Infinite Science
Publishing

Real-Time Haptic Testbed for Path-Following Control in Rehabilitation: System and Performance Characterization

S. Kadimisetty[*1], **S. Muheim**[1], **L. Chee**[1], **N. Gerig**[1], **G. Rauter**[1]

1. BIROMED-Lab, Department of Biomedical Engineering, University of Basel
* Corresponding author: Sri Sai Sathvik Kadimisetty, email: srisaisathvik.kadimisetty@unibas.ch

Abstract: Implementing advanced control strategies such as haptic guidance for path-following in body weight-supported rehabilitation systems requires validated, risk-free, real-time infrastructure. This paper presents the architecture and preliminary validation of a scaled haptic testbed integrating a Beckhoff CX2043 real-time computer with a low-cost haptic device (Novint Falcon). The system employs a distributed architecture: MATLAB/Simulink controllers compiled to TwinCAT3 PLC code executed on the embedded real-time platform, communicating with a multi-threaded C++ application interfacing the haptic device through USB. We characterize the system's timing performance, where the real-time controller works at a frequency of 1 kHz, with current end-to-end latencies of approximately 20 ms. Preliminary tests with sit-to-stand trajectories derived from motion capture data confirm functional path-following capability. This work establishes the foundational infrastructure for subsequent development and comparison of assist-as-needed control strategies on a safe test bed that can be translated to "the FLOAT" system (Reha-Stim Medtec AG, Schlieren, Switzerland).

I. Introduction

Developing and validating force-rendering controllers for different gait-related tasks in overground gait rehabilitation robots like "the FLOAT" (Reha-Stim Medtec AG, Schlieren, Switzerland) [1] presents significant challenges. These challenges are: i) ensuring safety for researchers and patients, ii) minimizing development costs, while iii) maintaining the required responsiveness for effective therapy. Haptic devices capable of rendering forces with high precision and low magnitude provide an ideal preliminary low risk testbed for such controllers. These haptic devices enable rapid iteration without exposing patients to untested and potentially unstable controller behavior. The critical requirement is that the testbed should reproduce the real-time constraints of the target system. In this case, this means deterministic timing and low-latency communication to ensure stable haptic rendering at 1 kHz or higher, as required in our case (Fig. 1).

This paper presents the architecture and preliminary validation of a real-time haptic testbed for safe user-in-the-loop path-following control research, where researchers interact with the system as a scaled proxy for patient-robot interaction. We use the TwinCAT3 real-time environment for control code execution, mimicking the FLOAT robot. For user interaction, a Novint Falcon haptic device is connected via USB (Fig. 2). Our primary contributions are: (1) a distributed safe-to-operate real-time architecture separating control execution from haptic device interfacing, (2) characterization of current system timing performance identifying optimization targets, and (3) demonstration of path realization using motion-capture-

derived trajectories for gait-related tasks such as sit-to-stand motions.
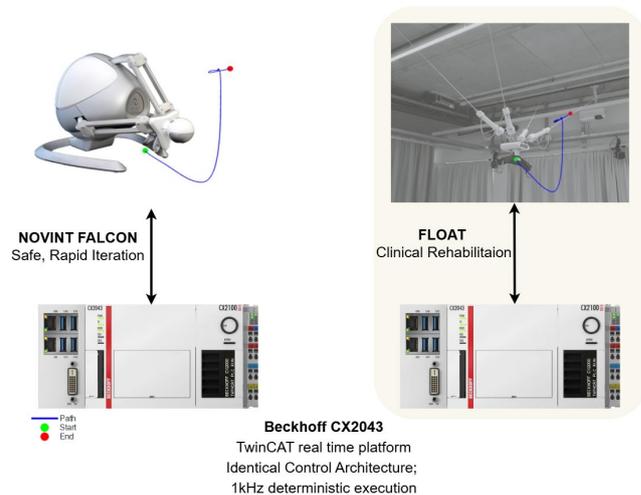

Fig. 1: System overview of the haptic testbed architecture. The Novint Falcon (left) provides a safe environment for rapid development and validation of controllers by rendering scaled rehabilitation trajectories realizable on the FLOAT (right).

II. Material and methods

Our testbed ensured deterministic control execution while accommodating the non-real-time constraints of standard operating systems interfacing with USB-based haptic hardware. Its components are as follows.

Real-Time PC: A Beckhoff CX2043 running TwinCAT3 (Beckhoff AG, Verl, Germany) served as the hard real-time control platform. Path-following controllers were

developed in MATLAB/Simulink (MathWorks, Natick MA, USA), compiled to TwinCAT3 real-time C++ code, and then integrated into a TwinCAT3 project. The TwinCAT3 runtime provides deterministic execution with configurable cycle times, targeting 1 ms for haptic control applications with consistent timing independent of external system load. TwinCAT3 is used within the FLOAT.

