Feedback-based Real-time Scheduling in Autonomous Vehicle Systems

Suzhen Lin and G. Manimaran
Dept. of Electrical and Computer Eng.
Iowa State University, Ames, IA 50011, USA
{*linsz,gmani*}*@iastate.edu*

B. L. Steward
Dept. of Agricultural and Biosystems Eng.
Iowa State University, Ames, IA 50011, USA
bsteward@iastate.edu

Abstract

The use of feedback control techniques has been gaining importance in the context of scheduling in real-time systems as a means to provide predictable performance in the face of uncertain workload. In this paper, we propose a novel feedback-based scheduling approach for task scheduling in real-time systems. We focus on systems with mobile nodes, where the mobility characteristics affect task parameters. The objective is to achieve low miss ratio and high CPU utilization. This objective is achieved by feeding back system performances and adapting nodes' mobility parameters. We study the new approach in a selective herbicide spraying problem in the agricultural production wherein the speed of an autonomous vehicle (and hence task parameters) is adapted based on weed distribution in the given farm field. Simulations and analysis show that our approach can achieve low miss ratio and high CPU utilization.

1 Introduction

In real-time systems, the correctness of the system depends not only on the logical correctness of the result, but also on the time at which the results are produced. Traditional real-time scheduling algorithms are based on estimations of the (pessimistic) worst-case execution time of tasks. However, such kinds of algorithms result in an extremely underutilized system. In many cases, it is preferable to monitor the performances of the systems and allocate resources such that critical timing requirements are met and high resource utilization is achieved at the same time. Such monitoring mechanism also can help us to deal with bounded change in load (e.g., execution time) dynamically.

One of the very successful areas in addressing performance in the presence of uncertainty is control theory. Feedback of measured quantities to correct the behavior of a system has been a powerful concept that has made technological advances in applications such as amplifiers and avionics. Through concerted use of feedback control, the concept has been used to deal with uncertainty inherent in most systems. The feedback strategy is useful primarily if uncertainty is present. Another attractive feature of control-oriented framework is that the behavior of the system need not be exactly modeled.

When the scheduler performs the task scheduling, it uses task parameters, such as the execution time, period and deadline. Among the parameters, the (actual) execution time of a task depends on conditional statements and data dependent loops that are influenced by the dynamics of the environment in which the system is operating. Thus, task execution time creates workload uncertainty in the system. Therefore, feedback control can be used to adjust the resource allocation and track the system performance.

In recent years, there has been much work in feedback-based scheduling of real-time systems. In [1][2], the authors present a feedback control EDF scheduling algorithm for real-time uniprocessor systems. In [3][4], authors present a closed-loop scheduling algorithm based on execution time estimation in multiprocessor systems. In [5][6], authors proposed a methodology for automatically adapting the rates of a periodic task set.

All these work consider that the node(s) are fixed, that is, nodes are not mobile. However, there are applications wherein the nodes are mobile. Examples include robots and autonomous vehicles in industrial and agricultural productions. To the best of our knowledge, there is no prior work on feedback-based adaptive scheduling of real-time tasks in autonomous vehicle systems.

The objective of our work is to develop and analyze feedback-based adaptive scheduling schemes for autonomous vehicle systems [7][8]. The proposed feedback-based scheduling scheme can be used in systems with mobile nodes. **When the nodes are mobile, nodes continuously move and execute certain tasks, so the mobility can affect the task parameters.** Thus for a particular application, we have to identify the relation between the mobility characteristics (e.g., speed) and the values of task parameters (e.g., execution time, deadlines and periods). The adaptation can be carried out on the mobility characteristics, which will further lead to the change of task parameters. Thus the schedule will be adjusted, that is, we can achieve a schedule which can satisfy the desired real-time requirements.

In this paper, we propose and study a novel scheduling scheme using feedback for the *selective herbicide spraying problem* in agricultural production [9]. A node is a vehicle which is equipped with sensing, processing, and actuating capabilities. Adaptation is focused on the speed of the vehicle, which will lead to changes of task parameters (deadlines and periods of tasks).

The rest of the paper is organized as follows. In Section 2, the selective herbicide spraying problem is described. In Section 3, we propose feedback-based solution for the autonomous vehicle problem. In Section 4, we discuss the simulation results. Finally, in Section 5, we make some concluding remarks.

2 Selective Herbicide Spraying Problems

In this paper, we address the selective herbicide spraying problem in agriculture production [9]. Vehicles are needed to detect weed conditions in the fields and then spray herbicide based on the weed conditions. In this system, images of field are obtained periodically by a camera mounted on the vehicle, and the images are processed to decide the weed density, then a command is sent to the sprayer. Weed detection and command transfer can be treated as a task. In weed area, the vehicle has to go slowly because the calculation of weed conditions needs longer time before the herbicide is applied. Thus, the weed distribution introduces load uncertainty in the system, and each task is associated with a deadline. We define deadline miss ratio as the ratio of number of tasks that miss their deadlines to the number of tasks submitted to the system.

