SlicerRoboTMS: An Open-Source 3D Slicer Extension for Robot-Assisted Transcranial Magnetic Stimulation

1st Wenzhi Bai
Department of Electrical and Electronic Engineering
University of Manchester
Manchester, U.K.
wenzhi.bai@manchester.ac.uk

2nd Yituo Guo
Department of Electrical and Electronic Engineering
University of Manchester
Manchester, U.K.
yituo.guo@postgrad.manchester.ac.uk

3rd Bhaskar Basu
Rehabilitation Medicine
Manchester University NHS Foundation Trust
Manchester, U.K.
bhaskar.basu@mft.nhs.uk

4th Andrew Weightman
Department of Mechanical and Aerospace Engineering
University of Manchester
Manchester, U.K.
andrew.weightman@manchester.ac.uk

5th Zhenhong Li
Department of Electrical and Electronic Engineering
University of Manchester
Manchester, U.K.
zhenhong.li@manchester.ac.uk

***Abstract*—Robot-assisted Transcranial Magnetic Stimulation (Robo-TMS) is an image-guided robotic intervention that enhances the accuracy and reproducibility of conventional Transcranial Magnetic Stimulation (TMS), a widely used non-invasive brain stimulation procedure in clinical treatment and neuroscience research. Despite its potential, the development of Robo-TMS remains challenging due to the need for multidisciplinary expertise spanning medical imaging, computer vision, and robotics. This paper presents *SlicerRoboTMS*, an open-source 3D Slicer extension that provides a unified interaction infrastructure for Robo-TMS research. By leveraging 3D Slicer's medical image computing and visualisation capabilities, the extension supports Magnetic Resonance Imaging (MRI)-based neuronavigation and interfaces with robotic systems through standardised communication protocols and configurable system descriptions. An example integration is presented to demonstrate how SlicerRoboTMS can be incorporated into a representative Robo-TMS workflow. Designed to support diverse hardware configurations and rapid prototyping, SlicerRoboTMS lowers the barrier to entry and facilitates reproducible and extensible research in Robo-TMS. The extension is available at https://github.com/OpenRoboTMS/SlicerRoboTMS.**

***Index Terms*—3D Slicer, Transcranial Magnetic Stimulation (TMS), Robo-TMS, Medical Robot, Image-Guided Robotic Interventions, Robot-Assisted Interventions**

I. INTRODUCTION

Transcranial Magnetic Stimulation (TMS) is a non-invasive neuromodulation procedure that uses rapidly changing magnetic fields to induce electric currents in targeted brain regions [1]. It is clinically validated for the treatment of neuropsychiatric disorders such as major depressive disorder and obsessive-compulsive disorder, and has demonstrated growing potential in neurorehabilitation for conditions including stroke and Parkinson's disease [2]–[4]. In addition, TMS is widely used for functional brain mapping in both clinical and research contexts, including pre-surgical planning for brain tumour resection [5], [6] and neuroscientific investigations of cortical organisation [7], [8]. To improve targeting accuracy and procedural consistency, Robot-assisted Transcranial Magnetic Stimulation (Robo-TMS) integrates neuronavigation with a robotic manipulator to position the stimulation coil relative to subject-specific anatomy [9]. By compensating for operator variability and subject head motion, Robo-TMS can enhance therapeutic reliability and procedural efficiency compared to manual TMS [10]. As a result, Robo-TMS has emerged as a rapidly growing research area with the potential to transform conventional TMS workflows.

Despite this promise, the development of Robo-TMS systems remains challenging. Effective Robo-TMS requires tight integration of medical image data, spatial tracking, and robotic control. However, Magnetic Resonance Imaging (MRI)-based neuronavigation, which is essential for target identification, is difficult to incorporate into robotic development environments such as the Robot Operating System (ROS) [11]. While ROS provides widely used software libraries and tools for robotics, it does not natively support image-guided clinical workflows. This disconnect increases system complexity and raises the barrier to entry for system development. 3D Slicer is a widely adopted open-source platform for medical image computing and image-guided therapy research [12]. It provides comprehensive tools for medical image processing and visualisation, together with extensible interfaces for integrating external hardware and software systems. In addition, several middleware frameworks have been proposed to facilitate the integration of medical imaging workflows with ROS [13]–[15]. These capabilities have supported a wide range of image-guided interventions [16]–[19], and recent work has demonstrated feasibility for TMS-related research through extensions such as SlicerTMS [20]. However, to the authors' knowledge, no open-source Robo-TMS extension currently exists within the 3D Slicer ecosystem, which limits access to advanced medical image computing capabilities and hinders broader participation.

