A Comprehensive Investigation into the Use of Behavior Trees in the Coordination and Control of Swarm Robotics Systems

Professor: Nicola Conci
Student: Zaida Brito Triana MAT. 230400
Department of Industrial Engineering
University of Trento
zaida.britotriana@studenti.unitn.it
https://github.com/zaidabri/BehaviorTreeSwarmRobotics

TABLE OF CONTENTS

01.

ABOUT THE PROJECT

1. This paper discusses the use of Behavior Trees (BTs) in Swarm Robotics, which are a hierarchical decision-making structure for creating complex autonomous behaviors. It provides an overview of the theoretical foundations of BTs, the state of the art in the topic, highlighting the challenges and limitations in previous research, and the advantages and disadvantages compared to other systems such as Finite State Machines.
2. It also presents the methodology, framework, and results of an example code implementation in Swarm Robotics and concludes that BTs offer a flexible and scalable approach in this field.

The research question of this paper is presented as: **To what extent does the use of Behavior Trees improve the coordination and control of swarm robotics systems?**

Roadmap: (Colledanchise, 2017), (Colledanchise & Ögren, 2017-2022) (Jones, 2020)

Theoretical introduction

Behavior Trees were originally designed to describe the behavior of autonomous non-player characters (NPC) in computer games. Non-player characters, like autonomous robots, have the ability to respond and make choices in situations that are both complex and uncertain. Their definition and key concepts for their application were initially published by R. G. Dromey in 2001, (Dromey, 2001) but one of the first significant projects to implement behavior trees was in the development of the Halo 2 and Halo 3 video games (Isla, 2005) (Isla, 2008). Despite of the origin of Behavior Trees, lately, their potential in the robotics and control community has been gaining attention.

Swarm robotics is a field of robotics that studies the behavior of groups of robots, called swarms, that are able to coordinate their actions to accomplish tasks. The core principles of a robot's swarm are based upon the assumptions that the agents are able to freely move within an environment. The way the robot swarm behaves as a community is based on the interactions of each robot with its peers and the environment. Generally, swarms are made up of same type robots, though there are examples of swarms with different types of robots (Dorigo et al., 2013).

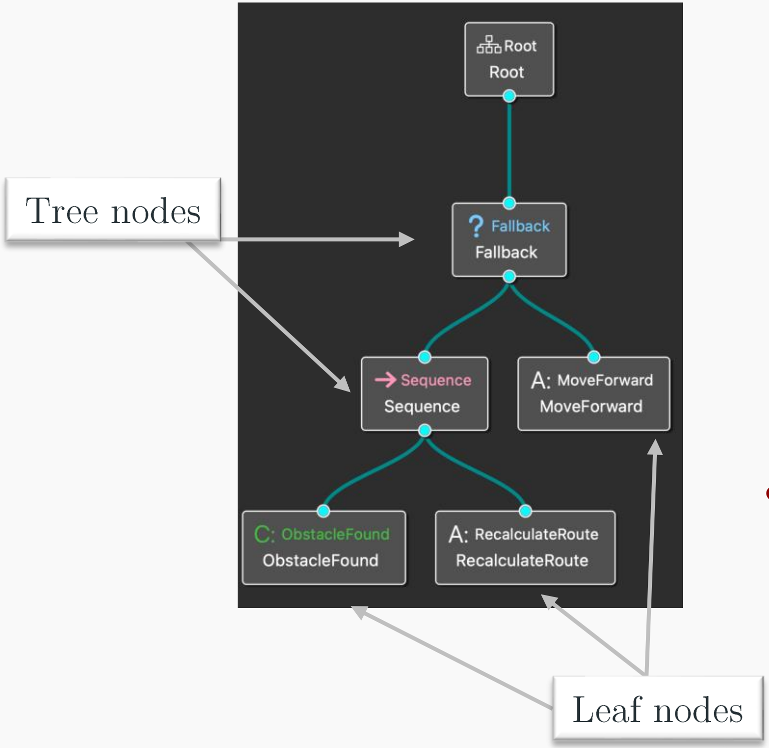
02.

Literature review

This section aims to provide an overview of the current state-of-the-art in the field of swarm robotics, followed by an examination of the theory and principles of behavior trees and a comparison with other commonly used methods in the coordination and control of swarm systems.

Behavior Tree theory

Behaviour Tree is a hierarchical tree of nodes. A behavior tree (BT) begins by running its root node, which sends out signals at a specific frequency called ticks. These ticks are then passed on to the root node's children. A node can only be executed if it receives these ticks. When the root of a tree sends a tick signal, it travels towards the leaves of the tree. Each Tree Node that receives the signal will then execute a callback function. This callback can return one of three possible results: success, failure, or running. If the running result is returned, it indicates that the action is still in progress and needs more time to be completed.
The actual commands of the behavior tree are found at the leaf nodes, and it is there that the tree interacts with the rest of the system.



