Fixed-Priority Scheduling of Variable Rate Tasks for an Autonomous Mobile Robot*

Ala′ Qadi Steve Goddard
Computer Science & Engineering
University of Nebraska–Lincoln
Lincoln, NE 68588-0115
{*aqadi, goddard*}*@cse.unl.edu*

Abstract

We present an autonomous, mobile, robotics application that requires dynamic adjustments of task execution rates to meet the demands of an unpredictable environment. The Robotic Safety Marker (RSM) system consists of one lead robot, the foreman, and a group of guided robots, called robotic safety markers (a.k.a., barrels). The application requires adjusting task periods on the foreman to achieve desired performance metrics with respect to the speed at which a system activity is completed, the accuracy of RSM placement, or the number of RSMs controlled by the foreman. A static priority scheduling solution is proposed that takes into consideration the strict deadline requirements of some of the tasks and their dynamic periods. Finally, a schedulability analysis is developed that can be executed online to accommodate the dynamic performance requirements and to distinguish between safe operating points and potentially unsafe operating points.

1. Introduction

The Robotic Safety Marker (RSM) system [2, 8, 7], poses a new and interesting real-time scheduling problem. The RSM system is a mobile, autonomous, robotic, real-time system that automates the placement of highway safety markers in hazardous areas, thereby eliminating risk to human workers. The RSMs operate in mobile groups that consist of a single lead robot—called the foreman—and worker robots—called RSMs—that carry a highway safety marker, commonly called a barrel.

Control of the RSM group is hierarchical and broken into two levels—global and local control—to reduce the per-robot cost. The foreman robot performs global control. To move the robots, the foreman locates each RSM, plans its path, communicates destinations points (global waypoints), and monitors performance. Local control is distributed to individual RSMs, which do not have knowledge of other robots and only perform local tasks.

In this work we consider finding a static priority scheduling solution for the tasks running on the foreman. Some of tasks running on the foreman have variable rates. In fact, the execution rate of many of the tasks are directly related to system performance criteria. The dynamic parameters of the system and desired performance, however, can lead to overload conditions. That is, the system is not schedulable unless the performance of one of the system activities is reduced. Thus, an online schedulability test is presented that can be used to distinguish between safe operating points and potentially unsafe operating points.

*Supported, in part, by grants from the National Science Foundation (EHS-0208619, CNS-0409382, and CCF-0429149).

2. Foreman Path Planning and Speed Control

The foreman depends on sonar sensors to plan its path by processing sonar signals to determine the presence of obstacles and their distance. The maximum speed at which the foreman can travel is related to the rate the sonar signals can be gathered and processed. If the foreman moves faster than the sonar signals can be processed, then the motion will be unsafe because there might be an obstacle in the path that will be undetected at that rate.

2.1 The Motion Control Task Set

The sonar unit consists of a ring of 24 active sonar sensors, with 15° separation. The 24 sonar sensor signals are pinged in sequence with a delay of $2ms$ between consecutive sensors to eliminate crosstalk. The motion control for the foreman code can be modeled as a set of periodic tasks: 24 tasks for sending sonar signals (one for each sensor), similarly another 24 tasks for receiving sonar signals and a path-plan/speed-control task. These tasks all execute with a common period p_s, which is called the scan period. Each sonar send task sends a command to its corresponding sonar sensor to transmit its signal. Each sonar receive task reads the corresponding sonar sensor after the signal is echoed back to the sensor. The parameters for the motion control task set are shown in Table 1, where e, p, d, ϕ and *max* J are the execution time, period, relative deadline, phase, and maximum jitter respectively. The phase represents the earliest possible release time for a task and maximum jitter is the maximum delay between the phase and the actual release time of the task. In this task set, jitter is caused by delays in receiving sonar signals, which are primarily dependent on the location of objects in the environment.