There are three ways to deal with the uncertainty in execution time for weed processing: (1) The vehicle travels at the lowest speed, that is, the image data arrive at the system at a lowest rate (longest period). This will result in zero miss ratio, but a high cost (low CPU utilization and long time to finish processing the field). (2) The vehicle travels at the highest speed, that is, the image data arrive at the system at a highest rate (lowest period). This will result in a high miss ratio, but high CPU utilization and short processing time (low cost). Thus, the task execution time introduces a trade-off between miss ratio and cost. (3) **The vehicle travels at an adjusted speed which is obtained using feedback control algorithm.** Unlike approaches (1) and (2), approach (3) has the potential to capture this trade-off in order to minimize both miss ratio and cost.

2.1 The Vehicle System

Under current practices, most herbicide is applied uniformly in crop fields, however, weeds are not uniformly distributed and they are distributed in patches [10]. Currently, chemical application systems are being investigated that can sense weeds in real-time with forward vehicle-mounted image sensors and adjust the application rate at herbicides using adjustable nozzles on a rear-mounted spray boom. The sensor senses weed conditions in the field, and then passes the data to an on-board processor. The processor performs image processing to estimate the weed density for the sensed area of the field. Then the processor sends the commands to control the action of the nozzles. Notice that the vehicle must travel the distance between the sensor and the nozzles before activating the nozzles. Due to non-uniform weed distribution of the field, image processing often requires variable computation times for weed density estimation. The computation task must be completed before nozzles arrive at the corresponding weed patch locations. These real-time requirements must be met in this system.

The system under discussion, shown in Figure 1, has a camera installed at 3.35m height such that the camera has a field of view (FOV) $2.44m \times 3.05m$. The FOV is divided into regions called control zones, and nozzles are activated based on weed detections in individual control zones. Control zones are 0.61m long in the travel direction and 0.51m wide, corresponding to the spray pattern of the nozzle directly behind them. Thus, the usable image area is a 4×6 matrix of control zones measuring 2.44m by 3.05m wide that consists of the number of complete control zones that fit in the FOV [9].

Due to the the size of FOV and the control zones that the sprayers can spray at each activation, image processing proceeds row by row, starting with the row closest to the spray boom, until all the rows are processed. After each control zone row was processed, a 2-byte nozzle command is sent to a nozzle controller.

2.2 Task Model

In order to reasonably arrange the computation of row images and meet real-time requirement, we abstract the task model from this chemical application system. Data acquisition is done by a frame grabber, which accepts the video signal of a FOV from the camera and then store the data into the processor responsible for image processing. To model as a real-time scheduling problem, we treat the image pro-

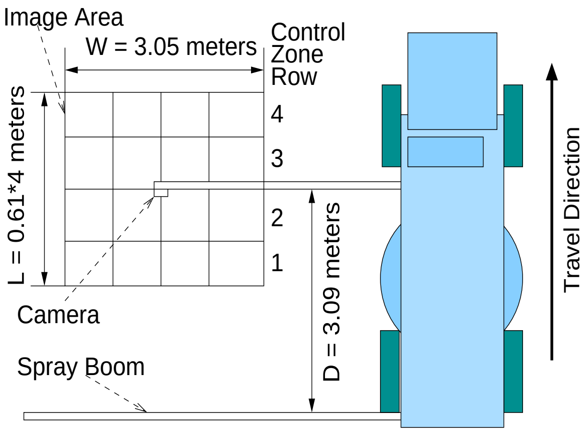


Figure 1. Physical configuration of the chemical application system modeled. The camera acquires an image of field surface and nozzles on spray boom are controlled based on image processing results

cessing and command transfer for each row as a computation task (T_i), shown in Figure 2. The command transfer will take a small part in the task's computation, and the image processing will take most of the task's computation time. Thus, the system has four periodic tasks. The relative deadline (d_i for task T_i) is equal to the time interval from the time that the camera gets the data to the time that the sprayer arrives at the row. From Figure 1, we can get the task model. In the system, the length of each control zone is $y = 0.61m$, the distance between the camera and the sprayer is $D = 3.09m$. Let the speed of the vehicle be s, the computation time of tasks be c_i, and the period of tasks be p_i, (i = 1, 2, 3, 4), a task T_i can be denoted by $\langle c_i, p_i, d_i \rangle$. The period can be calculated by Equation 1, and the relative deadlines of the four tasks can be calculated by Equation 2, 3, 4 and 5.

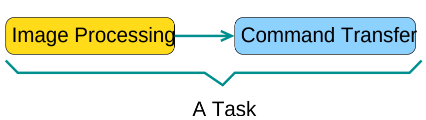


Figure 2. A task in the system

$$p_i = \frac{y \times 4}{s} \qquad \text{i = 1, 2, 3, 4} \tag{1}$$

$$d_1 = \frac{D - y \times 2}{s} \tag{2}$$

$$d_2 = \frac{D - y}{s} \tag{3}$$

$$d_3 = \frac{D}{s} \tag{4}$$

$$d_4 = \frac{D + y}{s} \tag{5}$$

If the computation time of each of the four tasks is $p_i/4$, then the tasks are schedulable on the uniprocessor by using EDF scheduling algorithm. However, when the computation time of each task is equal to $p_i/4 + \epsilon$, where ϵ is a very small positive value, all the tasks will not be schedulable after the application runs for a certain time. From Figure 3 we can find the reason.