Behavior Tree theory

Most common node types in Behavior Trees:

Node type	Symbol	RUNNING	SUCCESS	FAILURE
Sequence	→	If one child returns RUNNING	If **all** children return SUCCESS	I **one** child returns FAILURE
Fallback	?	If one child returns RUNNING	If **one** child return SUCCESS	If **all** child returns FAILURE
Parallel	⇉	Any other situation	If ≥ N children return SUCCESS	If > n - N return FAILURE
Decorator	◇	Custom	Custom	Custom
Action	▭	During completion	On clonclusion	If unfeasible
Condition	⬭	Never	If true	If false

Behavior Tree theory

Blackboard:

In a behavior tree, leaf nodes interact with the environment through a set of variables known as the blackboard. The blackboard mechanism allows for the introduction of events by providing memory to specialized tasks or by converting events to statuses at a lower level of the mission plan execution. These statuses can be stored in the blackboard and reset after processing by appropriate behavior tree tasks.

This serves as an illustration of the utilization of the Blackboard within the demonstration presented in this paper:

Blackboard		
KEY	**TYPE**	**VALUE**
{PickupStation}	string	"The chosen transfer station is 4"

Behavior Tree state of arts

Even though BTs have been around for more than a decade now, not too many years ago the complaint was that in the field of Behavior Trees "The available literature lacks the consistency and mathematical rigor required for robotic and control applications" (Marzinotto et al., 2014).
Here a review of the most recent papers evaluated for this research:

In 2021, Legarda.H conducted research on a method to collectively transport heavy objects using an industrial swarm in a simulated environment. The robots used behavior tree controllers optimized by genetic programming and decentralized communication for coordination. The study found the method to be successful in safely lifting and conveying multiple loads to a designated area, but more refinement is needed for real-world implementation.

Behavior Tree state of arts

Here a review of the most recent papers evaluated for this research:

Ligot et al. (2020) examines the potential, challenges, and limitations of using behavior trees in the automatic modular design of collective behaviors in swarm robotics. The study introduces an automatic design method called Maple, which combines predefined modules such as low-level behaviors and conditions into a behavior tree that encodes the individual behavior of each robot in the swarm. The study found that while behavior trees offer many appealing features, they do not consistently produce superior solutions compared to finite state machines. The research suggests that future studies should focus on examining the use of low-level behaviors as action nodes of behavior trees, and development of a tailored optimization algorithm that leverages the inherent modularity of behavior trees.

Colledanchise [2017] and Jones [2020] demonstrated as well that both BTs and FSMs are mathematically equivalent in terms of their capabilities.

Behavior Tree state of arts

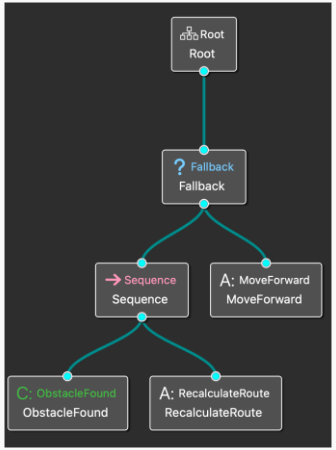
Here a review of the most recent papers evaluated for this research:

Jones (2020) and S. Jones et al. (2018) research focuses on designing optimal and readable systems for swarm robots. The main challenge is to design controllers for individual robots that will produce the desired collective behavior through interaction. The research studied current solutions that often rely on offline automatic discovery using artificial evolution of robot controllers, but found limitations such as the need for additional supporting infrastructure and evolved controllers being opaque and difficult to understand. The research addresses these issues by using behavior trees as the individual robot controller architecture, which are modular, hierarchical and human-readable. Automatic tools were developed to simplify large evolved trees, making it possible to understand, explain, and improve the evolved controllers.

Behavior Tree theory

BTs vs. FSM

Despite the multitude of possibilities that finite state machines offer, it has been noted that they encounter difficulties in satisfying two essential characteristics that are often important in autonomous agents: the ability to exhibit both reactive and modular capabilities.



For instance, the *obstacle* subtree used in the demonstration developed in this research, was utilized repeatedly in other parts of the tree. It also enables the robot to recalculate its route in case of encountering an obstacle.

03.

Methods and results

The example presented in this paper was designed and implemented using the "BehaviorTree.cpp" framework (Faconti, 2022), written in C++ 17. Additionally, the "Groot" editor (Faconti, 2019) was used to graphically create the behavior trees.

Methods and results

The demonstration presented in this section aims to provide a high-level overview of the **logic** utilized by an **autonomous robot within a swarm robotics environment**.

The focus is solely on the development of **software capabilities**, so hardware constraints such are outside the scope of this paper, as due to scalability issues.

