

ROBOTS OF THE FUTURE: Co-operating Swarms Instead of Individual Machines

A MULTI-TIER REFERENCE MODEL FOR ROBOT SWARM CONTROL SOFTWARE

AUTHOR: Risto Pitkänen, D.Sc. (Tech.), Principal Consultant in Atostek Ltd., risto.pitkanen@atostek.com

The most important and exciting future robots will co-operate in swarms instead of tinkering in isolation. Controlling such swarms requires multiple tiers of software of which only the lowest are deployed on board the machines themselves. From the point of view of business-critical efficiency, the topmost tiers are the most important to get right, because they have the most complete information of the mission and status of the swarm.

Introduction: Robot Swarms

In published media, discussion about the future of robots¹ and robotisation mostly revolves around individual autonomous moving machines that can perform tasks previously only humans were capable of. In this discourse, technical challenges to be solved are things such as computer vision, navigation, manipulation of objects, and other abilities an individual robot needs to have to be useful (Figure 1).

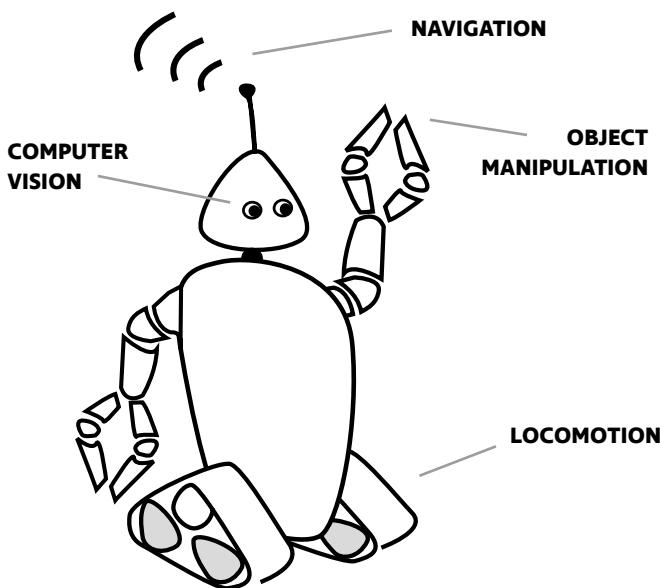


FIG1. INDIVIDUAL ROBOT CAPABILITIES.

However, many of the most exciting applications with the highest commercial potential actually require a swarm or fleet of co-operating robots (Figure 2). Examples include automated warehouse management, robotized shipping container terminals, robotized

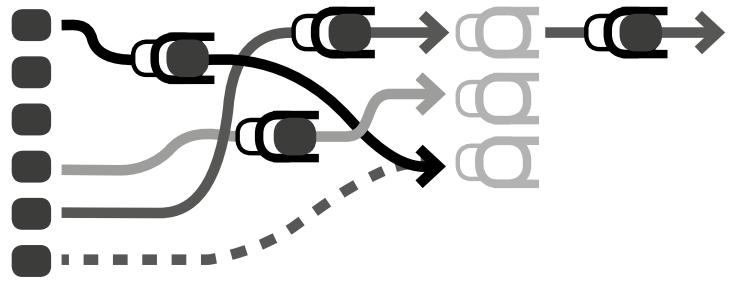


FIG2. CO-OPERATING SWARM OF ROBOTS.

public transport, and small packet delivery drone swarms. All these examples are already in use somewhere in the world, at least in a limited scale. In the future we'll almost certainly see many new applications.

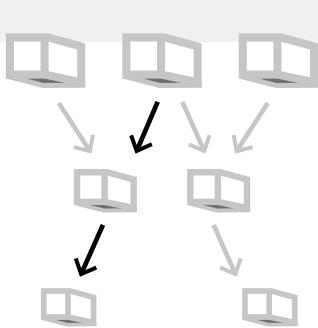
Controlling a swarm of robots to perform complex collaborative operations brings forth a new set of technical challenges to be solved. Joint operation must be planned, sub-operations must be scheduled, and the routes and actions of individual robots must be coordinated so that they do not interfere with each other. Failure to solve these challenges properly may lead to inefficiency, need of human intervention, material damage, or even loss of life (e.g. if robotised vehicles interfere with each other in a destructive manner).

Multi-Tier Model for Robot Swarm Control

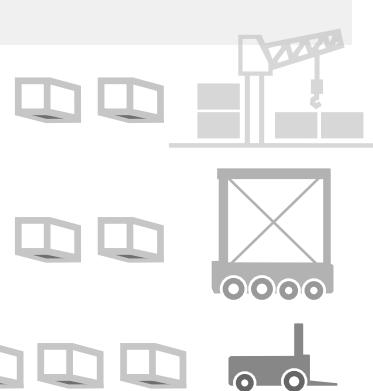
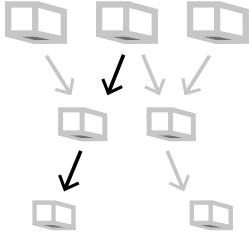
Multiple tiers can be recognized in a swarm control system: Task Planning, Task Scheduling, Routing and Onboard (Fig. 3). More tiers could be added towards business-level functions or coordination between multiple swarms, but we'll limit our scope to these four tiers. Depending on the application, Task Planning might not be needed, if the tasks correspond directly to orders or similar inputs received from outside of the control system (e.g. in the case of a robot taxi service, a task corresponds to a single ride request).