Existing Robo-TMS systems [21]–[25] are often device-specific and rely on proprietary software, further fragmenting research efforts. To address this gap, we present *SlicerRoboTMS*, an open-source 3D Slicer extension that provides a unified interaction infrastructure for Robo-TMS research. The proposed extension supports the integration of MRI-based neuronavigation with robotic systems through standardised interfaces, reducing system re-engineering and facilitating extensible research across diverse Robo-TMS applications.

The contributions of this paper are as follows:

- the first open-source 3D Slicer extension designed specifically for Robo-TMS research;
- a modular and configurable extension that supports extensible Robo-TMS applications;
- an example integration illustrating the practical use of the proposed extension.

The proposed extension supports open, extensible, and rapid prototyping of Robo-TMS systems, facilitating reproducible and collaborative research in this emerging field.

II. EXTENSION OVERVIEW

This section provides an overview of the SlicerRoboTMS extension within a Robo-TMS system, focusing on its design considerations, integration context, and user interaction.

A. Design Considerations

Robo-TMS requires tight integration of MRI data, spatial tracking, and robotic control within a unified software environment. SlicerRoboTMS is designed with the following considerations to support flexible, extensible, and reproducible research.

1) Openness and extensibility: SlicerRoboTMS provides fully open-source code together with example integration implementations, enabling transparent inspection, modification, and extension of the software. This supports rapid prototyping, reproducible research, and community-driven development.

2) Modularity and compatibility: The software follows the modular design principles of 3D Slicer to ensure compatibility with its existing ecosystem. By leveraging established 3D Slicer data structures and communication protocols, the extension integrates with existing tools and supports seamless incorporation into broader research workflows.

3) Hardware agnosticism: Standardised communication interfaces and configuration-based hardware descriptions are used to decouple image-guided workflows from hardware-specific implementations, allowing the extension to interface with a wide range of external devices.

4) Real-time performance: SlicerRoboTMS employs configurable visualisation update rates and event-driven data exchange to support real-time updates of robotic poses, target locations, and anatomical context during system operation.

B. Integration Architecture Overview

SlicerRoboTMS is organised as an extension comprising three functional modules: a calibration module, a registration module, and a navigation module. These modules operate independently within 3D Slicer while collectively supporting a typical Robo-TMS workflow. To place the extension in context, Fig. 1 illustrates the layered system architecture in which SlicerRoboTMS operates, separating medical image-based interaction, system coordination, and physical device execution.