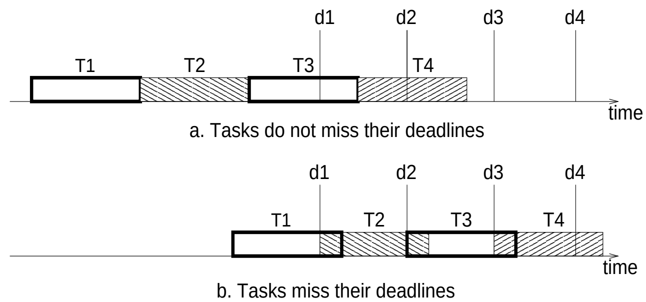


Figure 3. Unschedulable tasks

Figure 3a shows that at the beginning of the application, tasks will not miss their deadlines even $\sum_{i=1}^{i=4} \frac{c_i}{p_i} > 1$, this is because the tasks have different relative deadlines and some of them are greater than the period. However, as the time increases, tasks arrive in one period will be delayed longer time until the system reaches a situation as shown in Figure 3b. Figure 3b shows that task T_1 misses its deadline, which leads to the missing deadlines of the other three tasks. This is because the difference between relative deadlines is equal to $\frac{p_i}{4}$, which is shown in Equation 6.

$$d_4 - d_3 = d_3 - d_2 = d_2 - d_1 = \frac{y}{s} = \frac{p_i}{4} \tag{6}$$

In order to use feedback control technique to adjust the deadlines of tasks when the computation time of tasks exceed $\frac{p_i}{4}$, that is, when $\sum_{i=1}^{i=4} \frac{c_i}{p_i} > 1$, further prevent a 100% deadline miss ratio, we modify tasks' deadlines as follows:

$$d_1 = \min\{\frac{D - y \times 2}{s}, p_1\} \tag{7}$$

$$d_2 = \min\{\frac{D - y}{s}, p_2\} \tag{8}$$

$$d_3 = \min\{\frac{D}{s}, p_3\} \tag{9}$$

$$d_4 = \min\{\frac{D+y}{s}, p_4\} \quad (10)$$

The data of the system is shown in Table 1.

Control Zone Processing Time	$c = 0.156\ to\ 0.685(seconds)$
Speed	$s = 3.2\ to\ 14(km/h)$
Distance between Camera and Sprayer	$D = 3.09(meters)$
Length of a Control Zone	$y = 0.61(meters)$
Periods of Tasks	$p_i = 0.627\ to\ 2.742(seconds)$

Table 1. Data of the system

Task Model Modification

To show the necessity of task parameter modification, we carried out the following simulation. We fixed the speed of the vehicle at 14km/h, and set the computation times of tasks between $157.0ms$ and $157.5ms$, which is slightly larger than $\frac{p_i}{4}$. We simulated tasks with original deadlines and modified deadlines respectively. The miss ratios of these two studies are shown in Figure 4. From Figure 4, we noticed that if we use the original deadlines for tasks, the miss ratio will go up to 100% after some time. After modified the deadlines of tasks, the miss ratio stay at 25% from beginning to the end. Thus we can detect the computation exceeding at the beginning, and by using feedback control, we can adjust the speed of the vehicle and prevent 100% miss ratio.

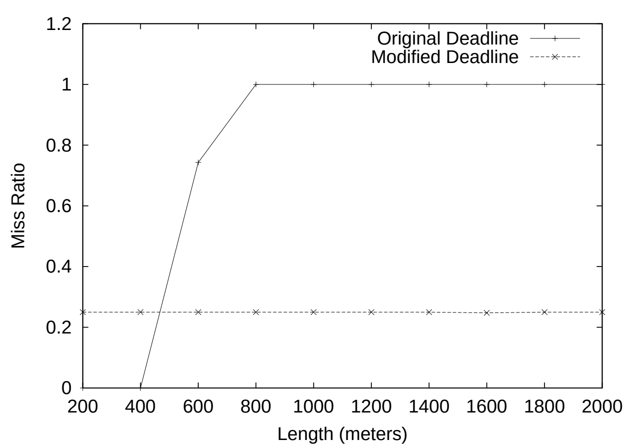


Figure 4. Miss ratio without feedback adaptation

3 Proposed Solution

In this section, we introduce the feedback control technique and propose a feedback approach to deal with the uncertainty in the scheduling in autonomous vehicle system.

