Abstract

Distributed computing extends its scope to address problems relevant to mobile computing where hosts are physically mobile. Since a robot can be seen as a mobile computer, it is natural to consider a group of autonomous mobile robots as a kind of mobile distributed system. However, there are two fundamental differences with conventional distributed systems. The first is that robots usually require knowledge about their physical positions, and the second is that robots must control their own motion.

Many interesting applications of mobile robotics envision groups or swarms of robots cooperating toward a common goal. Consider a distributed system composed of cooperative autonomous mobile robots cultivating a garden. This application requires that robots move in all directions sharing the same geographical space. We consider a category of robotic applications where mobile robots have limited energy resources and wide geographical distribution. There is no centralized control nor global synchronization.

It is very important to focus on the problem of preventing collisions between mobile robots. Collision prevention leads to a dependable system and prevents the occurrence of serious damages to the robots which causes failures in the system.

In order to achieve a fail-safe motion, robots need to coordinate their movement. Cooperation is however difficult to obtain under the weak communication guarantees offered by wireless networks, because retransmission of messages is needed to ensure messages delivery in wireless environments. The communication delays to deliver messages are difficult to anticipate. Therefore, a time-free collision prevention protocol is very important in wireless environments.

The main contribution of this dissertation is providing a motion coordination platform that makes a system of mobile robots fail-safe independently of timeliness properties of the system. Mobile robots rely on this platform for their motion planning. The mobility coordination platform consists of time-free collision prevention protocols for an asynchronous system of cooperative mobile robots. The platform guarantees that no collision between robots can occur. In this dissertation, we analyze the performance of the protocols. A performance analysis provides insights for a proper dimensioning of system's parameters in order to maximize the average effective speed of robots. We consider also the collision prevention in presence of robots failures by crash, and provide fault-tolerant collision prevention protocols that tolerate the crash of a certain number of robots. We consider two system models, closed group and dynamic group models.

The first contribution is to provide collision prevention protocols for asynchronous cooperative mobile robots in a dynamic group model. In this model, the composition of the

system of which robots have only a partial knowledge, can change dynamically. Robots have limited communication range, hence they naturally form an ad hoc network on which they rely for their communication. The collision prevention protocol relies on a Neighborhood Discovery primitive which is readily available through most of wireless communication devices. The collision prevention protocol is based on a locality-preserving distributed path reservation system that takes advantage of the inherent locality of the problem, in order to reduce communication.

The second contribution of this dissertation is to provide collision prevention protocols for asynchronous cooperative mobile robots in a closed group model, in which a robot knows the composition of the group and can always communicate with all robots of the group.

The third contribution is providing group membership and view synchrony protocols among robot teams, in a distributed system model composed of a group of teams of worker robots that rely on physical robot messengers for the communication between the teams. The protocols tolerate the crash of a certain number of messengers robots and teams. Unlike traditional distributed systems, there is a finite amount of messengers in the system, and thus a team can send messages to other teams only when some messenger robot is available locally.