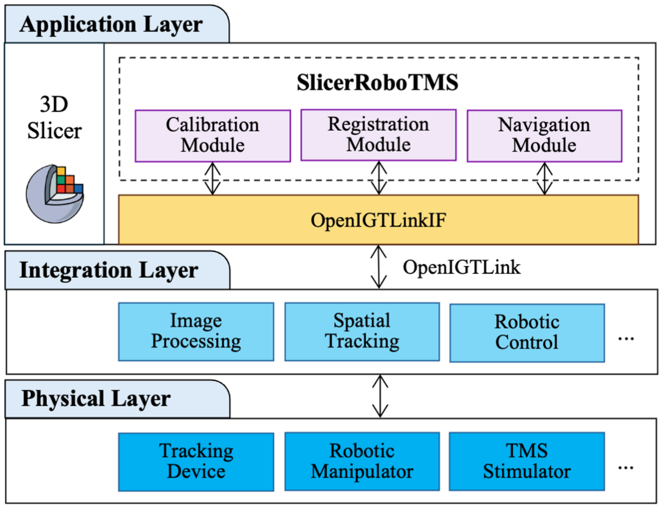
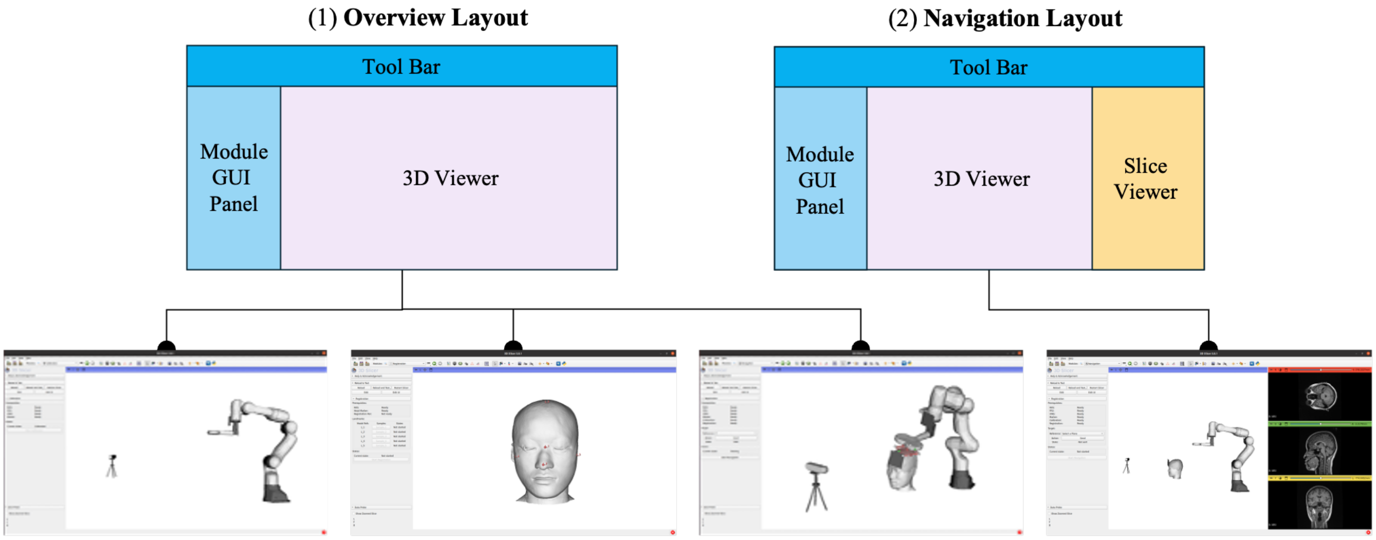


Fig. 1. Contextual system architecture illustrating the role of SlicerRoboTMS within a generic Robo-TMS setup. The architecture adopts a layered design comprising an application layer based on 3D Slicer, an integration layer responsible for system coordination, and a physical layer including tracking devices, robotic manipulators, and TMS stimulators. SlicerRoboTMS operates as an extension within the application layer and provides three functional modules. Standardised communication interfaces enable real-time data exchange between layers; in particular, OpenIGTLink is used for communication between the application and integration layers via the OpenIGTLink Interface (OpenIGTLinkIF) module in 3D Slicer.

Within this architecture, SlicerRoboTMS operates exclusively in the application layer and provides the primary interface for system monitoring, MRI-based neuronavigation, and user interaction. The application layer is built on 3D Slicer, which offers robust support for medical image visualisation, spatial transforms, and interactive planning, making it well suited for neuronavigation-driven Robo-TMS workflows. Implementing SlicerRoboTMS as a native 3D Slicer extension allows the reuse of established data structures and visualisa-



(3) Task 1 - Calibrating (4) Task 2 - Registrating (5) Task 3 - Tracking (6) Task 4 - Targeting

Fig. 2. User interaction and visualisation layouts provided by SlicerRoboTMS. Two primary layouts are supported: (1) a system overview layout with a 3D view for global monitoring, and (2) a navigation layout combining a 3D view with orthogonal slice views for MRI-guided neuronavigation. Task-specific views corresponding to different modules and workflow stages are illustrated in (3)-(6).

tion capabilities, while enabling communication with external components through standardised interfaces.