The sonar send tasks are released with a delay between them to eliminate crosstalk. The phase of these tasks, ϕ_{send_i}, is given by Equation (1), where i is the task index, τ

Task	e	p	d	ϕ	*max J*
Sonar-Send$_i$	$e_{send} = .085ms$	p_s	e_{send}	ϕ_{send_i}	0
Sonar-Receive$_i$	$e_{recv} = .03ms$	p_s	$e_{send} + e_{recv}$+*max* Δt	ϕ_{recv_i}	*max* Δt
Path-Plan/Speed-Control	$e_{plan} = 1.32ms$	p_s	e_{plan}	ϕ_{plan}	0

Table 1. Motion control task set. Phase parameters ϕ_{send_i}, ϕ_{recv_i}, and ϕ_{plan} are defined by Equations (1), (2), and (5) respectively. The maximum jitter parameter *max* Δt is defined by Equation (4).

is the delay used to eliminate crosstalk between consecutive sonar send tasks and e_{send} is the execution time of a sonar send task. These tasks have zero jitter and are required to execute as soon as they are released; hence a relative deadline equal to its execution time.

$$\phi_{send_i} = (i-1)\cdot(\tau + e_{send}) \qquad 1 \le i \le 24 \qquad (1)$$

$$\phi_{recv_i} = (i-1)\cdot\tau + i\cdot e_{send} \qquad 1 \le i \le 24 \qquad (2)$$

$$\Delta t = \frac{2\cdot D_{obstacle}}{340m/s} \qquad (3)$$

$$\textit{max}\ \Delta t = \frac{2\cdot D}{340m/s} \qquad (4)$$

$$\phi_{plan} = p_s - e_{plan} \qquad (5)$$

A sonar receive task is not released until its corresponding sonar send task has been executed and the signal is reflected back, which is called an echo. Equation (2) gives the phase for any sonar receive task i. The jitter of a sonar receive task, however, is dependent on the time delay between the transmission of a sonar signal and the reception of its echo, denoted as Δt. If an object is $D_{obstacle}$ meters away, the echo time delay can be computed using Equation (3) where the speed of sound is assumed to be 340 meters/second.[1] ($D_{obstacle}$ is multiplied by 2 in Equation (3) because the signal has to travel $D_{obstacle}$ meters before it is reflected back). Since we do not know the distance to objects *a priori*, a minimum distance, D, at which an object must be detected for the path-plan/speed-control task to safely control the robot's motion is defined. The maximum echo time delay—and hence maximum jitter—is then computed using D in Equation (4). If an object is farther than D meters away, the path-plan/speed-control task does not need to know about it because it will not provide any additional useful data in this scan period. Thus, receipt of an echo after *max* Δt time units is ignored.

The path-plan/speed-control task computes the path of the foreman and controls its speed based on the data collected from the sonar signals. The design of the control system is based on the assumption that this task executes at the end of the scan period, but after all of the useful sonar signals have been received.

[1]The actual speed of sound varies slightly depending on environmental conditions.

2.2 Continuous Motion Planning

The goal of this task set is to achieve a continuous safe movement of the foreman at the maximum safe speed while still being able to meet all task deadlines. To achieve continuous movement, the path is divided into a number of segments. Each segment is delineated by a scan point that marks the beginning of a scan period. We must, however, allow enough time for all motion control processing to complete within the scan period (i.e., before arriving at the next scan point). Thus, the length of the scan period, p_s, is dependant on the traveling speed of the foreman and the desired minimum object detection distance D. To simplify control, it is desirable for the foreman to have a constant speed between any two scan points. Under these constraints and assumptions, Equation (6) defines a lower bound on p_s that is required to safely control the foreman's motion.

$$\begin{aligned} p_s &\ge \phi_{recv_{24}} + d_{recv_{24}} + e_{plan} \\ &= 23\cdot\tau + 24\cdot e_{send} + \textit{max}\ \Delta t + e_{recv} + e_{send} + e_{plan} \\ &= 23\cdot\tau + 25\cdot e_{send} + \frac{2\cdot D}{340m/s} + e_{recv} + e_{plan} \end{aligned} \qquad (6)$$

