pipeline

TABLE I
COMPARISON BETWEEN AMBF TOPICS / API AND AMBF CRTK FRAMEWORK

Aspect	AMBF RAVEN-II	RAVEN-II AMBF CRTK Framework (Proposed)
Feedback and Data Provided	Multiple ROS topics provide basic joint-level data (e.g., position, velocity, effort), typically queried via API calls (`get_pos`, `get_rpy`, etc.)	A unified RAVEN State topic that publishes comprehensive feedback including joint pose, end-effector pose, motor pose, encoder values, and additional state details
Update Frequency	Variable update rates without strict synchronization to real-world RAVEN-II control loops	Consistently synchronized with the 1000Hz control loop, ensuring high-frequency and real-time updates
Data Detail	Limited to standard simulation outputs	Extended feedback including additional details such as end-effector position/orientation in the base link frame
Integration Method	Directly manipulates the AMBF simulator through a specific API (e.g., `set_joint_pos`, `set_pos`, `set_rpy`)	Implements a standardized CRTK command set to control the simulated RAVEN-II (e.g., `servo_jr`, `servo_cr`)

and a more strictly synchronized control loop, our CRTK-compliant framework provides a more accurate representation of the real RAVEN-II system, thereby simplifying the transfer from simulation to real-world application. The richer data pipeline and control loop consistency support better testing, reproducibility, and compatibility with higher-level applications [8].

IV. CONCLUSIONS AND FUTURE WORK

In this paper, a simulation-control framework is developed that brings together the AMBF simulator and the standardized control interface of the CRTK API to emulate the RAVEN-II surgical robot in a real-time environment. We developed a Python-based control loop operating at 1000 Hz and implemented CRTK-compatible topics for joint and Cartesian control and feedback. The framework allows researchers to interact with the virtual RAVEN-II as if it were the physical robot, all while receiving detailed and comprehensive feedback, real-time state updates via ROS. By integrating the strengths of AMBF and CRTK, our framework offers researchers a realistic, standardized, and safe environment for surgical robotics experimentation. This approach not only accelerates refinement of advanced robotic control method but also significantly reduces barriers related to cost, safety, and hardware availability, paving the way for broader adoption and innovation on robotic surgical applications.

To further improve the system's utility and realism, several enhancements are envisioned. First, we plan to expand the published RAVEN state to include more informative metrics, such as Jacobian matrices, which are essential for analytical control and learning-based strategies [10]. Coupled with this, we intend to incorporate machine learning models to simulate motor torque behavior, increasing the physical realism of the system. Second, we aim to extend the current single-arm support to include the right arm, enabling bimanual control scenarios that reflect more complex surgical procedures. Third, ongoing refinement of Cartesian control mechanisms will help improve end-effector responsiveness and precision. Lastly, we will continue to optimize the framework's stability and efficiency, ensuring it can handle high-frequency operations and scale to more complex research settings with minimal latency or degradation in performance.

REFERENCES

[1] B. S. Peters, P. R. Armijo, C. Krause, *et al.*, "Review of emerging surgical robotic technology," *Surgical endoscopy*, vol. 32, no. 4, pp. 1636–1655, 2018.

[2] D. B. Camarillo, T. M. Krummel, and J. K. Salisbury Jr, "Robotic technology in surgery: past, present, and future," *The American Journal of Surgery*, vol. 188, no. 4, pp. 2–15, 2004.

[3] L. Grespan, P. Fiorini, and G. Colucci, *The route to patient safety in robotic surgery*. Springer, 2019.

[4] Y.-H. Su, A. Munawar, A. Deguet, *et al.*, "Collaborative robotics toolkit (crtk): Open software framework for surgical robotics research," in *2020 Fourth IEEE International Conference on Robotic Computing (IRC)*, pp. 48–55. IEEE, 2020.

[5] A. Munawar, Y. Wang, R. Gondokaryono, *et al.*, "A real-time dynamic simulator and an associated front-end representation format for simulating complex robots and environments," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1875–1882, Nov 2019.

[6] B. Hannaford, J. Rosen, D. W. Friedman, *et al.*, "Raven-ii: An open platform for surgical robotics research," *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 4, pp. 954–959, 2013.

[7] H. Peng, A. Lewis, and B. Hannaford, "Ablation study on features in learning-based joints calibration of cable-driven surgical robots," *arXiv preprint arXiv:2302.04999*, 2023.

[8] H. Peng, A. Lewis, Y.-H. Su, *et al.*, "Efficient data-driven joint-level calibration of cable-driven surgical robots," *arXiv preprint arXiv:2408.01604*, 2024.

[9] H. King, S. Kosari, B. Hannaford, *et al.*, "Kinematic analysis of the raven-ii research surgical robot platform," *University of Washington, Tech. Rep. UWEETR-2012-0006*, 2012.

[10] B. E. Jackson, J. H. Lee, K. Tracy, *et al.*, "Data-efficient model learning for control with jacobian-regularized dynamic-mode decomposition," in *Conference on Robot Learning*, pp. 2273–2283. PMLR, 2023.