The integration and physical layers are included to contextualise how SlicerRoboTMS interfaces with external system components. The integration layer coordinates image processing, spatial tracking, and robotic control, and communicates with the application layer using OpenIGTLink [15] to maintain loose coupling between medical image computing and device-specific control. The physical layer comprises tracking devices, robotic manipulators, and TMS stimulators, which are accessed indirectly through the integration layer. This layered architecture supports flexible and extensible Robo-TMS research workflows while allowing SlicerRoboTMS to remain independent of specific hardware configurations.

C. *User Interaction and Visualisation*

Effective Robo-TMS interaction requires both real-time feedback on system state and accurate MRI-based neuronavigation. SlicerRoboTMS addresses these requirements through a layout-based Graphical User Interface (GUI) design that presents task-relevant information while maintaining a clear and consistent user experience, as illustrated in Fig. 2.

The GUI is implemented within the 3D Slicer environment and builds directly on its native view layout framework, which supports synchronised 3D visualisation and orthogonal medical image slices. SlicerRoboTMS customises these existing layouts to support different workflow tasks. A system overview layout, derived from the standard 3D-only layout in 3D Slicer, supports global monitoring by visualising the robotic manipulator, stimulation coil, tracking data, and anatomical models. A navigation layout, adapted from the conventional widescreen layout, combines a 3D view with orthogonal MRI slice views to enable precise target localisation relative to subject-specific anatomy.

During calibration, registration, and navigation, the interface selects the layout most appropriate for the task and provides tailored module panels to present essential parameters and system states. By leveraging and extending native 3D Slicer visualisation capabilities, SlicerRoboTMS enables intuitive user interaction while remaining consistent with established Robo-TMS workflows.

III. EXTENSION IMPLEMENTATION

This section details the implementation of the SlicerRoboTMS extension, focusing on data management and interface mechanisms for integration with external components.

A. *Communication Protocol*

SlicerRoboTMS serves as the interaction hub for image-guided Robo-TMS workflows, exchanging spatial information, system states, and user-defined commands with the integration layer. To enable real-time data exchange while maintaining loose coupling between software components, OpenIGTLink is adopted as the communication protocol between the application and integration layers. OpenIGTLink is a TCP/IP-based protocol designed for real-time data exchange in image-guided medical applications and is supported in 3D Slicer via the OpenIGTLink Interface (OpenIGTLinkIF) module. This enables SlicerRoboTMS to interface with tracking and robotic systems without embedding hardware-specific dependencies.

All data in 3D Slicer are managed using the Medical Reality Modelling Language (MRML) [12], where each data entity is represented as a node within a scene and updated through event-driven mechanisms. In SlicerRoboTMS, MRML nodes are mapped to standard OpenIGTLink message types to support different aspects of the Robo-TMS workflow:

- `TRANSFORM` messages transmit real-time spatial information, such as tracking data and robot poses, enabling dynamic visualisation;
- `POLYDATA` messages represent geometrical objects, such as registration fiducials;
- `STATUS` messages convey system states and execution commands for monitoring and user interaction.

Each message is assigned a unique identifier to ensure consistent handling within the communication stream. This design preserves a clear separation between image-guided interaction in the extension and hardware-specific control logic implemented in the integration layer, while supporting flexible and extensible system integration.

B. Transform Management

Accurate and consistent spatial relationships are essential for Robo-TMS, as dynamic visualisation depends on continuous updates of tracking and robotic pose information. In the overall workflow, real-time spatial data are maintained in the integration layer and transmitted to the application layer via OpenIGTLink, where visualisation is driven by transform updates within the 3D Slicer scene. SlicerRoboTMS leverages 3D Slicer's native transform hierarchy to manage these spatial relationships in a manner consistent with image-guided workflows.

Within SlicerRoboTMS, all visualised entities are represented in the MRML scene using separate model and transform nodes to define geometry and relative pose, respectively. Model geometry is specified using stereolithography (STL) files, while corresponding transform nodes encode spatial relationships between coordinate frames using linear transformation matrices. These transforms are organised into a hierarchical tree structure, allowing updates at any level to propagate automatically and preserve spatial consistency. The default transform hierarchy is initialised from a Unified Robot Description Format (URDF) configuration file, which specifies model paths, parent-child relationships, frame identifiers, and visual properties. During operation, updated transforms received via OpenIGTLink modify the associated transform nodes and trigger scene update events, enabling real-time and coherent visualisation of the robot, stimulation coil, and tracked anatomy.

