Using a Control Architecture for Real-Time Dynamic Resource Allocation

Yong Xun Mieczyslaw M. Kokar
Kenneth Baclawski

Northeastern University
Boston, Massachusetts

Abstract

In this paper we apply control-theory techniques to a real-time resource allocation problem. The specific problem is to control a sensor that monitors sources of electronic radiation in the environment (such as radars). The sources emit radiation in different frequency bands and with various illumination patterns and periodicities. For the sensor to detect an illumination by an emitter, the sensor must be tuned to the frequency band being emitted, and it must be sensing at the time when the illumination occurs. The goal is to minimize the time spent attempting to observe each emitter, while also maximizing the probability of detecting an illumination by the emitter. The environment is highly dynamic; both emitters and sensors can move, and illumination patterns can vary over time. To build a scheduler for this resource allocation problem, we first formulated it as a control problem. We then mapped it to a two-loop feedback control architecture. Finally, we evaluated our solution in a simulated environment. The results were compared with two other scheduling techniques. We show that the control-theory based scheduler outperforms both of the other schedulers. Furthermore, our solution uses only about 20% of the observation time as the fixed scan schedule. The main reason for the performance advantages of our solution is that neither of the other two schedulers can adapt to dynamic changes in the environment.

1 Introduction

Self-Controlling Software has been recently recognized as a new paradigm in software architecture (cf. [3, 5, 6]). In one of our earlier papers [4] we described a *control theory metaphor for self-controlling software*. This formulation was the result of our experiments with adaptive software (cf. [1, 2]). In this paper we apply control-theory techniques to a real-time resource allocation problem.

The specific problem is to control a sensor that monitors sources of electronic radiation in the environment (such as radars). The sources emit radiation in different frequency bands and with various illumination patterns and periodicities. For the sensor to detect an illumination by an emitter, the sensor must be tuned to the frequency band being emitted, and it must be sensing at the time when the illumination occurs. Because of the large number of actual and potential emitters in an environment, it is important to minimize the time spent attempting to observe each emitter, while also maximizing the probability of detecting an illumination by the emitter. It is also important to keep frequency bands available for other uses, such as jamming. This results in additional pressure to reduce the time scheduled for the sensor to perform observations in each frequency band.

The environment is highly dynamic. Both emitters and sensors can move, and illumination patterns can vary over time. Furthermore, events occur over a very large range of time scales. Individual radiation pulses may be as brief as a nanosecond, while illumination periodicities can be as long as several seconds.

To build a scheduler for this resource allocation problem, we first formulated it as a control problem. We then mapped it to a two-level feedback control architecture. Finally, we evaluated our solution in a simulated environment. The results were compared with two other scheduling techniques. The most commonly used scheduler uses a schedule determined in advance of a mission. This type of scheduler is called a fixed scan scheduler. Another scheduling technique uses prior knowledge about the emitters. We show that the control-theory based scheduler outperforms both of the other schedulers. Furthermore, our solution uses only about 20% of the observation time as the fixed scan schedule, and our solution does not depend as much on a priori knowledge about the emitters as the second scheduling technique. The main reason for the performance advantages of our solution is that neither of the other two schedulers can adapt to dynamic changes in the environment.

In Section 2 we introduce the basic background of sensors. Fixed scan schedules are introduced in Section 3. An algorithm for constructing fixed scan schedules is presented, and some examples are given. Although this scheduling technique is the most commonly used one, it does not have very good performance, and the reasons for this are discussed in some detail. In Section 4, a different scheduling methodology is introduced based on control theory. We introduce a general architecture for control theory, and then map this particular resource allocation problem to a control theory architecture. Simulation results are presented in Section 5. These results show how fixed scan scheduling compares with our control theory approach to dynamic scan scheduling. We also simulate a dynamic scan scheduling technique that does not use control theory and compare it with the other two scheduling techniques. We end the paper in Section 6 with our conclusions and an outline of future research directions.

