

ASSISTIVE ROBOT TELEOPERATION USING PHASE SWITCHING BEHAVIOR TREES

Mohamed Behery[†], Minh Trinh[‡], Christian Brecher[‡], and Gerhard Lakemeyer[†]

[†]Knowledge-Based Systems Group (KBSG), RWTH Aachen University, Germany
[‡]Laboratory for Machine Tools and Production Engineering (WZL), RWTH Aachen University, Germany



Overview

Assistance to people with Tetraplegia has recently gained a lot of traction. Brain Computer Interface (BCI) extracts signals from the user's brain to control a robot arm for performing different actions. This work represents Activities of Daily Living (ADLs) using Behavior Trees (BTs) whose inherent readability and modularity enables an end user to construct new activities using a simple interface. To achieve this, we augment BTs with Shared Control Action Nodes (SCANs), which guide a user's input on a trajectory facilitating task performance. We compare our work to the state of the art in ADL execution through robot teleoperation in terms of expressive power.

Background and State of the Art

BTs are a high-level robot control framework following a tree structure [1, 2]. The nodes can belong to two classes: execution and control flow. Execution starts by activating (*ticking*) the root node, which in turn starts ticking its children. After receiving a tick, a node will become active and return its status $S \in \{\mathcal{S}, \mathcal{R}, \mathcal{F}\}$; either Success, Running, or Failure respectively to its parent. BTs can have different node types as follows:

Node	Type	Behavior
Action	Execution	execute an action and return \mathcal{S} or \mathcal{F} depending on the action's success, or \mathcal{R} while it executes the action
Condition	Execution	check a condition and return \mathcal{S} or \mathcal{F} according to its status and never return \mathcal{R}
Sequence	Control flow	execute the children in order from left to right. They return \mathcal{S} iff <i>all</i> the children succeed, \mathcal{F} iff at least one fails, and \mathcal{R} otherwise
Selector	Control flow	implements a fallback behavior by executing the children in order from left to right and return \mathcal{S} iff <i>one</i> child succeeds, \mathcal{F} iff all the children fail, and \mathcal{R} otherwise
Decorator	Control flow	used to implement custom behavior (e.g., repeat n times) and manipulate the returned status of their child (e.g., negation)

The state of the art in assistive teleoperation for ADL execution is the Shared Control Template (SCT)[3].

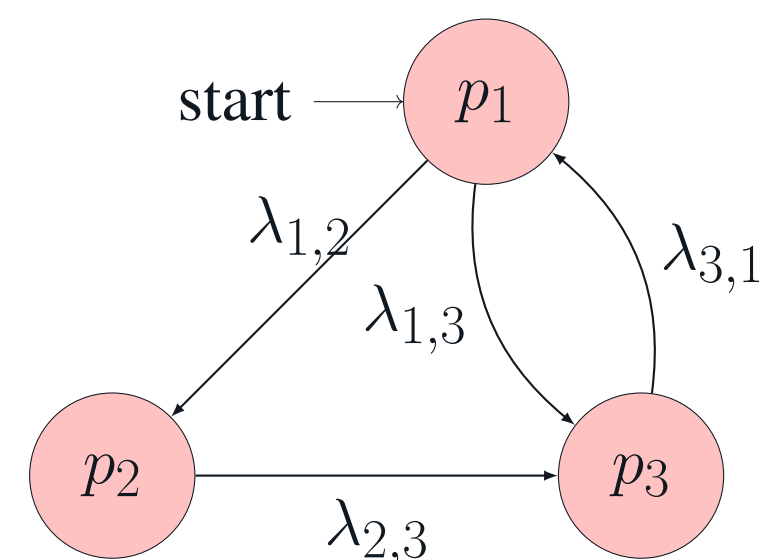


Fig. 1: SCTs represent an activity as a transition system where each state is a phase of the activity.

SCTs model activities as a labeled transition system $L = \langle P, \Lambda, E \rangle$, each state $p \in P$ applies a set of constraints on the user input. Transitions $\langle p, \lambda, p' \rangle \in E$ are conditioned on λ which can be geometric or symbolic.