C. Scene Integration

In SlicerRoboTMS, the MRML scene provided by 3D Slicer is used as a unified data container to manage imaging data, models, spatial relationships, and interaction states within the extension. The scene includes MRI volumes, model nodes representing physical components, and transform nodes defining spatial relationships among these components. Together, these elements form a coherent virtual representation of the Robo-TMS setup, enabling consistent visualisation of anatomy, tracking data, and robotic pose within a single coordinate framework. Each functional module in SlicerRoboTMS initialises its scene from predefined configurations, while real-time states are maintained in the integration layer and synchronised with the extension during operation. Model geometries and transform hierarchies are initialised from URDF configuration files in Extensible Markup Language (XML) format, and system-level parameters are specified using Python configuration files.

Scene updates are performed through a periodic synchronisation mechanism driven by a Qt timer, which triggers updates at a configurable rate of 30 Hz. In the present setup, this update rate supports real-time visualisation and is comparable to the refresh frequency of commonly used optical tracking systems. Incoming spatial updates modify the corresponding transform nodes in the scene and trigger event-driven updates to ensure responsive visual feedback. All models and transforms are defined and loaded in the RAS (Right, Anterior, Superior) coordinate system to align with 3D Slicer's spatial conventions and common robot development environments, thereby reducing coordinate ambiguity and simplifying integration.

IV. EXAMPLE INTEGRATION

This section presents an example integration to demonstrate how SlicerRoboTMS can be integrated into an experimental Robo-TMS system.

A. Experimental Setup

An example Robo-TMS setup, shown in Fig. 3, is implemented to illustrate the practical use of SlicerRoboTMS.

Spatial tracking is performed using an Intel RealSense D455 camera and customisable fiducial tags (AprilTags [26]) attached to the robotic end-effector, head phantom, and a registration stylus. The colour camera operates at a resolution of 1280×800 pixels and a frame rate of 30 Hz, enabling real-time pose estimation. A Franka Research 3 robotic manipulator [27] positions the TMS coil relative to the 3D-printed head phantom. The setup runs on a host computer equipped with an Intel Core i9 processor and 64 GB of memory, using Ubuntu 20.04 with real-time kernel patches and ROS Noetic. This configuration represents a typical laboratory environment and is used to demonstrate the integration and operation of SlicerRoboTMS.

B. Integration Details

The integration architecture of the example setup is illustrated in Fig. 4.

In the example integration, system configurations and hardware descriptions are defined using configuration files to initialise SlicerRoboTMS with the appropriate models, transforms, and parameters. The integration layer is implemented using ROS, which provides a modular framework based on distributed processes, referred to as nodes. ROS topics are used for real-time data streaming, and the `tf` library is employed to manage spatial transforms between coordinate frames. Communication between the ROS-based integration layer and SlicerRoboTMS is achieved using the ROS-IGTL-Bridge [14],