2 Basic Background of Sensors

The sensors being considered in this paper are receivers located on a moving platform such as an aircraft. When aircraft are operating in hostile territory, attempts will be made to detect and to track their movements. This is most often done by using ground or missile based radars. Such radars emit electromagnetic radiation that illuminates the aircraft. Radiation reflected from the surfaces of the aircraft is used by the radar for detection, identification and tracking purposes. We refer to these radars as *emitters*.

Because of the tactical importance of these radars, it is important for aircraft to detect that they are being illuminated. For this reason, aircraft are equiped with receivers that attempt to sense when the aircraft is being illuminated by an emitter. The aircraft receivers will be called *sensors*.

There are many kinds of emitter. The most common three kinds are searching emitters, tracking emitters and missile emitters. Searching emitters and tracking emitters either search for targets (by scanning the environment) or track targets, depending on their antenna. A sensor can detect an emitter if the sensor is within the beam width from the central beam line of the emitter's antenna (cf. Figure 1). To detect an emitter, the sensor must be looking in the emitter's frequency band for a duration equal to at least three consecutive electromagnetic pulses produced by the emitter.

It is normally assumed that the kinds of emitter that might be encountered during a mission are known in advance. In particular, the system has data on the frequencies used by the emitters and the times between successive

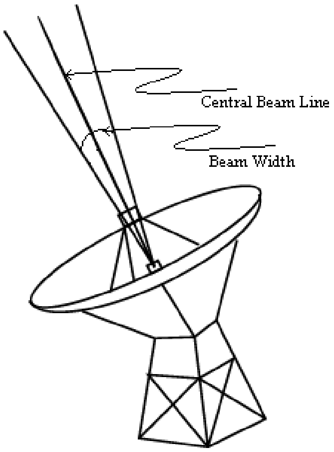

Figure 1: Radar

illuminations when the emitters are operating. It is not normally assumed that one will know when a given kind of emitter will begin operating or how long it will operate.

In Figure 2 we show a typical emitter illumination pattern. The radiation is produced in short pulses. These, in turn, form batches called *illuminations*. The pulses are shown here as square pulses, but the actual pulses are high frequency waveforms. The line over each illumination group is meant to represent the envelope of the illumination. It is not another signal.

While the pulses are produced electronically, the illuminations are usually the result of the mechanical motion of the radar antenna. This has some important consequences. The time between pulses can be very short and can be very regular. On the other hand, the time between successive illuminations is typically much longer and need not be as regular.

The most important emitter and sensor parameters are the following:

f_e - The frequency of the emitter's signal (waveform).

f_r - The frequency band of the sensor.

T_e - Pulse Repetition Interval (PRI) of the emitter. This is the time between successive pulses within a single illumination. Pulses are formed by modulating the signal by the frequency $1/T_e$.

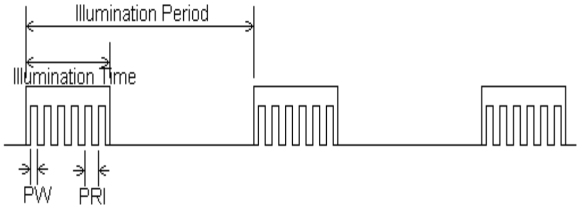

Figure 2: Signal

T_{ei} - The period of the illuminations. This is the time between successive illuminations as observed by the sensor.

τ_e - Duration of a single illumination. An illumination will usually consist of tens or hundreds of pulses.

P_e - Pulse width (PW) of the emitter. This is the amount of time during a single pulse.

T_{er} - Rotation period of the emitter antenna.

B_e - Beam width of the emitter. This is the angle within which a target can be effectively detected.

τ_d - Dwell time for the sensor. This is the amount of time that the sensor is tuned to a particular frequency band. The sensor can reliably detect that an emitter has illuminated the aircraft only if its frequency band f_r includes the frequency of the emitter f_e and if the dwell time τ_d is long enough for the emitter signal to be distinguished from the background noise. A typical requirement for reliable detection is that the dwell time includes at least three consecutive pulses by the emitter at a power level above a threshold.

t_{ds}, t_{de} - Times of the beginning and the end of the dwell, respectively.