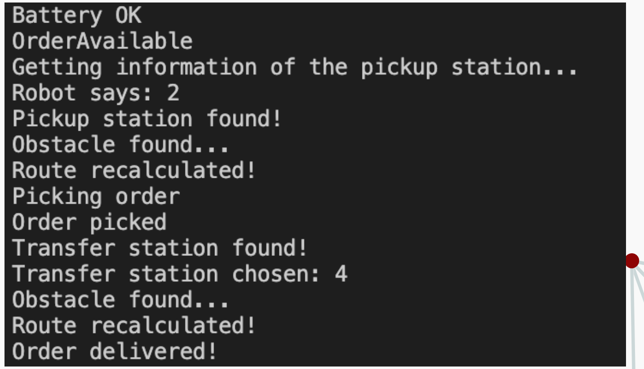
This research presents an experiment of a **warehouse setting** where robots on one side **collaborate** through transfer stations with their counterparts on the other side to achieve efficient package deliveries. The environment disseminates directives to all robots on one side with order requirements and selected pickup station information, and each robot assesses the availability of orders and communicates with other robots for collaborative task execution.

Methods and results

It is worth noting that the compatibility of Groot 1.0 is exclusively with BehaviorTree.CPP version 3.8.x

The **blackboard** mechanism is utilized to show the robot's capability of acquiring orders from various pickup points, the signals received and transmitted by the robot are currently hardcoded. However, in a real-world implementation, it would be imperative to incorporate the updating of the blackboard with data obtained from the sensors and the overall system in order to enhance the functionality of the system.

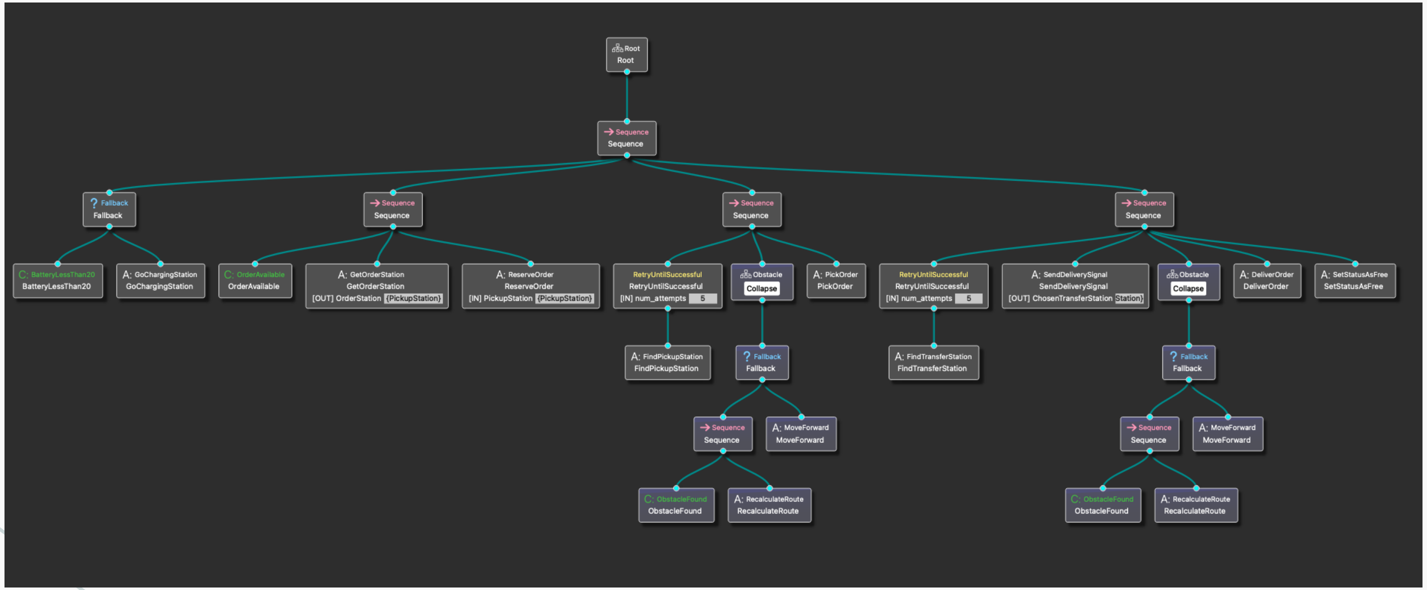
On the right the example of the output result of a successful mission for one autonomous robot.



```
Battery OK
OrderAvailable
Getting information of the pickup station...
Robot says: 2
Pickup station found!
Obstacle found...
Route recalculated!
Picking order
Order picked
Transfer station found!
Transfer station chosen: 4
Obstacle found...
Route recalculated!
Order delivered!
```

Methodology and results

Graphical representation of the behavior tree of the demo in Groot



04.

Conclusions & Future steps

The research question was **"To what extent does the use of Behavior Trees improve the coordination and control of swarm robotics systems?"** and the conclusion is that behavior trees can serve as a powerful tool in controlling the decision-making of a swarm of robots, enabling them to coordinate and accomplish a given task with efficiency. Furthermore, the use of behavior trees in swarm robotics offers several advantages, such as the ability to clearly represent complex logic and decision-making processes, and the capability to adapt to changing conditions in the environment. The ideal next steps in the development of the method proposed would be to test it in a real-world environment in order to better evaluate its sensitivity to the reality gap.

THANKS

More info in:
zaida.britotriana@studenti.unitn.it
https://github.com/zaidabri/BehaviorTreeSwarmRobotics