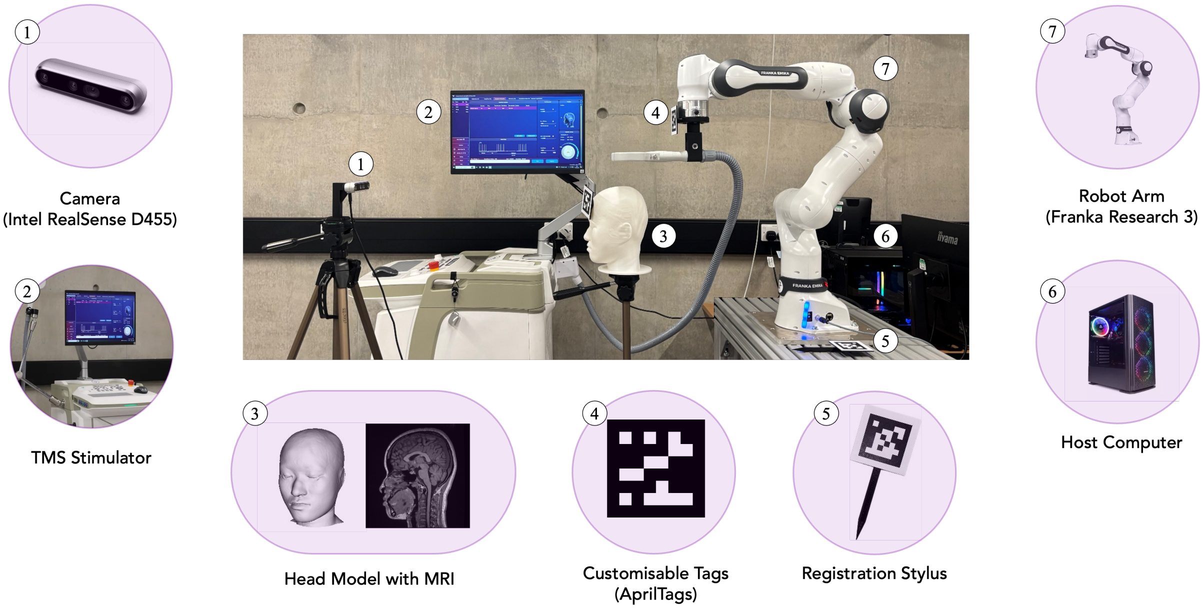


Fig. 3. Experimental setup used for the example integration. The configuration includes: (1) an optical tracking camera (Intel RealSense D455), (2) a TMS stimulator, (3) a 3D-printed head phantom generated from subject-specific MRI data, (4) customisable fiducial tags (AprilTags), (5) a 3D-printed registration stylus with the fiducial tag, (6) a host computer, and (7) a robotic manipulator (Franka Research 3). Numbered annotations indicate the main components, which are shown in separate subfigures for clarity.

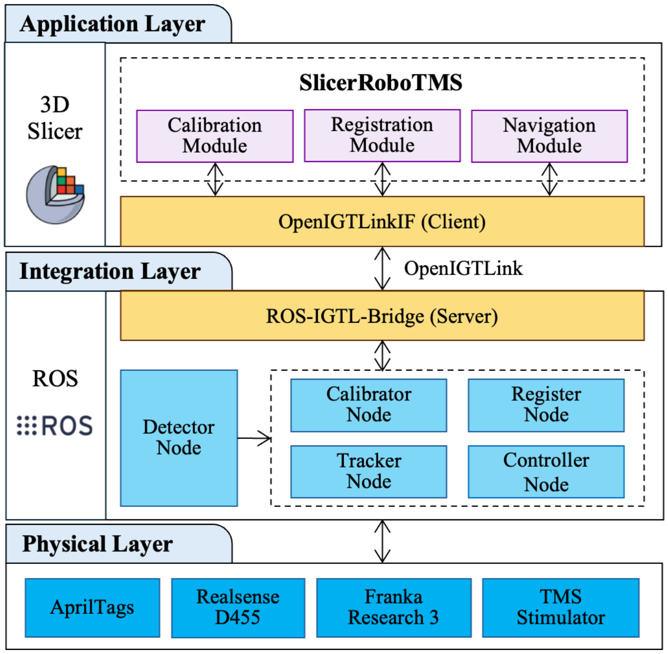


Fig. 4. Integration architecture of the example SlicerRoboTMS-based Robo-TMS setup. The integration layer, implemented using ROS, interfaces with physical devices and maintains real-time system states through distributed nodes. Bidirectional communication between the integration and application layers is achieved via the ROS-IGTL-Bridge (server) and OpenIGTLinkIF (client) using the OpenIGTLink protocol, enabling synchronised visualisation, navigation, and robotic control within 3D Slicer.

which translates ROS messages into OpenIGTLink messages transmitted over a TCP/IP connection to 3D Slicer. Conversely, messages generated by the extension are converted into ROS messages and published within the ROS network. This bidirectional translation enables the exchange of spatial data, system states, and execution commands while maintaining loose coupling between image-guided interaction and robotic control.