Approach

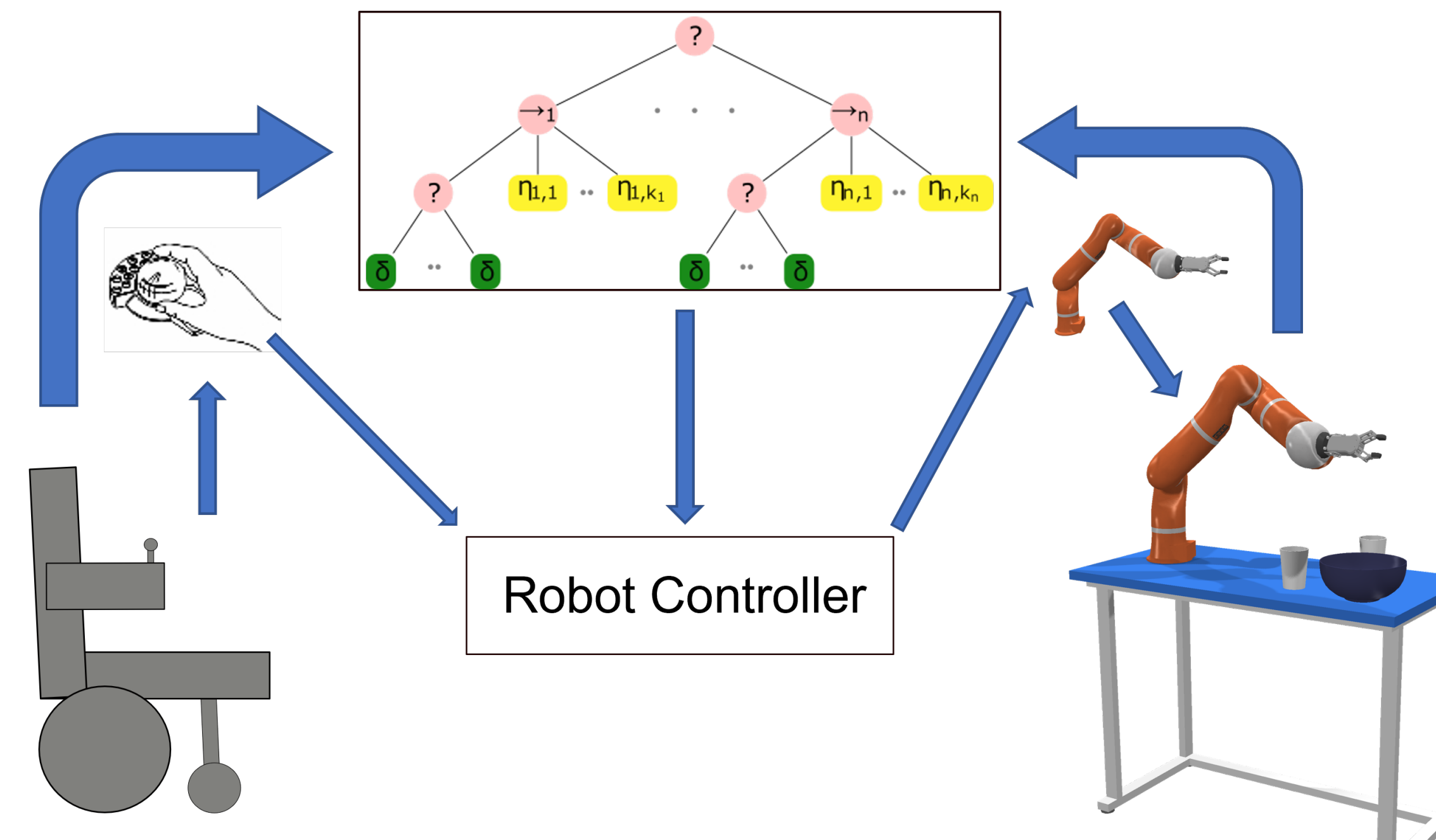


Fig. 2: An overview of the framework, a user can describe an ADL as a BT offline. The BT is used along with the user's input to assist with the teleoperation of the robot in real time.

We allow a user to describe an ADL using a BT. Each subtree of the BT is a phase of the activity describing the motion of the End Effector (EE). At runtime the tree projects the user's input on the predefined paths according to a set of predefined geometric and symbolic constraints.

Why Behavior Trees?

Behavior trees (BTs) offer a robot programming solution to this problem due to their **reactivity**, **modularity**, and **reusability** leading to several properties that appeal to this domain:

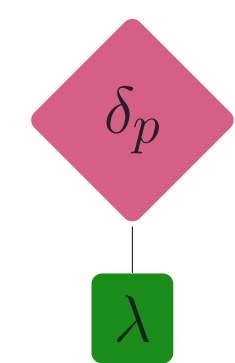
- Inherent safety
- Easy to program with little robotics experience
- Highly modular and reusable

Contribution:

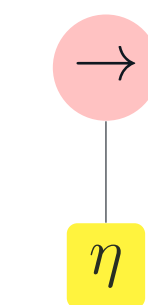
In this work, we introduce PHase Switching Teleoperation Behavior Trees (PHAST BTs) to represent and execute ADLs. We limit the structure of the BTs and introduce new nodes that allow it to project the user input to specific trajectories.

Proposed Nodes

We introduce two new nodes to BTs



δ -branches are decorator nodes that check the previously executed sequence



SCAN direct the user's input according to their function η

PHAST BTs

PHAST BTs introduce a SCAN that can use predefined auxiliary functions to direct the user's input. The structure of a PHAST BT is shown in Figure 5. The root is a fallback node, each of its children represent a phase of the activity. These subtrees are guarded by chained δ -branches to limit the transition between phases.

