Collision avoidance for robot swarms focused on maneuvering and data collection rate from a RTOS approach

Yu Shan Hsieh, *Engineer, Hewlett Packard,* Ignacio Garita, *Engineer, Hewlett Packard,*
Horacio Hidalgo, *Engineer, Hewlett Packard*

***Abstract*—The ability to convey critical information promptly and efficiently between robots swarms is critical in collision avoidance scenarios. Different approaches to data collection are important considerations that must be taken in Real Time Systems where different units must effectively coordinate spatial movement to prevent collisions. Through the use of the computer networking concepts usch as the multicast protocol, our team has devised a new method for data collection in order to meet the timing constraints of a dynamic system such as this. The multicast concept allows for information management and establishment of policies related to a set of subscribers to a particular multicast group, which receives messages from a spatial element who manages and coordinates information and policies between the subscribed elements. The dynamics of different multicast groups (of which a single subscriber can be a part of many) creates localized information hubs and information that need not be distributed across entire networks. The resulting hubs are flexible, sometimes redundant, localized environments that have better adaptability to the immediate surroundings. Future studies and application of this networking concept can provide legitimization of the algorithms presented here; for example, use of robots swarms in an environment which is not homogeneous and where subscribers will obtain only relevant information to a specific area of this environment.**

***Index Terms*—MCAST, collision avoidance, RTOS, data collection, manuevering.**

I. Introduction

ADvances in technology have allowed automatons to gradually replace humans in many tasks involving control. Automated spacecrafts and UAVs (Unmanned Aerial Vehicle) are being heavily deployed in todays military operations and the scientific community uses unmanned spacecrafts in space explorations. Although these automatons are usually expensive and sophisticated, more economic solutions are now being used in civilian applications [YS1].

Well established engineering technologies have been made for these mobile UAVs to move around as a single unit and most of the time you will see a military drone fly alone to strike its target and then escape without being traced. Or a single spacecraft wandering over the surface of Mars and other planets. And rarely if not never will you see more than one Google vehicle cruising through the streets and roads to map every block of the city. Will the era of large fleet of planes like those of World War I and II one day return? Back in those times, large formations of planes consisting of many light fighters would often accompany a few large bomber in its bombing mission. Should a hostile plane come into the area to take the bomber down, the fighters will break formation, attack the intruder and return to guard the bomber. Such behaviour can be described as a swarm behaviour as it is similar to the movement of a swarm of bees or a flock of birds. Popular culture[YS2] in television and science fiction forsee the return of these swarm-like formations, often depicted in the form warplanes and spacecrafts using it's advantage in numbers to spam and overwhelm the opponent. Returning to the present real world, imagine a large fleet of unmanned spacecraft moving in a formations across Mars; if one or few vehicles stumbles upon a large obstacle such as a lake or a crater, it can quickly inform the other spacecrafts behind him to detour and find another route. And if one of the spacecrafts breaks down or gets stuck in a mud-like terrain, other spacecrafts close to it can assist him in getting out. Dispersion and redundancy are it's main strengths.

If the "robots" in this case scenario were to be exemplified as computers each with an operating system inside, it would require an RTOS (Real Time Operating System) to manage all its tasks as it exhibits all the characteristics that calls for an real-time solution. According to Stankovic [YS3], that system must be:

- Fast.
- Predictable.
- Reliable.
- Adaptive.

For this paper, we will delimit the use of real time system in the context of the robot's manuevering and data collection system to allow the robot to move freely without human intervention while preventing collision with obstacles including other units of the same type. It will focus solely on conceptual aspects and ignore all implementation considerations.

Additionally, there are some assumptions that are taken:

1) The spatial location is determined using 3-dimentional coordinates.
2) The units have a defined radius of communication.
3) All tasks are dynamic and aperiodic.
4) The arrival of the tasks are sporadic in nature.