2.1 Sensor Control and Scheduling

A sensor is controlled by giving it a command called a *Control Description Word* (CDW). A CDW specifies when and for how long the sensor is to be tuned to a particular frequency band. Each CDW commands a single dwell of the sensor. Mathematically, a CDW is a triple:

$$CDW = (t_{ds}, \tau_d, f_r). \tag{1}$$

A *scan schedule* (SS) is a sequence of CDWs, one per dwell:

$$SS = \{CDW_1, CDW_2, \ldots, CDW_n\}. \tag{2}$$

A scan schedule is simply the sequence of commands given to a sensor. Within a scan schedule, one can define the notion of the *revisit time*, τ_{er}, the time from the end of the last dwell until the beginning of the next dwell on a particular frequency band. From the end time of the last dwell and the revisit time, one can compute the start time of the next dwell.

The most basic constraint that any sensor must satisfy is called the *Capacity Constraint*. This constraint simply states in mathematical form the fact that a sensor can only be tuned to one frequency band at a time. For a scan schedule for which the number of frequency bands is a fixed number N and for which the dwell time for each frequency band is fixed, the capacity constraint is:

$$\sum_{i=1}^{N} \tau_d(i)/\tau_{er}(i) \leq 1 \tag{3}$$

Another way to view the Capacity Constraint is introduce the notion of the *duty cycle*. For each emitter, the fraction of the time allocated to this emitter is the emitter's duty cycle. It is the ratio of the dwell time to the revisit time for this emitter. The Capacity Constraint states that the total of all duty cycles can be no larger than 1.

While the Capacity Constraint is a necessary condition, it is not sufficient to ensure that a scan schedule exists. One limitation is that sensors require a small amount of time to switch from one frequency band to another. Another limitation that is more subtle is that the revisit times of different frequency bands may cause scheduling conflicts: two or more frequency bands may require the sensor at the same time. For this reason scheduling is an important part of the control of a sensor.

When an emitter has been detected, an attempt is made to identify the kind of emitter from a database of potential emitters. If this is successful, then the entry in the database is transferred to a table called the *Active*

Emitter Table (AET). The AET is the primary output of the sensor, and it can also be used for scheduling the sensor.

If the emitter cannot be identified as a known kind of emitter, then the system creates a new AET entry as well as a new database entry. This usually requires that the emitter be detected more than once in order to obtain an estimate for the illumination period. Needless to say, such unanticipated kinds of emitter have important tactical consequences.

3 Fixed Scan Scheduling

The most commonly used sensor scheduling technique is the *Fixed Scan Schedule* (FSS) which is defined by the following properties:

1. It is loaded at the launch time and is executed during the entire mission.
2. It consists of a fixed sequence of CDWs that do not change during the mission.
3. The CDWs for a specific emitter (frequency band) differ only in the start time t_{ds}, i.e., all the dwell times τ_d and the revisit times τ_{er} for that frequency band are the same.

Aside from the fact that the FSS is the most commonly used scheduling technique, it is important because it is usually the initial schedule for any other kind of scan scheduling. In the absence of any new information (i.e., in the absence of any emitter detections), an FSS has reasonably good performance for a known scenario.

An FSS has the advantage that it is simple and thus easy to implement. However, it is based on *a priori* knowledge about the environment, so it is not applicable in unknown environments or environments that change. In particular, since it does not have a feedback mechanism, it cannot adapt to changes in the environment.