3.1 Feedback Control - Background

Figure 5 shows a typical control system, consisting of a controller, a plant to be controlled (controlled system), sensors, and actuators [11]. The system defines four variables: (1) exogenous variables are inputs from outside of the system, e.g., set points (desired values of the output values) and disturbance. (2) regulated variables are the output values that the system regulates. (3) measured variables are values that the sensors measure. (4) control variables are the inputs to the actuators. The actuators will actuate the plant based on the control variables. Besides, the system also defines the error, which is the difference between the set points and the feedback information.

The system works as follows: The sensors periodically monitor the regulated variables and get the error to feed to the controller. The controller computes the required control, using the control function of the system, based on the error. The actuators change the control (manipulated) variables to control the system.

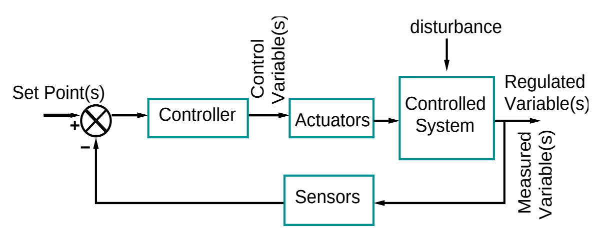


Figure 5. Control system architecture

3.2 Task Scheduling in the Vehicle System

In real-time systems, we use $\langle c_i, p_i, d_i \rangle$ to denote a periodic task T_i, c_i is the computation time of the task, p_i is the period of the task, and d_i is the deadline of the task. Usually, uncertainty exists in the image processing computation time. In the scheduling algorithms, we can measure and feedback the miss ratio and CPU utilization (regulated and measured variables), adjust the computation times, periods, or deadlines (control variables) to achieve good scheduling result. The goal is to achieve low miss ratio (MR) and high CPU utilization (U). The architecture is shown in Figure 6, and the control law is summarized in Equation 11, 12 and 13. K_{pm}, K_{pu}, K_{cm}, K_{cu}, K_{dm} and K_{du} are coefficients that map the measured errors to the change of the corresponding control variables. MR_s and U_s are set points for MR and U respectively. The subscript k and $k-1$ means the corresponding values at time instance k and $k-1$ respectively. The error of miss ratio is used to prevent high deadline miss ratio and the error of CPU utilization is used to prevent low CPU utilization. The main idea is to measure the difference between the measured value and the set point, and then use PI (Proportional and Integral) control law to calculate the new control variable. The integral part is used to achieve small error in the steady state, the proportional part is used to achieve quick response in transient state. Different normal control which use PID (D stands for Derivative), we do not use the derivative part because derivative control will exagerate disturbance in such systems.

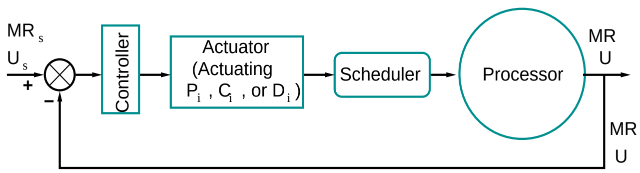


Figure 6. Local feedback control

$$p_{ik} = p_{i(k-1)} - K_{pm}(MR_s - MR_{k-1}) - K_{pu}(U_s - U_{k-1}) \tag{11}$$

$$c_{ik} = c_{i(k-1)} - K_{cm}(MR_s - MR_{k-1}) - K_{cu}(U_s - U_{k-1}) \tag{12}$$

$$d_{ik} = d_{i(k-1)} - K_{dm}(MR_s - MR_{k-1}) - K_{du}(U_s - U_{k-1}) \tag{13}$$

In this paper, we develop an algorithm that adjusts the speed of the vehicle to adjust tasks' periods and deadlines.

3.3 Controller Design

To use feedback control technique, two important parameters must be decided. One is the sampling period for the controller, and the other is the coefficient for the controller. Since we want to feedback the change in the system performances as quick as possible so that we can take immediate action to deal with the change, a short sampling period will be preferred. However, higher overhead will be incurred when shorter sampling period is used. According to the time used for an instruction's execution in RT-Linux, which is nearly 200 μs, assume that 20 lines of instructions are used to calculate the feedback information and conduct feedback functions, then time used for feedback calculation is about 4 ms. Compared with the minimum computation time of a task, which is 156 ms, this 4 ms is very small. Thus we can use whatever small sampling period we want. Since the camera gets data for four tasks at a time, we feedback the performance when the vehicle encounters every four tasks, that is, the sampling period is the time that the processor processes or discards four tasks, which include those meet deadlines and those miss deadlines.