Within the integration layer, five independent ROS nodes are used to process and maintain system data. A detector node estimates fiducial poses from camera images. A calibrator node computes spatial relationships among different components. A register node establishes the transform between MRI image space and the physical head phantom. A tracker node converts stimulation targets defined in SlicerRoboTMS into corresponding robotic end-effector poses. Finally, a controller node executes robotic control based on the desired poses. Together, these nodes form a modular and extensible integration layer that enables the practical use of SlicerRoboTMS in a Robo-TMS workflow.

C. *Observations*

The example integration indicates that incorporating SlicerRoboTMS into a Robo-TMS setup primarily involves development within the integration layer. This layer encapsulates system-specific algorithms, including tracking, calibration, registration, and robotic control, and interfaces directly with physical devices. Consequently, algorithm development and hardware control can be carried out independently of medical image visualisation and user interaction, which are handled by SlicerRoboTMS within 3D Slicer.

Based on the example integration, three aspects are essential for effective use of SlicerRoboTMS:

- Communication between the integration layer and the extension should rely on the OpenIGTLink protocol as the sole interface for exchanging spatial transforms, target definitions, and system states.
- Using ROS as the foundation of the integration layer is a practical choice due to its mature robotic development ecosystem and native support for distributed processing and transform management. Tools such as the ROS-IGTL-Bridge facilitate bidirectional message mapping between ROS and 3D Slicer.
- System configuration should be aligned with the physical setup. Configuration files define model meshes, MRI volumes, and transform hierarchies to ensure correct scene initialisation and consistent spatial interpretation.

By providing ready-made modules for medical image visualisation and user interaction, SlicerRoboTMS enables researchers to avoid re-engineering the Robo-TMS interface and instead focus on developing and validating calibration, registration, and robotic control algorithms within a unified research framework.

V. CONCLUSION

This paper presents SlicerRoboTMS, an open-source 3D Slicer extension designed to support medical image–guided Robo-TMS research. The extension provides a unified interaction infrastructure for Robo-TMS, enabling MRI-based neuronavigation, real-time visualisation, and user interaction within the 3D Slicer ecosystem. The example integration demonstrates that researchers can implement specific calibration, registration, and robotic control algorithms externally while relying on SlicerRoboTMS for consistent visualisation and interaction, helping reduce development effort. By lowering the barrier to developing Robo-TMS systems, SlicerRoboTMS allows researchers to focus on methodological innovation rather than software infrastructure. The extension provides a practical foundation for reproducible research and supports future advances in Robo-TMS.

REFERENCES

[1] D. J. Edwards *et al.*, *A Practical Manual for Transcranial Magnetic Stimulation*. Cham: Springer Nature Switzerland, 2024.

[2] S. Vucic *et al.*, "Clinical diagnostic utility of transcranial magnetic stimulation in neurological disorders. Updated report of an IFCN committee," *Clinical Neurophysiology*, vol. 150, pp. 131–175, 2023.

[3] S. L. Cohen *et al.*, "A visual and narrative timeline of US FDA milestones for Transcranial Magnetic Stimulation (TMS) devices," *Brain Stimulation*, vol. 15, no. 1, pp. 73–75, 2022.

[4] S. Rossi *et al.*, "Safety and recommendations for TMS use in healthy subjects and patient populations, with updates on training, ethical and regulatory issues: Expert Guidelines," *Clinical Neurophysiology*, vol. 132, no. 1, pp. 269–306, 2021.

[5] N. Sollmann *et al.*, "Mapping of Motor Function with Neuronavigated Transcranial Magnetic Stimulation: A Review on Clinical Application in Brain Tumors and Methods for Ensuring Feasible Accuracy," *Brain Sciences*, vol. 11, no. 7, p. 897, 2021.

[6] A. F. Haddad *et al.*, "Preoperative Applications of Navigated Transcranial Magnetic Stimulation," *Frontiers in Neurology*, vol. 11, p. 628903, 2020.

[7] C. K. Kahl *et al.*, "Reliability of active robotic neuro-navigated transcranial magnetic stimulation motor maps," *Experimental Brain Research*, vol. 241, no. 2, pp. 355–364, 2023.

[8] A. Giuffre *et al.*, "Reliability of robotic transcranial magnetic stimulation motor mapping," *Journal of Neurophysiology*, vol. 125, no. 1, pp. 74–85, 2021.