3.1 Constructing Fixed Scan Schedules

There are several ways to construct an FSS. A *monotonic rate fixed scan scheduler* [?] is a general task scheduling technique that uses two pieces of

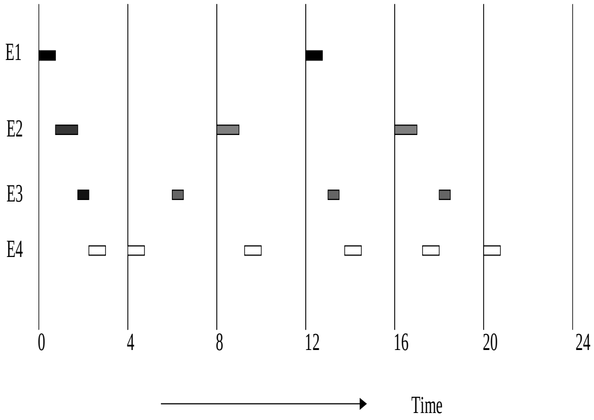

Figure 3: Example: A Fixed Scan Schedule

data about tasks: task execution time and period. For sensors, the execution time is the dwell time and the period is the revisit time. If one assumes that the dwell time is equal to K pulse repetition intervals (where K is typically equal to 3), then the following equations describe the monotonic rate scan scheduler:

$$\tau_d = 3T_e \tag{4}$$

$$\tau_{er} = \tau_e \tag{5}$$

This assumes that the emitters emit pulses in a uniform pattern. It would not apply to emitters that use a non-uniform pattern, such as radars that employ a stagger pattern in which a pattern of pulses is emitted repeatedly.

Figure 3 shows an example of a monotonic-rate FSS. This schedule includes only four emitters ($E1, E2, E3, E4$). Their characteristics (dwell time, illumination time, priority) are given by: $E1 : (0.5, 4, 1)$, $E2 : (0.3, 6, 2)$, $E3 : (0.8, 8, 3)$ and $E4 : (0.6, 12, 4)$, respectively.

The computation of a monotonic rate Fixed Scan Schedule starts by determining the least common multiple of the revisit times of the emitters. We call this the *scheduling cycle*. In the example of Figure 3, the revisit times are 4, 6, 8 and 12, so that the scheduling cycle is 24. In the next step, the emitter having the highest priority is allocated the sensor. This process is then repeated until the scan schedule has been determined for the entire scheduling cycle. If the Capacity Constraint (3) is violated, then some revisit times must be increased using some rule, such as increasing the revisit time of lower priority emitters until a scan schedule satisfies the Capacity Constraint and a schedule can be found. The schedule developed for one scheduling cycle is then repeated for all consecutive cycles.

3.2 Disadvantages of Fixed Scan Schedules

We now discuss the problems that arise when sensors are controlled using only an FSS. Consider, for example, a monotonic rate FSS and a simple scenario, consisting of a single emitter and a single sensor having the following parameters:

$$T_e = 2.0 \cdot 10^{-3} \text{ sec} \tag{6}$$

$$P_e = 1.5 \cdot 10^{-5} \text{ sec} \tag{7}$$

$$T_{er} = 4.0 \text{ sec} \tag{8}$$

$$B_e = 2^o \tag{9}$$

From these equations one can calculate the illumination time to be:

$$\tau_e = 4.0 \text{ sec} \cdot (2^o/360^o) = 0.02 \text{ sec} \tag{10}$$

The dwell and revisit times for a monotonic rate fixed scan scheduler will then be:

$$\tau_d = 3 \cdot T_e = 6.0 \cdot 10^{-3} \text{ sec} \tag{11}$$

$$\tau_{er} = \tau_e = 0.02 \text{ sec} \tag{12}$$

In a monotonic rate FSS for this scenario, at least 180 CDWs will be needed during one scheduling cycle in order to guarantee the detection of two consecutive illuminations. This corresponds to a duty cycle of

$$d = (180 \cdot 6 \cdot 10^{-3})/4 = 0.27 \tag{13}$$

for just this one emitter. When there are tens or even hundreds of emitters in the environment, this kind of scheduling will not be efficient. The goal is to reduce the number of CDWs required by each emitter, or, equivalently, to reduce the emitter duty cycles.

