

TEK9010 - Exam prep summary

Sebastian G. Winther-Larsen

December 1, 2020

9: Swarm Robotics 1

What is swarm robotics?

Quick answer: swarm intelligence applied to robotics.

There is no explicit definition of a *swarm* in literature. A swarm is defined via its behavior.

The *size of a swarm* is defined by what it is not: “not as large as to be dealt with statistical averages” and “not as small as to be dealt with as a few-body problem”. The size of a swarm N is

$$10^2 < N \ll 10^3,$$

not Avagadro-large.

Swarm robotics is “the study of how a large number of relatively simple physically embodied agents can be designed such that a desired collective behavior emerges from local interactions among agents and between agents and the environment”, according to Dorigo and Sahin. But! A swarm is not necessarily

There are some key features. The fact that local interactions between agents and the environment should be possible requires robots to have local sensing and probably also communication capabilities. In fact, (local) communication is considered a key feature of swarms.

Collaboration is required to go beyond a mere paralllisation in swarm a swarm system. We want to go beyond the performance of simple parallelisation. Think of some clearing task with each robot cleaning a small assigned area.

Swarm performance.

Some keywords are *contention* or *inference* and (lack of) *coherency*, given by parameters α and β , repectively. The robots need to share limited resources and communicate. This is difficult.

In the context of swarm robotics we can interpret contention as interference between robots due to shared resources, such as an entrance to a base station or generally space. Collision avoidance is a waiting loop because the shared resource *space* is currently not available. This can be compared to an airplane flying in a holding pattern because the resource “runway” is currently in use and should certainly not be shared. Incoherency, in turn, can be interpreted as inconsistencies or overhead due to limited communication of information or due to imperfect synchrony.

The universal scalability law is important,

$$R(N) = \frac{N}{1 + \alpha(N - 1) + \beta N(N - 1)}.$$

Its inventor, Gunther, identifies four qualitatively different situations,

1. If contention and lack of coherency are negligible, then we get “equal bang for the buck” and have a linear speedup ($\alpha = 0, \beta = 0$),
2. If there is a cost for sharing resources in the form of contention, then we have a sublinear speedup ($\alpha > 0, \beta = 0$),
3. If there is an increased negative influence due to contention, then the speedup clearly levels off ($\alpha \gg 0, \beta = 0$).
4. If in addition there is also an increased influence of incoherence, then there exists a peak speed up and for bigger system sizes the speedup decreases ($\alpha \gg 0, \beta > 0$).

One can identify some “regions” of performance; super-linear, sub-linear, optimal and inference. As more agents are added, performance starts to increase (sub-/super-)linearly, then we get to an optimum after a while. After that comes the inference region.

In parallel computing, superlinear speedups can occur due to some interplay between problem size per computing unit and available memory. For example, if the problem can be divided into pieces that fit completely into a CPU’s cache, then one can observe a considerable speedup. In swarm robotics, superlinear performance increases occur due to qualitatively different collaboration modes that are accessible with increasing swarm size as seen in bucket brigades.

It is possible for a system-wide deadlock to occur in a swarm robotics system. For instance with a very high swarm density, such that all robots permanently try to avoid collisions resulting in zero performance.

Modelling swarms (as a series of mappings).

A swarm system of size N in 2D space can be described by the state vector,

$$\gamma = (r_1, r_2, \dots, r_N, v_1, v_2, \dots, v_N, s_1, s_2, \dots, s_N),$$

where r_i are the positions, v_i are the velocities and s_i are the discrete states for agents $i \in [1, N]$. We denote the configuration space by Γ , i.e. $\gamma \in \Gamma$, with $\dim \Gamma = 2N + 2N + N = 5N$.

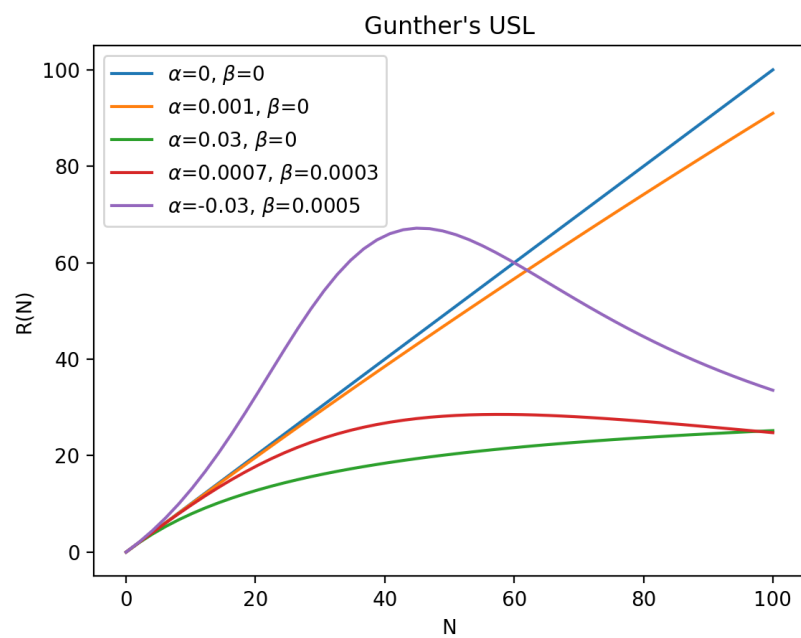


Figure 1: Gunther's Universal law of Computational Scalability

This is a large space to keep track off, and we can only observe one sequence, $\gamma_t, \gamma_{t+1}, \gamma_{t+2}, \dots$ at a time for a specific initial state γ_0 . What we ideally need instead is to understand how the system operates in general for any setup. We need to omit certain parts, s.t. we obtain a different definition of a configuration, $\phi \in \Phi$ with $\dim \Phi \ll \dim \Gamma$. We seek a mapping,

$$f : \Gamma \mapsto \Phi.$$

In a discrete time model, we also need two update rules, $g : \Gamma \mapsto \Gamma$ and $h : \Phi \mapsto \Phi$. We now have the following functionality,

$$f(\gamma_t) = \phi_t, \quad g(\gamma_t) = \gamma_{t+1}, \quad h(\phi_t) = \phi_{t+1}.$$

The following requirement should hold (we want this),

$$h(f(\gamma_t)) = f(g(\gamma_t)),$$

that is, the modelling abstractions implemented by f should be chosen carefull such that the model update rule h is able to predict the correct model configuration for the next time step.

We would also like an inverse map $f^{-1} : \Phi \mapsto \Gamma$ that reverses the model abstractions and rebuilds the configuration γ of the real system from the model configuration ϕ . However, typically f^{-1} cannot usefully be defined, because f is surjective, that is, we can have $\gamma_1, \gamma_2 \in \Gamma$ with $\gamma_1 \neq \gamma_2$ and $f(\gamma_1) = f(\gamma_2)$ due to the reduction in dimensionality caused by f .

Why do we need models in Swarm Robotics? Quote from Schweitzer:

To gain insight into the interplay between microscopic interactions and macroscopic features, it is important to find a level of description that, on the one hand, considers specific features of the system and is suitable for reflecting the origination of new qualities, but, on the other hand, is not flooded with microscopic details. (...) A commonly accepted theory of agent systems that also allows analytical investigations is, however, still pending because of the diversity of the various models invented for particular applications.

An extreme example of an extreme model abstraction,

$$f(\gamma) = \frac{|\{s_i | s_i = A\}|}{N} = \phi,$$

where $\dim \Phi = 1$. How does the update rule h look like? Non-trivial!

When are rate equations appropriate?

The Langevin equation.

The Focker-Planck equation.