Proceedings on Automation in Medical Engineering
Proc AUTOMED, 2026, Vol. 3, No 1, Article ID 2512, 2 Pages

Infinite Science Publishing

Virtual Technician: A multi-modal interface facilitating therapists' adoption of rehab robots

M. von Waldow[1*], R. Riener[1], M. Sommerhalder[2], P. Wolf[1]

[1] *Sensory-Motor Systems Lab, ETH Zurich, Zurich, Switzerland*
[2] *Bio-Inspired Robots for Medicine, University of Basel, Allschwil, Switzerland*
[*] *Corresponding author, email: maxole.vonwaldow@hest.ethz.ch*

Abstract: Rehabilitation robots support physiotherapy but therapeutic intention can be difficult to express as robotic parameter adjustments. A multi-modal interface, the Virtual Technician, is conceptualized to provide an intuitive mapping of intention to robot parameters. A demonstrator of the Virtual Technician based on a Large Language Model (LLM) showed that LLMs of ≥4 billion parameters can cope with abstract natural language commands to adjust typical parameters of rehabilitation robots.

I. Introduction

After brain injury, e.g. stroke, neuroplasticity allows the brain to reconfigure itself, meaning other parts of the brain can take over some functionality of the damaged parts [1]. The recovery is usually supported by therapists who guide the affected limbs during therapy. Rehabilitation robots facilitate physiotherapy and allow for more intensive training with more repetitions [2].

Versatile robots applicable to a wide variety of patient groups are complex to use and therapists must be specially certified and trained. They must adapt the robot's parameter settings (e.g. step length, step height, speed, support) to achieve the desired movement behavior in the patient for each exercise, often requiring trial and error and iterative adjustments. Improved interfaces can simplify the process of adapting parameters. We propose the Virtual Technician (VTec), an intuitive, multimodal interface to instruct the robot during therapy. An unintrusive, context-aware Large Language Model (LLM) helps adjust parameters through voice control and other modalities.

II. Methods

In this section we introduce the architecture and components of the VTec along with design choices. Common therapists' instructions can be categorized by abstraction and ambiguity levels. We build a demonstrator to evaluate the feasibility of the VTec handling these different instructions.

II.I. Conceptualization of the Virtual Technician

An LLM is suited as a core of the VTec as it can natively understand and interpret human dialogue. The goal is to translate therapists' intentions into robot parameters, while they retain decision authority. Natural instructions are diverse and differ depending on personal preference, experience and the situation. The LLM must be equipped to handle precise but also abstract, ambiguous and even inferred instructions that are not explicitly expressed. Using precise terminology with exact commands is possible, yet time inefficient and sometimes irritating to the patient. To enhance the LLM's capabilities to interpret more natural abstract intentions, it requires extensive situational awareness and added sensors, e.g. a microphone or camera.

Our current Virtual Technician (Figure 1) is largely based on voice input. Humans tend to shorten sentences and avoid repetitions by using referring expressions, e.g., "it" and "that". A history of recent dialogue is stored to extract these references for otherwise ambiguous instructions. Pre-session data is also stored, e.g., about the patient's pathology. This data can be useful, e.g., if the therapist refers to an affected limb without explicitly mentioning which one it is. Vision capabilities allow many insights into the situation. Combined with object detection and skeleton tracking, the LLM can understand references to objects, directions, and understand demonstrations the therapist performs. In future, the vision modality can be implemented by transcribing the output of dedicated tools for object detection and skeleton tracking, or directly by using Vision Language Models (VLM) [3]. Finally, a handheld physical input device may allow quickly selecting suggestions and confirming adjustments, where forming sentences is unnecessary or could irritate the patient.

In our demonstrator, LLM inference is run with the Ollama program [4] on a local server at the university. The server receives prompts and returns the updated parameters. The LLM generates valid processable JSON output by using a generation strategy that forces it to adhere to a fixed Pydantic [5] schema. The prompt sent to the LLM is created from a structured text template, merging context and user input. It consists of the system prompt, the history and finally the current input. The system prompt informs the LLM of the task and sends the validation rules and Pydantic output structure, both in JSON format. The history consists of a series of past inputs to the LLM and its response if successful. The current input is appended containing the transcribed voice input and parameter states.

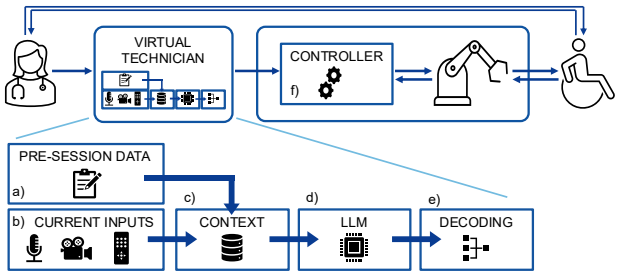

Figure 1: The Virtual Technician architecture as an input interface in robotic therapy. Pre-session data a) and multi-modal sensor inputs b) are aggregated in a context base c). The LLM d) parses it for recent commands and creates parameter adjustments. These are postprocessed e) for robustness, e.g., by safety rules (upper & lower bounds for value and change per instruction), and sent to the existing controller f). The LLM asks to clarify or confirm unclear or safety critical commands, resp.