Using an FSS with a larger revisit time is not the answer, because in such a case the probability that an illumination will be missed increases. Accordingly, it is necessary for the scan schedule to respond to the environment in some way. A scan schedule that does this will be called a *dynamic scan schedule* (DSS). Rather than being a fixed scheduling pattern, a dynamic scan schedule is an algorithm that dynamically modifies the scan schedule in response to detection events.

There are several ways that one can introduce dynamics into a scan schedule. One possibility is to modify the FSS algorithm in the following way:

1. An FSS is loaded at the launch time and is executed until a detection occurs.

2. When a detection and identification occurs, the FSS is recomputed so that the identified emitter is only observed at the predicted illumination times as specified by the entry in the AET.

3. After recomputation the CDWs for a specific emitter differ only in the start time t_{ds}, i.e., all the dwell times τ_d for the frequency band are the same.

We will call this technique *informed dynamic scan scheduling.* The only difference between this technique and the FSS is that detected emitters will have a much longer revisit time.

While informed dynamic scan schedule seems like it solves the problem of resource allocation in this case, it has some serious disadvantages. One obvious one is that it cannot deal with emitters that are not in the database. A more serious problem is that the illuminations of the aircraft will vary over time for a variety of reasons. The radar antenna, being a mechanical device, could speed up or slow down. However, a more significant effect is due to the motion of the aircraft. As the aircraft moves relative to the emitter, the apparent time between illuminations changes. For instance, consider an emitter with a 6 sec illumination period (i.e., the radar antenna rotates by one full cycle in 6 sec). Now suppose that the aircraft is 60,000 feet away from the radar and is moving in a circle around the radar at 1000 feet/sec, in a direction opposite to the rotation of the radar. In such a case while the radar rotates by 360 degrees, the aircraft moves by 6,000 feet, which is equivalent to about 6 degrees. Consequently, the apparent time between illuminations will be shorter by approximately 0.1 sec. This might seem like an insignificant shift, but since the sensor dwells on a particular target for the duration of just three Pulse Repetition Intervals, say $T_e = 6 \cdot 10^{-3}$ sec,

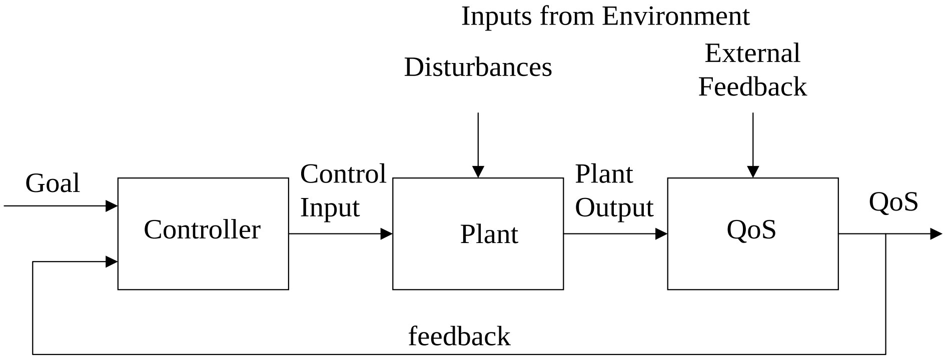

Figure 4: Feedback Control Architecture

this shift is actually quite important. In fact, this dwell time is only about 6% of the 0.1 sec shift, which is more than sufficient to cause a detection failure.

The question is how to determine the revisit time so that the impact of dynamics, like in the example above, is reduced. So far we can say that the scheduler should have the following characteristics. First, it must be adaptable to changes in the environment. It should be able to adjust its revisit time. Second, it must have some kind of feedback mechanism to adapt to the environment. In order to achieve such a goal, we use the control theory metaphor in the development of a Dynamic Scan Scheduler.

4 Mapping to Control Theory

We now show how the sensor scheduling problem can be mapped to a control theory architecture. We first review some background in control theory. A specific mapping to a control theory architecture is then proposed. This mapping is based on the variables that must be varied in order to adapt to the dynamically changing environment.