We now quantify the relationship between p_s, the foreman's speed, D, and objects in the environment. Let each scan point in the foreman's path be denoted S_i. At least p_s time units must elapse before the foreman leaves point S_i and arrives at point S_{i+1}. A Scanning Zone, or simply $Zone\ i$, is defined as the area we can travel safely in without the need for another sonar scan. Scanning $Zone\ i$ is the area between point S_i and point S_{i+1}. The foreman achieves continuous motion by scanning $Zone\ i+1$ while traveling through $Zone\ i$. Of course, this requires that the foreman scan $Zone\ 0$, the first zone, before starting its movement. Figure 1 shows the distribution of scan points in time and distance from the moment the system starts (*note: v_{max} is the foreman's maximum speed*).

Let v_{maxi} denote the *maximum safe speed* at which the foreman can move through $Zone\ i$ while guaranteeing a continuous, crash-less motion. Obstacles in the environment, p_s and τ, all constrain v_{maxi}.

Let M_{safe} represent the maximum distance the robot can move safely. In this case, M_{safe} is the minimum distance scanned by the sonar sensors: D. As we can see in the top part of Figure 1, at time $t = 0$ the foreman is initially at scan point S_0. We start our initial scan but do not start the motion until the end of the first scan period. At this time

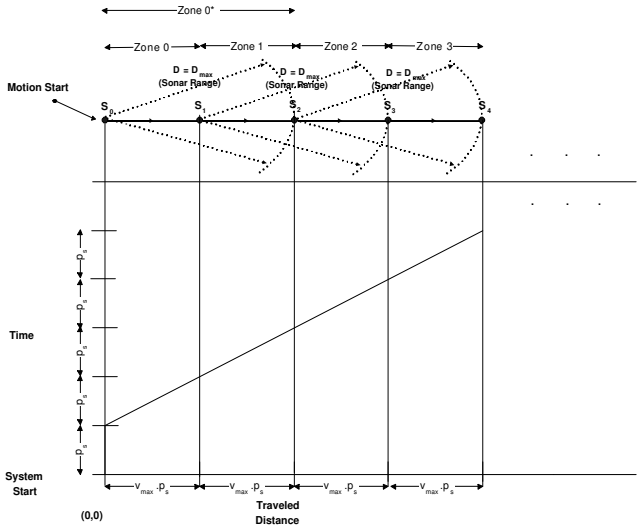

Figure 1. Scanning point distribution in time and space with no obstacles.

$M_{safe} = D$ because the foreman does not start the motion until the end of p_s time units. Therefore $Zone\ 0^*$ extends to a distance of D. We can keep moving safely in $Zone\ 0^*$, but this will imply the need to stop at point S_2 at the end of $Zone\ 0^*$ to scan $Zone\ 2$. We can avoid the stop if we divide $Zone\ 0^*$ into two smaller zones—$Zone\ 0$ and $Zone\ 1$—and scan $Zone\ 2$ while moving in $Zone\ 1$. With this modified division of zones v_{max} can be calculated from Equation (7).

$$v_{max} = \frac{M_{safe}}{\textit{Avilable Time to Complete the Motion}} = \frac{D - v_{max} \cdot p_s}{p_s} = \frac{D}{p_s} - v_{max} = \frac{D}{2 \cdot p_s} \tag{7}$$

If at any scan point S_i we change the sonar period p_s or change the sonar detection range D, then Equation (7) becomes

$$v_{max\,i+1} = \frac{D_i - v_i \cdot p_{s_i}}{p_{s\,i+1}} \tag{8}$$

3. RSM Motion Planning and Tracking

The RSM motion planning and tracking performed by the foreman can be modeled as a set of periodic tasks with attributes e, p, d, ϕ, and *max* J, as listed in Table 2. This RSM motion planning and tracking task set is nearly the same task set that was analyzed in [8]. However, a path prediction task has been added to identify and correct deviations from the planned path in the actual path taken by RSMs. (Full details of the path prediction algorithm are presented in [6], while [8] provides a description of the other tasks in this task set.)