In this paper, we will start off by discussing various concepts pertaining RTOS and how it applies to our model. We will also describe the use of Multicast protocol to improve the performance of the real time solution. The first part of this paper covers various concepts and definitions related to this subject. The second part describes the test we did to simulate such model. Finally, we will summarize our findings in the conclusions and discuss the advantages and disadvantages of such implementation.

This paper makes the following contributions:

1) it presents a problem that requires real-time solution.
2) it presents a novel solution to reduce the amount of information to handle.
3) argues the importance of reducing information to process to meet the timing constraints.
4) it presents a simple experiment to prove these ideas.

II. Research

A. RTOS

A control unit responsible for the movement of any unmanned vehicle is constantly handling inputs from sensors to get information like the current speed, acceleration, the direction it is heading. It must be able to detect near term obstacles and perform various tasks like computing the shortest path, the amount of energy it needs to inject to keep the current velocity, etc. As such, it requiere a time sharing notion from a operating system's standpoint. Apart from being time sharing, the tasks have different priorities among them. There are critical tasks where the very existance of the robot is threatened should it miss its deadline and then there are tasks that can be done in its idle time. These characteristics makes it a suitable candidate for using a real-time multitasking operating system.

Real time operating systems can be divided into four clases depending if they are Static or Dynamic, periodic or aperiodic. In this case, we can assume in the worst case that the task is dynamic and aperiodic meaning that the tasks are constructed in runtime and the arrival times of the tasks are not constant, in fact, they can be sporadic. An example of a sporadic task would be the obstacles encountered by the vehicle along it's path.

B. Collision Avoidance

C. Data Collection

D. Manuevering

III. Experiment

A. Experiment Introduction

1) What problem did we set out to resolve?:

2) What precisely was your contribution?:

3) Why should the reader care?:

4) What Larger question does this address?:

B. Experiment Body

1) What knowledge have you contributed that the reader can use elsewhere?:

2) What previous work do you build on?:

3) What is your new result?:

C. Experiment Conclusions

1) Why should the reader believe your result?:

IV. Conclusions

From the research collected in section I and the practical experiment demostrated in section II we can draw up the following conclusions:

1) The Mcast protocol can be used to reduce the amount of information being conveyed by the robots. Data collection rates are minimized by the subscription to an Mcast groups to obtain information of interest.
2) A real-time solution is needed to tackle the many critical aspects of the design. One of such aspects is the collision avoidance mechanism. Even if this is not the only activity a robot of this class can perform, the need to perform this task is, by itself, reason enough to use a real-time solution.
3) The nature of the arriving tasks are sporadic and unpredictable. The arrival of tasks depends of a number variables including
   - The distance and trayectory it is heading.
   - The speed with which it is moving.
   - The number of robots in it's vicinity.
   - The number and size of the obstacles.
4) The more channels of communication a robot ha access to, the more information it has to take better decisions but at the same time, the more information it has to process, risking missing the deadline for some tasks.
5) There is potencial demand in the future for such systems. As robot swarm technologies continues to evolve, the availability of these systems will increase and we will see more and more applications being made for it.
6) There is huge area for improvement in swarm robotics real-time paradigms. If we compare it to the personal computer industry, this is at it's infant stage.
7) Aspects of classic RTOS concepts and scheduling mechanisms can be applied. Future projects may include implementing a hybrid RM (Rate Monotonic) and a EDF (Earliest Deadline First) scheduler inside the kernel and test under what circumstances one is prefered over the other.

While the swarm robotics industry has yet to take off, it poses many challenges in making these systems robust and reliable. A relatively simple and practical experiment using simulated models can prove this point.

For the RTOS community, this presents many rooms for improvements on the current scheduling paradigms and opens a door for new and exciting problems to solve.

V. Acknowledgment

References

[1] *Swarm Robotics* 27 Feb 2013, 01:47 UTC. In Wikipedia: The Free Encyclopedia. Wikimedia Foundation Inc. Encyclopedia on-line. Available from *http://en.wikipedia.org/wiki/Swarm_robotics*. Internet. Retrieved 3 March 2013.