4.1 Background in Control Theory

In the control approach in Figure 4, a system consists of two basic elements: a Plant (the system being controlled) and a Controller. The Controller has some reference input, or control Goal. It is capable of assessing the performance of the Plant with respect to the Goal. For this, the Controller implements a control law, i.e., an algorithm (also called a control algorithm) that changes some of the Plant's inputs (control inputs) so that the Plant's output is directed towards the Goal. The Plant's output must be measurable in the same coordinates as the Goal. For this, it is sometimes necessary to implement a Quality-of-Service (QoS) module that assesses Plant's output in terms of a Quality-of-Service measure. (Note, that in the control literature QoS is not considered as a separate module, it is simply part of the controller.) The connection of the QoS parameter to the Controller is called *feedback*. The feedback is compared with the Goal. The difference between the two is called the *error*; it is the input to the Controller. Such a scheme is called a *feedback control scheme*.

4.2 Mapping to a Control-Based Architecture

Unlike FSS, where a schedule was fixed from one CDW to the next for a given emitter, our control-based DSS will continually vary the revisit time from one CDW to the next. We begin the process of mapping this idea to a control-based architecture by identifying what the Plant and QoS should be in this case. The Plant includes the sensor and the sensor's information processing. The Plant's function is to detect illuminations by the emitters. The Plant thus reads the signal within the time interval provided by Revisit Time and Dwell Time, i.e., starting at the new Start Time (computed from Revisit Time and end of last dwell time) and ending after the Dwell Time completes. If the Plant encounters at least three emitter's pulses in its input, it declares a detection, otherwise it declares a no-detection.

The purpose of the control based scan scheduler is to optimize the probability of detection, while keeping the duty cycle as low as possible. The metric we wish to optimize is the ratio between the number of detected illuminations, N_d, and the number of all possible illuminations, N_p. We call this ratio the *hit rate*:

$$QoS = \frac{N_d}{N_p} \tag{14}$$

This Quality of Service measure is specific to one emitter. Accordingly, the overall design of our system is to have one controller for each (potential) emitter. The controllers for the various emitters are then combined by

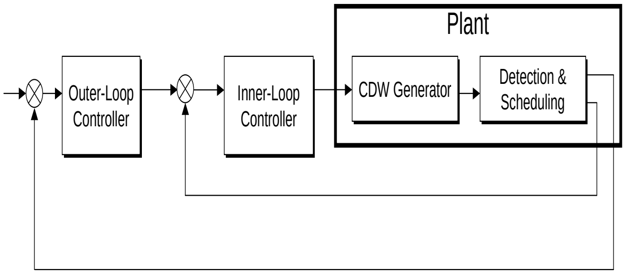

Figure 5: Control Architecture

using a scheduling algorithm. We first discuss how we map the control theory paradigm to the case of a single emitter, and later we discuss how the scheduling is accomplished.

While the number of detected illuminations can be measured directly, as described above, the value of the number of possible illuminations is not directly measurable. Instead, we assume that there is a database of *a priori* knowledge about potential kinds of emitter, from which one can determine what kinds of emitter are expected in each frequency band. The number of detected illuminations is measured by simply keeping count of detections, i.e., incrementing this variable by one after each detection. Similarly, the number of CDWs (N_{CDW}) is measured through counting.

In this scenario, we also want to reduce the number of CDWs, i.e., the number of dwells, but without lowering the detection performance. Therefore, the QoS should also depend on the number of CDWs in the scan schedule. This gives us another formula for the Quality of Service that we denote QoS':