II.II. Categorizing commands

We evaluated the effect of abstraction and ambiguity levels by testing increasingly imprecise commands:

C1) "Change the step length to 45cm."
C2) "Change it to 47cm."
C3) "Increase the step height by 2cm."
C4) "Do the same with the step length."
C5) "Reduce the speed just a bit."
C6) "This seems to be too fast."
C7) "That was a lot of work. Let's make it more relaxing now."

Command C1) explicitly mentioned the variable and its target value. C2) included an ambiguity, referencing the last changed variable as "it". C3) used an explicit increment and highlighted simple calculation capabilities. C4) included another ambiguity and referenced the last change instead of the variable. C5) implicitly mentioned the target value with an abstract size of the change. C6) made a statement with an indirect mention of a command. C7) represented an abstract intention achievable through multiple parameters with different solutions.

II.III. Demonstrator Analysis

We experimented in an imagined therapy scenario. The user posed as a therapist performing robotic physiotherapy on an imagined patient. The demonstrator updated simulated parameters.

Each run was repeated with LLMs of the Qwen3 series [6] with the sizes 0.6B, 1.7B, 4B, 8B, 14B and 32B parameters. Each LLM was queried in a fixed procedure with commands C1) – C7) in sequence as written text for repeatability and with zero temperature for a deterministic outcome.

III. Results and discussion

The commands were executed successfully with some exceptions (Figure 2). The smallest LLM, Qwen3-0.6B, made no changes in response to C4), C6) and C7). Qwen3-1.7B violated the rules by reducing the speed from 0.35 m/s to 0.33 m/s, a change smaller than the minimally allowed 0.05 m/s increment. Thus, the change was rejected by the VTec.

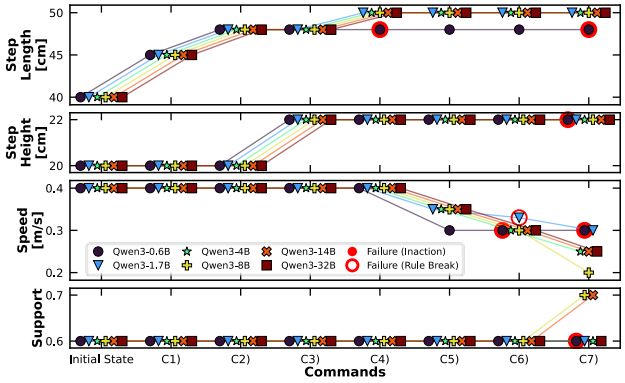

Figure 2: Parameter changes after each command.

Most LLMs interpreted "a bit" of change in C5) as a reduction by 0.05 m/s, whereas Qwen3-0.6B decreased by 0.1 m/s. In response to C7), Qwen3-8B and Qwen3-14B made multiple changes, reducing speed and simultaneously increasing support. The successful speed reductions were by 0.05 m/s, except for Qwen3-8B which reduced it by 0.1 m/s.

The commands C1), C2), and C3) with clear target values were handled well by all LLMs. Issues started with more abstract commands in combination with small LLMs of ≤1.7B parameters. The failures were either a command not being recognized or an increment rule being ignored. Vague commands can be interpreted differently as seen by the different adjustments to speed and support in C5) and C7).

IV. Conclusions

The demonstrator of the Virtual Technician shows how a rehabilitation robot can be adjusted in an intuitive, hands-free manner. Our experiments require a minimum LLM size of 4 billion parameters to handle commands with different levels of abstraction and ambiguity.

Vague quantifiers would require additional pre-computing or online learning for reliable interpretation. Additional input modalities would enable more interaction options with the Virtual Technician.

Author's Statement

Research funding: M.v.W. is funded by the Swiss State Secretariat for Education, Research and Innovation via the EU TAILOR Project, a Marie Skłodowska-Curie Actions doctoral network (101168724). Conflict of interest: Authors state no conflict of interest.

References

[1] T. H. Murphy and D. Corbett, "Plasticity during stroke recovery: from synapse to behaviour," *Nat. Rev. Neurosci.*, vol. 10, no. 12, pp. 861–872, Dec. 2009.
[2] C. G. Burgar, P. S. Lum, P. C. Shor, and H. M. Van der Loos, "Development of robots for rehabilitation therapy: The Palo Alto VA/Stanford experience," *J. Rehabil. Res. Dev.*, vol. 37, no. 6, pp. 663–674, 2000.
[3] J. Zhang, J. Huang, S. Jin, and S. Lu, "Vision-Language Models for Vision Tasks: A Survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 8, pp. 5625–5644, Aug. 2024.
[4] "Ollama." Accessed: Dec. 14, 2025. [Online]. Available: https://ollama.com
[5] S. Colvin *et al.*, *Pydantic Validation*. (Oct. 2025). Python. Accessed: Dec. 14, 2025. [Online]. Available: https://github.com/pydantic/pydantic
[6] Qwen Team, "Qwen3 Technical Report," May 14, 2025, *arXiv*: arXiv:2505.09388.