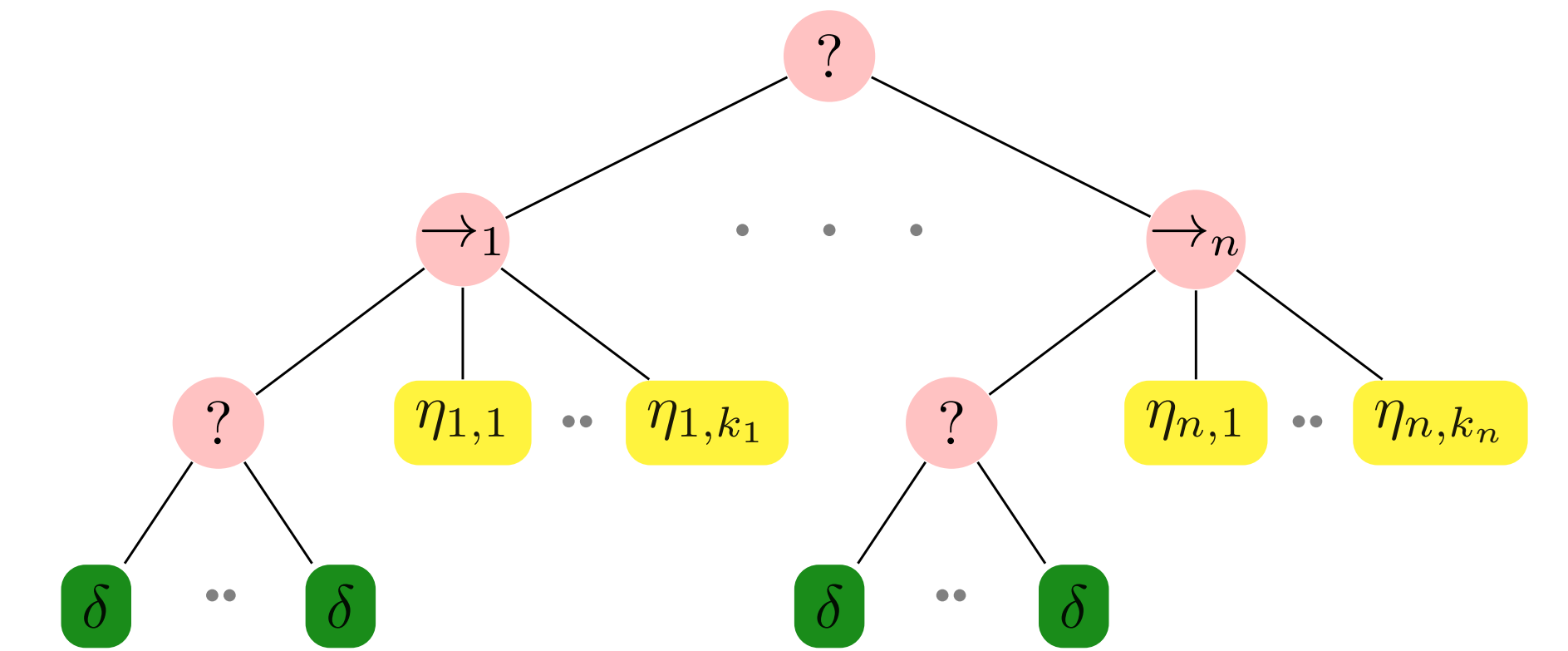


Fig. 5: PHAST BTs are limited to four levels and a fallback node as the root. Each branch in the second layer branch corresponds to an ADL phase. Each of which is a parent to a fallback node with δ -node children (green) representing the combination of previous states and geometric condition(s) followed by the SCANs (yellow).

Conclusion

PHAST BTs allow users to describe an ADL and execute it using shared autonomy, which maintain the user satisfaction and usability displayed in task performance found in approaches such as [3] by exploiting the modularity, **composability**, and **extensibility** of traditional BTs. We can maintain the ability to introspect the tree as well as the **trust and predictability** of the robot's actions by having a live display interface similar to the one mentioned in [3] displaying the states for each node in real time.

Acknowledgements

Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC-2023 Internet of Production – 390621612. This work is partially supported by the EU ICT-48 2020 project TAILOR (No. 952215).

References

- [1] Michele Colledanchise and Petter Ögren. *Behavior Trees in Robotics and AI*. CRC Press, July 2018. ISBN: 9780429950902. DOI: 10.1201/9780429489105. URL: <http://dx.doi.org/10.1201/9780429489105>.
- [2] Matteo Iovino et al. *A Survey of Behavior Trees in Robotics and AI*. 2020. arXiv: 2005.05842 [cs.LG].
- [3] Gabriel Quere et al. "Shared Control Templates for Assistive Robotics". In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. 2020, pp. 1956–1962. DOI: 10.1109/ICRA40945.2020.9197041.