$$QoS' = \frac{N_d}{N_p \cdot N_{CDW}} \tag{15}$$

The Goal of the Controller (or the reference input) should be expressed in terms of QoS'. Notice that between detections only N_{CDW} changes (grows) and thus QoS' is either zero (before the first detection) or gets smaller with each new revisit. The Controller then would have to either decrease the revisit time, which would result in even more dwells (N_{CDW}), or increase the revisit time, which might result in missing some illuminations. This is the result of having two different time scales of control: one (coarse) for detections and one (fine) for revisits. Consequently, we propose a two-loop control structure as shown in Figure 5. In this scheme we have two controllers: an *inner-loop controller* and an *outer-loop controller*. The inner loop controller controls the QoS' by changing the revisit time for a given emitter (τ_{er}). This affects the number of dwells (and thus N_{CDW}). The outer-loop controller controls the QoS. It provides the reference input for the inner-loop controller. The reference input for the outer-loop controller is set to 1. It is a desired outcome to have $QoS = 1$. It is achieved when there is one detection for each illumination, while the number of dwells (N_{CDW}) is also equal to 1. More than one detection per one illumination is an indication of false detections (false alarms), while more than one revisit per one illumination means resources have been wasted by requesting unnecessary dwells. For this scheme to work, N_{CDW} cannot be equal to 0. Consequently, we set it to 1 after each detection.

For the inner-loop controller we use the proportional integral (PI) control algorithm:

$$\frac{Y(s)}{X(s)} = K(1 + \frac{1}{T_i \cdot s}) \tag{16}$$

In the left-hand side of this equation, X is the error (i.e., the difference between the reference input and the value of the QoS) and Y, the controller output, is the value of the next Revisit Time (τ_{er}). In the right-hand side, K and T_i are constants.

For the outer-loop controller we use only proportional control. The output of this controller is the reference input to the secondary controller. The feedback for this controller is the value of the QoS', as defined above.

The constants in the PI control law are different for different emitters. For instance, for the outer-loop controller and emitter E_1 the value is: $K = 0.1$. For the inner-loop controller for the same emitter we use $K = 0.45$ and $T_i = 17,000$. The information needed for computing these constants is discussed below.

The proposed scheme does have one problem. If an emitter is not active, but the controller is used to generate CDWs to detect the emitter, then the revisit time generated by the control method could become shorter and shorter until it could be even shorter than that of an FSS. The result would be a larger duty cycle for this emitter than would be the case for an FSS.

This problem is solved by modifying the control theory mechanism in two ways. First, we limit the revisit time to the illumination time. Second, the scan schedule is initialized by using an FSS to detect the emitters, switching to the dynamic scheme when an emitter is detected. However, even when an emitter is detected, it is still necessary to identify it, yielding information such as its frequency, rotation time, beam width, illumination time, illumination period and pulse repetition interval. This information is used in calculating the QoS and the control. As discussed above, the entries in this table are either based on *a priori* knowledge about the potential emitters in the environment or are constructed based on observations during the mission.

4.3 Scheduling the Sensor

The control scheme described above applies to controlling the number of CDWs, or the number of dwells for a single emitter. We have one such controller for each emitter. However, we have only one sensor and thus the sensor needs to be scheduled among different emitters. In this paper we present our initial results by using a simple scheduling first-in first-out (FIFO) algorithm. The FIFO scheduler makes its decision based upon the ends of dwell times and revisit times for the emitters. It adds the revisit time to the end of last dwell time to compute the start time for each emitter. Then it selects the emitter with the start time that is smallest and larger than the current time. In other words, it selects the smallest start time among those that have not expired, yet. If the start time for a specific emitter expires, i.e., it is smaller than the current time, then this request is dropped from the active requests. Note that this scheme is slightly different from the scheme used by computer operating systems, since our scheme is essentially preemptive.

Note that lowering the number of CDWs, which was the goal of our scheme described above, results in a much smaller computational burden (workload) on the scheduler. Nevertheless, the scheduling problem is still quite difficult compared to an FSS because revisit times must be continually recomputed rather than being fixed.

5 Simulation Results

We simulated eight emitters. The generation of the CDWs for each emitter was controlled by the controllers described above. The scheduling of the sensor among the eight emitters was implemented using the $FIFO$ scheduling policy. To implement the simulations we used Matlab and the Simulink toolbox.