4. Real Time Scheduling

In this section we analyze the schedulability of the system and derive an online schedulability test. An affirmative result from the schedulability test ensures that all relatively deadlines will be met. A negative result indicates a possible overload condition in which performance guarantees cannot be made.

From an application point of view, it is preferable that the motion control task set execute with strictly greater priority than the RSM motion planning and tracking task set. Combining this desire with the optimality of deadline monotonic scheduling [5] results in the task priority assignment shown in Table 3. The priority assignment is not strictly deadline monotonic since the range for p_s is $55.34ms \leq p_s \leq 3650ms$, while the range for p_l is $50ms \leq p_l \leq 1000ms$. Under most operating conditions, however, the chosen priority assignment is deadline monotonic.

Note that Tasks 1, 3, and 9 in Table 3 are not single tasks but actually groups of tasks with common characteristics. For brevity, we assign them a single task index and priority. This is reasonable as long as priority ties are assumed to be broken in favor of the task with the smaller index i subscript (and hence earlier phase for the send and receive tasks).

The task set has predefined static priorities with phases and deadlines less than or equal to periods. The task set also has two dynamic periods. The goal is to find an efficient schedulability test for the task set that can be executed online (because of the dynamic work load). Our approach is based on the principles of time demand analysis presented in [4, 1].

A schedulability test using the time demand analysis requires finding a solution to an iterative time demand equation for every task in the task set. This is inefficient for two reasons. First, it assumes worst-case alignment of periods for all tasks, which over states the response time for most of the tasks in this task set. Second, dynamic periods in the task set require this test to be done online and the computation time to find a solution for Equation (3) in [1] is not deterministic. Therefore we present a more efficient schedulability test for this task set based on time demand analysis principles and proprieties of the task set.

Theorem 4.1. *All Sonar Send tasks (Task 1) will always meet their deadlines if* $p_s \geq 23 \cdot \tau + 24 \cdot e_{send} + max\ \Delta t + e_{recv} + e_{send} + e_{plan}$.

Theorem 4.2. *The Path-Plan/Speed-Control task (Task 2) will always meet its deadline if* $p_s \geq 23 \cdot \tau + 24 \cdot e_{send} + max\ \Delta t + e_{recv} + e_{send} + e_{plan}$.

Theorem 4.3. *All Sonar Receive tasks (Task 3) will always meet their deadlines if* $p_s \geq 23 \cdot \tau + 24 \cdot e_{send} + max\ \Delta t + e_{recv} + e_{send} + e_{plan}$.

For the rest of the tasks, offline analysis is not enough to determine the schedulability of the tasks because some of the tasks have dynamic periods that are independent of p_s. Dynamic periods introduce complexity in determining the scheduling condition because of the need to do the time demand analysis online. Applying the time demand analysis method presented in [1] to Tasks 4 to 9 requires finding a solution to Equation (3) in [1] for each task iteratively. A careful analysis of the task set, however, reveals that even

Task	e	p	d	ϕ	*max J*
Scanning	$12ms$	p_l	p_l	0	0
Detecting	$.0172 \cdot n^2 + .1695 \cdot n + 12.69$	p_l	p_l	0	0
Predicting	$e_{predict} = 3.8 \cdot n$	p_l	p_l	0	0
Planning	$16ms$	$1500ms$	$1500ms$	0	0
Way Point$_i$	$8.33ms$	$1500ms$	$1500ms$	0	0
Window Resizing	$2ms$	p_l	p_l	0	0

Table 2. RSM motion planning and tracking task set. The variable n represents the number of RSMs being controlled by the foreman.