Development PC: A Windows 11 workstation (Intel Xeon W-2123, GeForce GTX 1060 and 32GB RAM) ran a multi-threaded C++ application responsible for haptic device communication and visualization. Three concurrent threads handled distinct functions: (1) a User Datagram Protocol (UDP) communication thread managing bidirectional data exchange with the real-time controller, (2) a haptics thread interfacing with the Novint Falcon using the Force Dimension SDK, and (3) a visualization thread rendering the end-effector position relative to the reference trajectory for real-time monitoring.

Haptic Device: The Novint Falcon provides a 3-DoF translational workspace of approximately $10 \times 10 \times 10$ cm^3 with force output up to 10 N at update rates up to 1 kHz.

II.I Real-Time Requirements

Stable haptic rendering requires control frequencies of 1 kHz or faster to prevent perceptible force discretization and ensure contact stability. The end-to-end latency, from position measurement at the Falcon to force application, directly impacts perceived stiffness and stability margins. Haptic devices can tolerate latencies up to ~6 ms before degradation becomes apparent [2], but minimizing latency improves transparency and enables higher virtual stiffness.

II.II Trajectory Generation

Reference trajectories from gait-related tasks such as sit-to-stand motions were derived from optoelectronic tracking data collected from healthy subjects. Raw position data was preprocessed: threshold-based filtering to remove clustered stationary points at trajectory endpoints, distance-based down sampling to ensure uniform point spacing along the path, and robust local regression smoothing (RLOESS) for noise reduction. The cleaned points were then fitted with quadratic B-splines to create a continuous parametric path representation [3]. Spatial scaling was applied to fit the trajectory within the Falcon workspace, preserving the kinematic profile characteristics of sit-to-stand movements (forward trunk lean, vertical rise, and stabilization phases).
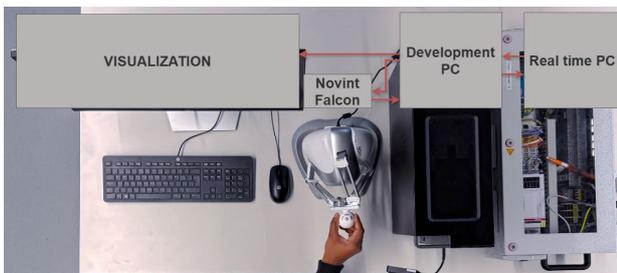

Fig. 2: Experimental setup

Additional test trajectories (straight lines, circles) were generated to characterize system behavior under controlled conditions before testing motion-derived paths.

III. Results and discussion

The TwinCAT3 runtime achieved stable execution at the target rate. Cycle time confirmed consistent 1 ms task execution with jitter below the measurement resolution of the TwinCAT3 diagnostics. The control algorithm computation, including the Simulink-generated C++ code, completed consistently within 1 ms.

End-to-end round-trip latency achieved an average of 20 ms, measured from UDP transmission of position data to receipt of corresponding force commands (Fig. 2). The target of <6 ms roundtrip delay was exceeded. Nevertheless, functional path-following could be successfully demonstrated with perceivable haptic and visual feedback. The latency bottleneck was caused by Windows non-deterministic thread scheduling and its network stack buffering behavior on the development PC.

IV. Conclusions

This work presents a real-time haptic testbed for developing path-following controllers targeting body weight support rehabilitation systems. The distributed system integrated a Beckhoff CX2043 real-time controller with a Novint Falcon haptic device via UDP, achieving 1 kHz control execution. Successful rendering of rehabilitation-relevant gait-related paths such as for sit-to-stand movements derived from motion capture data confirmed functional system integration.

However, current end-to-end latencies of approximately 20 ms do not meet the target latencies smaller than 6 ms required for transparent haptic rendering. However, controller logic and trajectory tracking could be validated. Future work will migrate the Falcon interface from Windows to Linux for achieving more deterministic performance and lower latency. Subsequent development will implement and compare path-following control strategies, including impedance control, potential field-based control, and hybrid approaches [3], ultimately supporting translation of assist-as-needed controllers to the FLOAT [1] for clinical rehabilitation.

REFERENCES

[1] M. Bannwart, D. Emst, C. Easthope, M. Bolliger, and G. Rauter, *Automated stand-up and sit-down detection for robot-assisted body-weight support training with the FLOAT*, in 2017 International Conference on Rehabilitation Robotics (ICORR), London: IEEE, July 2017, pp. 412–417.

[2] T. Hulin, R. Gonzalez Camarero, and A. Albu-Schaffer, *Optimal control for haptic rendering: Fast energy dissipation and minimum overshoot*, in 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, IEEE, Nov. 2013, pp. 4505–4511.

[3] G. Rauter, N. Gerig, R. Sigrist, H. Vallery, R. Riener, and P. Wolf, *Hybrid path controller enables haptic guidance along self-crossing paths*, in at - Automatisierungstechnik 2025, vol. 73 (issue 12), pp. 895–910.