Now, the remaining problem is to decide the value of the coefficients of the controller. The control algorithm we are going to use is shown in Equation 14.

$$speed_k = speed_{k-1} + K_m(0.0 - MR_{k-1}) + K_c(c_p - c_a) \tag{14}$$

$speed_{k-1}$ is the speed when the processor processes the four tasks, $speed_k$ is the speed the vehicle is going to run at, MR_{k-1} is the miss ratio of the four tasks, $c_p = \frac{1}{4}p_i$, c_a equals to the average computation time of the four processed tasks if no tasks miss deadlines. K_m and K_c are coefficients of the controller. The reason to adjust the speed of vehicle is that from Equation 6-10, the change of speed will lead to the change of tasks' deadlines and periods. When the computation times are long, tasks may miss deadlines, by slowing down the vehicle we can extend the deadlines of tasks, thus tasks can have enough time to finish the calculation before reaching the deadline. Also, we assume the kinematic model of the autonomous vehicle, that is, we assume that the speed can change immediately. The feedback of the computation difference is to measure the CPU utilization, since in the ideal schedule $c_i = \frac{p_i}{4}$ for task T_i and $\sum_{i=1}^{4} \frac{c_i}{p_i} = 1$. We consider two situations, one is when there are tasks missing deadlines, that is, the miss ratio is greater than zero, we need to slow down the vehicle to extend the deadlines of tasks, that is why we have $K_m(0.0 - MR_{k-1})$ in Equation 14. The other situation is that when there are no tasks missing deadlines, that is, the miss ratio is equal to zero. In this case, it is possible that the average computation time of the four tasks are far less than $c_p = \frac{1}{4}p_i$, which means the vehicle can run faster to process more tasks. This is why we have $K_c(c_p - c_a)$ in Equation 14. Obviously, in this system the measure of computation time is in fact a measure of the system utilization.

Since there are two situations to control, we add conditions for Equation 14. We measure the miss ratio and the average computation when the miss ratio is zero. If miss ratio is greater than zero, we let $c_p = c_a$ and then use Equation 14. If miss ratio is equal to zero and $c_p - c_a > 0.05 \times c_p$, we use Equation 14 too. Since these two situations happen at different time, we can consider K_m and K_c separately. We know that to use feedback control theory efficiently, the controller should not lead the speed to exceed the maximum speed or less than the minimum speed of the vehicle. Since the maximum change of speed is from 0.89m/s to 3.89m/s, and the maximum miss ratio might be 1, thus we have $0.89 \geq 3.89 + K_m(0.0 - 1.0)$ and we get $K_m \leq 3.0$. Therefore, in the simulation, we will change K_m in the range (0.0,3.0]. Similarly, if the speed of the vehicle changes from 0.89 m/s to 3.89 m/s, then the computation time changes from 0.156 s to 0.685 s, thus we have $0.89 - 3.89 \leq K_c(0.156 - 0.685)$, that is, $K_c \leq 5.67$. Therefore, in the simulation, we will change K_c in (0.0, 5.67]. Here, what we need to mention is that the ranges for K_m and K_c are not sufficient, however, we can adjust these two values in the ranges to observe the system behavior, and then decide the values of K_m and K_c.

Determining Controller Coefficients (K_m and K_c):

The simulations assume that the vehicle travels a distance of 250.0 meters, after the vehicle travels 22.4 meters, weeds appear for the remaining length, which causes the computation time of tasks increase and the miss ratio greater than zero. After the vehicle travels 156.8 meters, no weeds exist, which causes the computation time of each task decrease. Figure 7 shows the results when we change K_m from 0.5 to 3.0 while fixing K_c at 1.0. Since K_m is used to decrease the speed of the vehicle, we observe the decreasing edge of the figure, we notice that when $K_m = 2.0$, the speed curve is closest to the ideal speed curve, and the miss ratio curve is also closest to the ideal miss ratio curve. In the third part of Figure 7, we see that when $K_m < 2.0$, the time used to travel the 250.0 meters increase when K_m increases, this is because near the decreasing edge of the ideal speed curve, larger K_m can make the speed decrease more quickly. When $K_m = 2.0$, the time is less than the time used when $K_m = 1.5$, this is because the speed is decreased too much when $K_m = 1.5$. When $K_m > 2.0$, the speed is decreased much more, this is why the time is greater than the time used when $K_m = 2.0$. Thus, we decide to use $K_m = 2.0$. Since the value of K_c will not affect the decrease of speed, so we only use one value ($K_c = 1.0$) when we decide the value of K_m.

After choosing K_m, we determined the value of K_c. In the simulation, we fixed K_m at 2.0, and changed the value of K_c between 1.0 and 5.67. From Figure 8, we see that when $K_c = 5.0$, the speed curve is closest to the ideal speed curve. When $K_c = 5.67$, the curve fluctuates, this is because when the controller detects the speed is larger, it decreases the speed, but the change is too large, which leads to a low speed, then the controller detects that the speed is too low and increases the speed. But the new increase lead to a too large speed again, and the above procedure keeps repeating. This is the reason for the fluctuation. Thus 5.0 is a good value for K_c. The third part of Figure 8 show the time used for each K_c value. We see that when $K_c \leq 5.0$, the time decreases when K_c increases, and when $K_c = 5.67$, the time increases. This is because when there is no fluctuation, larger K_c will make the speed increase more quickly when the speed is too low. Thus, we choose K_c to be 5.0.