Task Index	Task	p	Priority
1	Sonar Send$_i$	p_s	1
2	Plan/Speed	p_s	2
3	Sonar Receive$_i$	p_s	3
4	Scanning	p_l	4
5	Detecting	p_l	5
6	Predicting	p_l	6
7	Window Resizing	p_l	6
8	Planning	1500	7
9	Way Point$_i$	1500	8

Table 3. Task priority assignments.

though several tasks have dynamic periods there are only three distinct periods in the task set at any one time. Theorem 4.4 states that schedulability can be established online with an affirmative result from two conditions.

Before we present the schedulability condition, we define the following notation:

- $DEM_{T_i,\ i=j,\ldots,k}(L)$: Demand (interference) by tasks j through k in any interval of length L.
- $\sum_{p_i=L} e_i$: Total execution time of tasks with period L.

Theorem 4.4. *All tasks will meet their deadlines if Equations* (9) *and* (10) *hold.*

$$\sum_{p_i=p_l} e_i + DEM_{T_i,\ i=1,\ldots,3}(p_l) \leq p_l \tag{9}$$

$$\sum_{p_i=1500} e_i + DEM_{T_i,\ i=1,\ldots,7}(1500) \leq 1500 \tag{10}$$

The demand created by higher priority can be calculated by application of Equations (2) and (11) in [1] by Audsley et al. Equation (11) is a generalization of Equation (2) and computes the demand of "sporadically repeating tasks" with inner and outer periods [1].

5. Conclusion and Future Work

We presented a mobile robotic application that requires adjusting sensor sampling rates to produce desired performance levels. A static priority scheduling solution is proposed that takes into consideration the strict deadline requirements of some of the tasks and their dynamic periods. We have shown how system parameters and environment changes can create overload conditions on the system processor and how system schedulability can be evaluated online.

The online schedulability test can be used to distinguish between safe operating points and potentially unsafe operating points. Moreover, the analysis and online schedulability test provides a framework for a future application-level control algorithm that can make dynamic performance/schedulability tradeoffs. Future work will also include generalizing the modeling and schedulability analysis presented here so that it can be applied more easily to tasks of other real time mobile autonomous systems.

References

[1] N. Audsley, A. Burns, M. Richardson, and K. T. A. Wellings. Applying new scheduling theory to static priority pre-emptive scheduling. *Software Engineering Journal*, 8(5):284–292, September 1993.

[2] S. Farritor and M. Rentschier. Robotic highaway saftey marker. In C. Mellish, editor, *ASME International Mechanical Engineering Congress and Exposition*, Montreal, May 2002.

[3] S. Goddard and K. Jeffay. Analyzing the real-time properties of a dataflow execution paradigm using a synthetic aperture radar application. In *Proceedings of the 3rd IEEE Real-Time Technology and Application Symposium*, pages 60–71, Montreal,CA, June 1997.

[4] J. Lehoczky, L. Sha, and Y. Ding. The rate monotonic scheduling algorithm - exact characterization and average case behaviour. In *Proceedings of 10th IEEE Real-Time Systems Symposium*, pages 166–171, December 1989.

[5] J.-T. Leung and J. Whitehead. On the complexity of fixed-priority scheduling of periodic real-time tasks. *Performance Evaluation*, 2(4):237250, December 1982.

[6] A. Qadi, S. Goddard, J. Huang, and S. Farritor. A performance and schedulability analysis of an autonomous mobile robot. Technical Report TR-UNL-CSE-2004-0015, Computer Science & Engineering, University of Nebraska-Lincoln, December 2004.

[7] X. Shen. Control of robotic highway saftey markers. Master's thesis, Mechanical Engineering, University Of Nebraska-Lincoln, 2003.

[8] J. Shi, S. Goddard, A. Lal, and S.Farritor. A real-time model for the robotic highway safety marker system. In *Proceedings of the 10th IEEE Real-Time and Embedded Technology and Application Symposium*, pages 331–440, Toronto, CA, May 2004.