In Figure 6 we show the results of the evaluation of our control based DSS versus an FSS and an *Informed DSS*. In this figure the top diagram shows probabilities of detection for eight emitters. The left bar (shown in blue) is for a Fixed Scan Scheduler. The middle bar (green) is for our control based DSS and the right one (red) is for an Informed DSS. The second diagram shows CDWs generated by a Fixed Scan Schedule. Each stroke represents one CDW. The horizontal axis represents time. The two bottom diagrams show the CDWs for the Informed DSS (bottom) and our control based DSS (next from bottom).

6 Conclusions and Future Research

From these figures we can see that the performance of the control based DSS is dramatically better than that of the FSS in both accuracy and the number of CDWs. The probability of detection (hit rate) for the control based DSS is much higher and, at the same time, the number of CDWs is reduced to only one tenth of the CDWs in the FSS. Moreover, our control based DSS has shown a much better performance than the Informed DSS. The probability of detection is significantly higher, while the number of CDWs is somewhat higher. This is understandable since our DSS does not assume that the illumination period is constant. Instead, it settles on a fixed period only through the application of its control law. This gives it the robustness that is especially important when the environment undergoes dynamic changes.

Acknowledgments

References

[1] Y. A. Eracar and M. M. Kokar. An experiment in using control techniques in software engineering. In *Proceedings of the 1997 IEEE International Symposium on Intelligent Control*, pages 275–280, 1997.

[2] Y. A. Eracar and M. M. Kokar. An architecture for software that adapts to changes in requirements. *Journal of Systems and Software*, 50:209–219, 2000.

[3] G. Karsai and J. Sztipanovits. A model-based approch to self-adaptive software. *IEEE Intelligent Systems*, May/June 1999:46–53, 1999.

[4] M. M. Kokar, K. Baclawski, and Y. Eracar. Control theory-based foundations of self-controlling software. *IEEE Intelligent Systems*, May/June 1999:37–45, 1999.

[5] R. Laddaga. Creating robust software through self-adaptation. *IEEE Intelligent Systems*, May/June 1999:26–29, 1999.

[6] P. Robertson and J. M. Brady. Adaptive image analysis for aerial surveillance. *IEEE Intelligent Systems*, May/June 1999:30–36, 1999.

Things to Fix

1. Yong - Need some references, especially on Scheduling, but also on dynamics, control, architectures, and such
2. Yong - need more detail on the simulations - how many, ...
3. Mitch - It would be good to have more discussion on the issue of the application of control theory to software engineering in general. As it is now, we have more emphasis on the scenario.
4. Yong - Could split the results into two figures, one for CDWs and one for probability of detection.
5. Mitch & Ken - Future directions.

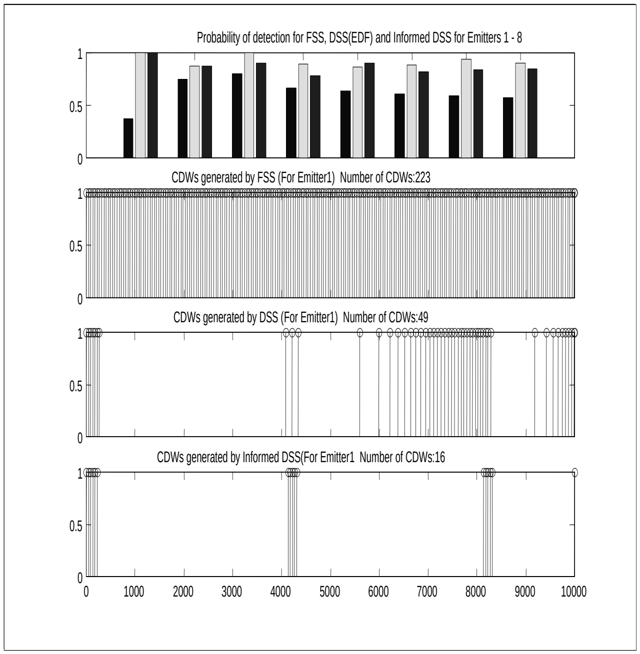

Figure 6: Results