4 Simulation Studies

In this section, we measured the time that the vehicle needs to travel around 32km in different field conditions. We use $K_c = 5.0$ and $K_m = 2.0$ in the simulation. To simulate different field conditions, we assume the ranges of patches of weed as shown in Table 2.

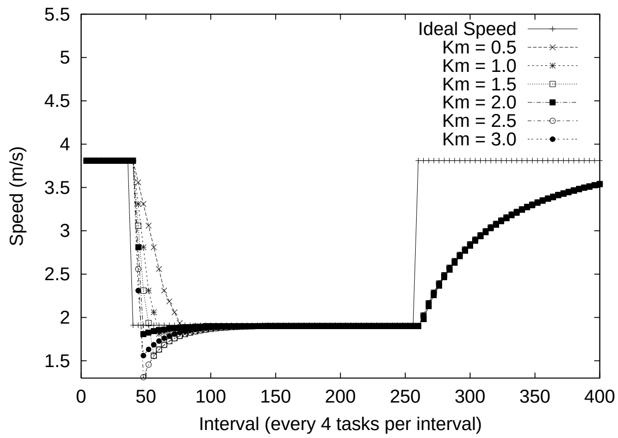
Weed	No Weed	Unknown
50% [62.5, 125]	40% [50, 100]	10% [12.5, 25]
60% [100, 200]	30% [50, 100]	10% [16.67, 33.3]
70% [175, 350]	20% [50, 100]	10% [25, 50]
80% [400, 800]	10% [50, 100]	10% [50, 100]
90% [450, 900]	0% 0	10% [50, 100]

Table 2. Field conditions

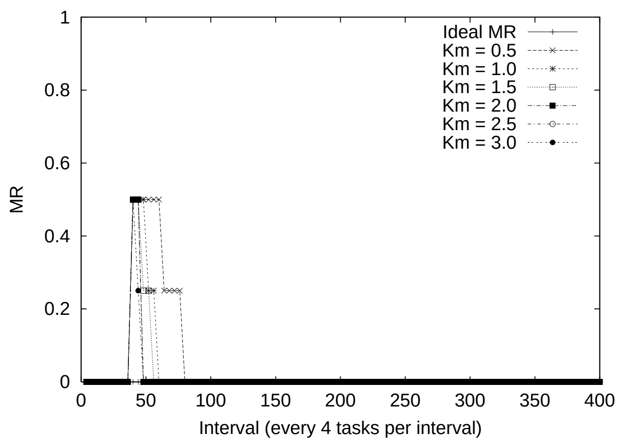
In Table 2, each row stands for a field condition. For each field condition, we choose the weed length in the range specified in the first column of the table, the percentage number refers to the percentage of the weed specified by the first column in the whole field. Similarly, we choose the no-weed length in the range specified in the second column of the table, the percentage number refers to the percentage of the no-weed area specified by the second column in the whole field. Besides, we choose another length of area, which may contains weed or not, uniformly in the range specified in the third column of the table, the percentage number refers to the percentage of length chosen from the third column. We decide the area has weed or not randomly. These three steps are repeated in the simulation. Thus, as an example, the field specified by the first low contains $55\%(50\% + \frac{10\%}{2} = 55\%)$ weed, and $45\%(40\% + \frac{10\%}{2} = 45\%)$ no-weed area. The width of weed or no-weed is the same as the width of the image area the camera can get.

We compare the performance of the adaptive approach using feedback with that of nonadaptive approaches operating at high speed and low speed. We use time (cost) and miss ratio (MR) as the performance metrics; and we also plot the speed of the vehicle.

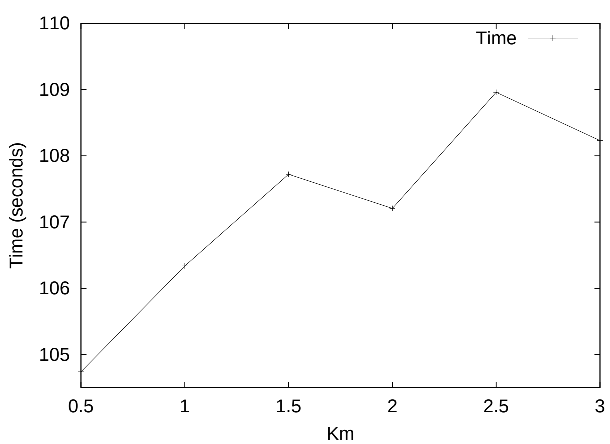
Figure 9 shows the time used for the three approaches. We see that when more weeds appear in the field (weed percentage changes from 50% to 90%), the time used in the