[9] G. Fichtinger *et al.*, "Image-Guided Interventional Robotics: Lost in Translation?" *Proceedings of the IEEE*, vol. 110, no. 7, pp. 932–950, 2022.

[10] W. Bai *et al.*, "Robot-Assisted Transcranial Magnetic Stimulation (Robo-TMS): A Review," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 33, pp. 2606–2621, 2025.

[11] S. Macenski *et al.*, "Robot Operating System 2: Design, architecture, and uses in the wild," *Science Robotics*, vol. 7, no. 66, p. eabm6074, 2022.

[12] A. Fedorov *et al.*, "3D Slicer as an image computing platform for the Quantitative Imaging Network," *Magnetic Resonance Imaging*, vol. 30, no. 9, pp. 1323–1341, 2012.

[13] L. Connolly *et al.*, "SlicerROS2: A Research and Development Module for Image-Guided Robotic Interventions," *IEEE Transactions on Medical Robotics and Bionics*, vol. 6, no. 4, pp. 1334–1344, 2024.

[14] T. Frank *et al.*, "ROS-IGTL-Bridge: An open network interface for image-guided therapy using the ROS environment," *International Journal of Computer Assisted Radiology and Surgery*, vol. 12, no. 8, pp. 1451–1460, 2017.

[15] J. Tokuda *et al.*, "OpenIGTLink: An open network protocol for image-guided therapy environment," *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 5, no. 4, pp. 423–434, 2009.

[16] M. Sahu *et al.*, "ENTRI: Enhanced Navigational Toolkit for Robotic Interventions," *IEEE Transactions on Medical Robotics and Bionics*, vol. 6, no. 4, pp. 1405–1408, 2024.

[17] P. Moreira *et al.*, "In vivo evaluation of angulated needle-guide template for MRI-guided transperineal prostate biopsy," *Medical Physics*, vol. 48, no. 5, pp. 2553–2565, 2021.

[18] S. Choueib *et al.*, "Evaluation of 3D Slicer as a medical virtual reality visualization platform," in *Medical Imaging 2019: Image-Guided Procedures, Robotic Interventions, and Modeling*, B. Fei and C. A. Linte, Eds. San Diego, United States: SPIE, 2019, p. 38.

[19] T. Ungi *et al.*, "Navigated Breast Tumor Excision Using Electromagnetically Tracked Ultrasound and Surgical Instruments," *IEEE Transactions on Biomedical Engineering*, vol. 63, no. 3, pp. 600–606, 2016.

[20] L. Franke *et al.*, "SlicerTMS: Real-time visualization of transcranial magnetic stimulation for mental health treatment," in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2024*, M. G. Linguraru *et al.*, Eds. Cham: Springer Nature Switzerland, 2024, pp. 575–585.

[21] Z. Yang *et al.*, "Neuromodulation robot for precision TMS," *Aperture Neuro*, vol. 5, no. SI 1, 2025.

[22] P. M. Kebria *et al.*, "Haptically-Enabled Robotic Teleoperation for Transcranial Magnetic Stimulation (TeleTMS)," in *2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. Honolulu, Oahu, HI, USA: IEEE, 2023, pp. 2100–2105.

[23] A. Noccaro *et al.*, "Development and Validation of a Novel Calibration Methodology and Control Approach for Robot-Aided Transcranial Magnetic Stimulation (TMS)," *IEEE Transactions on Biomedical Engineering*, vol. 68, no. 5, pp. 1589–1600, 2021.

[24] R. Ginhoux *et al.*, "A custom robot for Transcranial Magnetic Stimulation: First assessment on healthy subjects," in *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. Osaka: IEEE, 2013, pp. 5352–5355.

[25] G. Pennimpede *et al.*, "Hot spot hound: A novel robot-assisted platform for enhancing TMS performance," in *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2013, pp. 6301–6304.

[26] M. Krogius *et al.*, "Flexible Layouts for Fiducial Tags," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 1898–1903.

[27] S. Haddadin, "The Franka Emika Robot: A Standard Platform in Robotics Research [Survey]," *IEEE Robotics & Automation Magazine*, vol. 31, no. 4, pp. 136–148, 2024.