¹Note that traditional industrial robots used in manufacturing are left outside the scope of this article. Here the word "robot" refers to an (at least partially) autonomous moving machine, like, for example, a Roomba or R2D2.

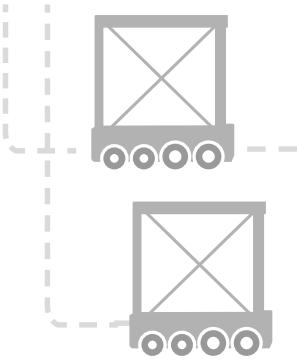
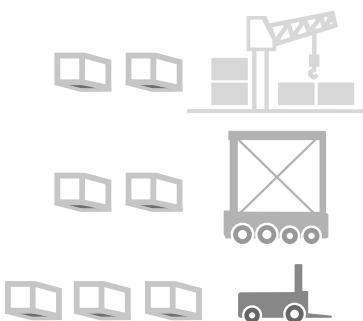
1. Task planning



2. Task scheduling



3. Routing



4. Onboard

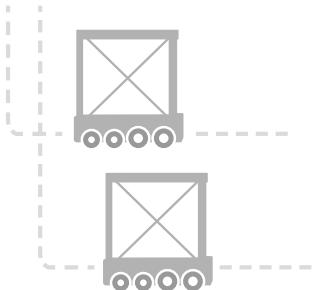


FIG. 3. MULTI-TIER REFERENCE MODEL FOR ROBOT SWARM CONTROL.

For example in a container ship terminal, the **Task Planning** software plans the unloading of a container ship carrying thousands of cargo containers. It produces a partially-ordered set of tasks, each identifying a container and its destination in the terminal storage area. The partial order reflects the physical limitations of the unloading and operation; for example, the topmost containers must be unloaded first, and the balance of the ship cannot be compromised by e.g. unloading the whole starboard side before proceeding to the port side.

Continuing with the same example, the **Task Scheduling** software receives the plan computed by the Task Planning software, and assigns the individual

tasks (container id, destination) to specific container-handling robots (ship-to-shore cranes, straddle carriers, stacking cranes, etc.) at the right time, considering the current locations and job queues of each machine.

The **Routing** software then computes an optimized routing plan for each of the container-handling robots, taking into account the current task assigned to each robot, the locations and routes of all other robots, and safety.

Finally, the **Onboard** software of each robot implements the autonomous operation of each machine that enables it to perform the task currently assigned to it. Its functions include reading sensors, controlling actuators, positioning and navigation, safety, etc.

Implementation Multi-Tier Robot Swarm Control

While a peer-to-peer architecture for swarm control is possible, it is usually better to deploy Routing and higher levels on a separate computing node or network (Fig. 4). Routing, scheduling and planning are computation-intensive applications that require advanced algorithms and a lot of processing power. They are also coupled with business information systems such as ERP. Task Planning may even be part of ERP functionality.

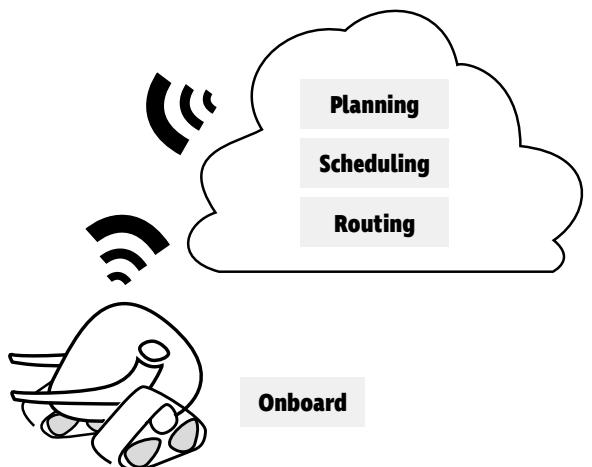


FIG. 4. DEPLOYMENT OF SWARM SOFTWARE COMPONENTS.

From the viewpoint of efficiency and performance, the top tier planning and routing components are very important. However polished and perfect the control algorithms of an individual, autonomously operating robot are, they cannot really optimize the performance of a whole swarm. Upper tiers have a more complete picture of the whole operation and can therefore perform much more significant optimizations. This can lead to dramatic business benefits.

Design and implementation of Onboard, Routing, Scheduling and Planning software require specialist understanding of algorithms as well as domain engineering knowledge. Modern software techniques and tools such as functional programming languages, computer-aided verification, and automated testing are necessary to achieve efficient and reliable results. ■