(a) Speed

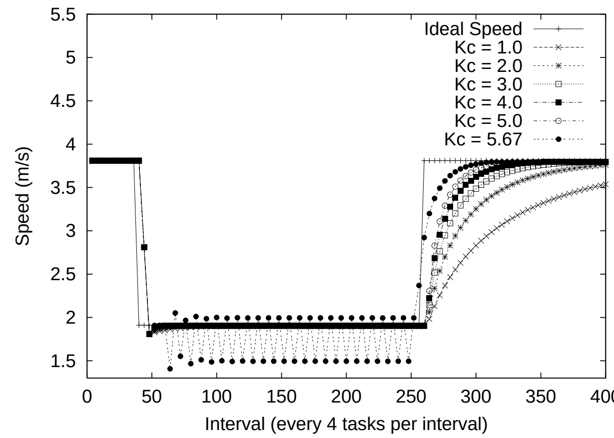


(b) Miss ratio

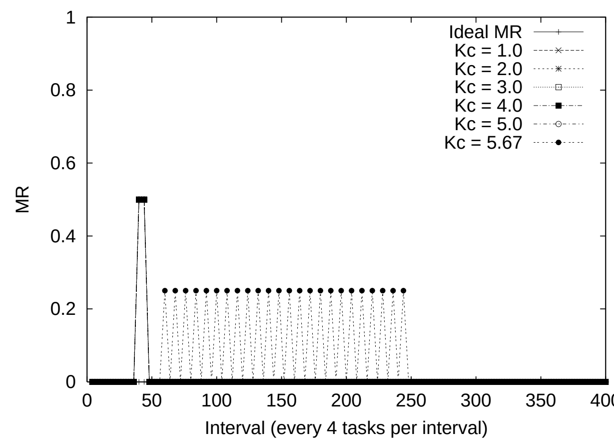


(c) Time

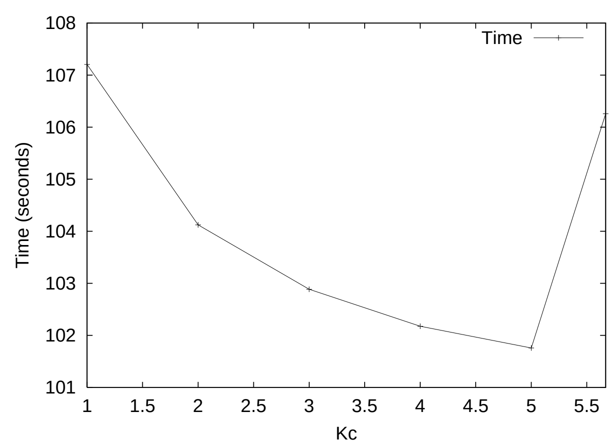
Figure 7. The speed, miss ratio and time when fixing K_c at 1.0 and adjusting K_m from 0.5 to 3.0.



(a) Speed



(b) Miss ratio



(c) Time

Figure 8. The speed, miss ratio and time when fixing K_m at 2.0 and adjusting K_c

feedback approach increases when the weed percentage increases, but the time is always between the times used in the other two approaches.

Figure 10 shows the miss ratio of each approach. In the

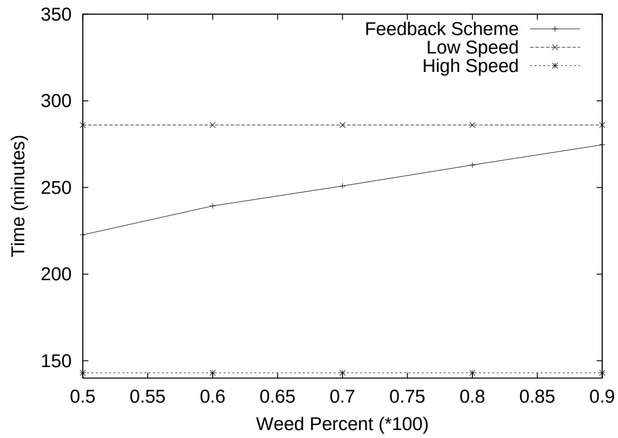


Figure 9. Time to process difference fields

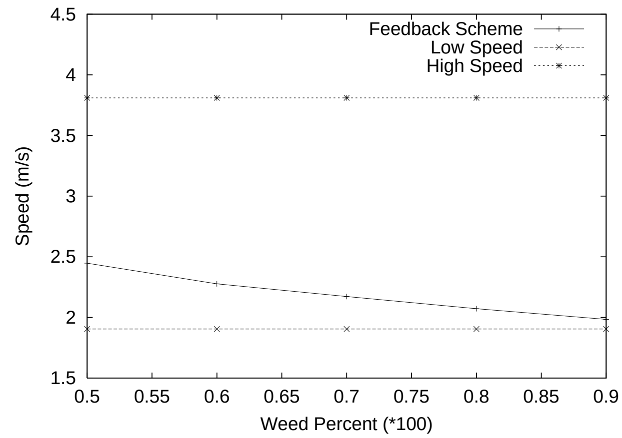


Figure 11. Average speed on different fields

approach using high speed, the miss ratio increases when the weed percentage increases. In the approach using low speed, no tasks miss deadlines, but the cost is very high because it takes more time. In the approach using feedback, the miss ratio is very small, and the miss ratio decreases when the weed percentage increases. This is because when the weed percentage is large, the weed patches are big, then decrease and increase of speed happen less often, thus less tasks miss their deadlines.

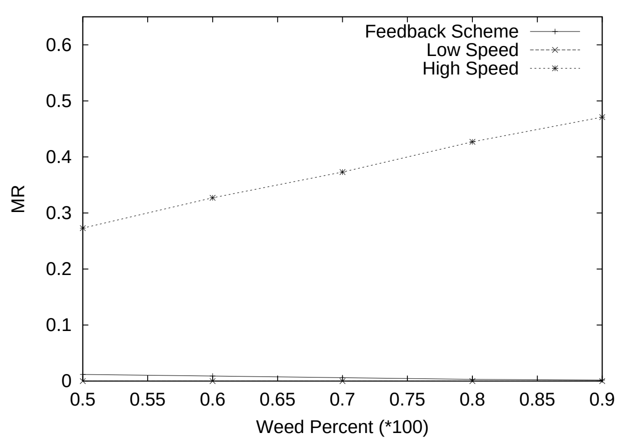


Figure 10. MR on different fields

Figure 11 shows the average speed for each approach. The speeds are got from the time, so the reasons are obvious.

From Figure 9, 10 and 11, we also find that the approach with low speed always use the longest time and achieve a zero miss ratio. The approach with high speed always use the shortest time and have the highest miss ratio. The feedback approach is a trade-off between this two approaches, it adjust the speed of the vehicle and achieve a nearly zero miss ratio, in addition, the performances offered are between the other two approaches. Our studies demonstrate that the proposed feedback based approach can achieve a desired miss ratio (set point), that is, a nearly zero miss ratio, while minimizing the cost.

5 Conclusions

In this paper, we proposed a feedback-based scheduling approach in real-time systems with autonomous vehicles. The goal is to achieve low miss ratio and high CPU utilization. The objective is achieved by feeding back system performances, adjusting mobile nodes' speed, which will lead to the adjusting of task parameters. We developed the new approach for selective herbicide spraying application in agricultural production. Simulations and analysis show that our approach can achieve a low miss ratio and high CPU utilization for the autonomous vehicle system.

Since real-time scheduling systems are nonlinear and uncertain, modeling the scheduling system as a linear system is difficult and fixed controller parameters do not result in satisfactory results, our future work includes online designing controller for fields with different weed densities. Besides, due to the heterogeneity among vehicles, our future work also includes allocating and scheduling of workload in an multi-vehicle system with nodes having heterogeneous capabilities in terms of sensing, processing, and actuation.

References

[1] C. Lu, J. A. Stankovic, G. Tao, and S.H. Son, "Design and evaluation of feedback control EDF scheduling algorithm", in Proc. *IEEE Real-Time Systems Symposium*, pp.56-67 1999.

[2] J. A. Stankovic, Chenyang Lu, S. H. Son, and G. Tao "The case for feedback control real-time scheduling", in Proc. *Euromicro Conference on Real-Time Systems*, pp.11-20, 1999.

[3] D. R. Sahoo, S. Swaminathan, R. Al-Omari, M. V. Salapaka, G. Manimaran, and A. K. Somani, "Feedback control for real-time scheduling", in Proc. *American Controls Conference*, vol.2, pp.1254-1259, 2002.

[4] R. Al-Omari, G. Manimaran, M. V. Salapaka, and A. K. Somani, "New algorithms for open-loop and closed-loop scheduling of real-time tasks based on execution time estimation", in Proc. *Intl. Parallel and Distributed Processing Symposium*, pp.7-14, 2003.

[5] G. Buttzaao, G. Lipari and L. Abeni, "Elastic task model for adaptive rate control", in Proc. *Real-Time Systems Symposium*, pp.286-295, 1998.

[6] G. Buttazzo and L. Abeni, " Adaptive workload management through elastic scheduling", *Real-Time Systems*, vol.23, no.1-2, pp.7-24, 2002.

[7] T. Hague, J. A. Marchant, N. D. Tillett, " Ground based sensing systems for autonomous agricultural vehicles", *Computers and Electronics in Agriculture*, vol.15, no.1-2, pp.11-28, 2000.

[8] Toru Torii, "Research in autonomous agriculture vehicles", *Computers and Electronics in Agriculture*, vol.25, no.1-2, pp.133-153, Japan, Jan. 2000.

[9] B. L. Steward, L. F. Tian and L. Tang, "Distance-based control system for machine vision-based selective spraying", *Trans. of American Society of Agricultural Engineers*, vol.45, no.5, pp.1255-1262, June 2002.

[10] D. A. Mortensen, G. A. Johnson, D. Y. Wyse and A. R. Martin, "Managing spatially variable weed populations", p.397-415, *Site-Specific Management for Agricultural Systems*, P. C. Roberts, R. H. Rust, and W. E. Larson, eds. Madison, 1995.

[11] C. Siva Ram Murthy and G. Manimaran,"Resource Management in Real-Time Systems and Networks", *MIT Press*, April 2001.