TASK PRIORITIZATION ON PHASED-ARRAY RADAR SCHEDULER FOR ADAPTIVE WEATHER SENSING

Ricardo Reinoso-Rondinel[1,2,*], Sebastián Torres[2,3,4], and Tian-Y. Yu[1,2]

[1] School of Electrical and Computer Engineering, The University of Oklahoma, Norman, OK
[2] Atmospheric Radar Research Center, The University of Oklahoma, Norman, OK
[3] Cooperative Institute for Mesoscale Meteorological Studies, The University of Oklahoma, Norman, OK
[4] and NOAA/OAR National Severe Storm Laboratory, Norman, OK

Abstract

Electronically steered phased array radar was developed in mid-1960s mainly for military applications. It has the capability of instantaneously and dynamically controlling beam position on a pulse-to-pulse basis, which allows a single radar to perform multiple functions such as search, target tracking, and weapon controls. The recent-installed phased array radar (PAR) at the National Weather Radar Testbed (NWRT) in Norman, Oklahoma is the first phased array system in the nation dedicated to weather radar research and can electronically steer the beam in both azimuth and elevation. To fully unleash the power of the PAR for adaptive weather sensing, scheduling multiple competitive tasks (i.e. surveillance and storm cells tracking) in a sequence to meet the requirement of the update time for each task is the core of this study.

Time Balance (TB) is an adaptive process that schedules competing tasks, by balancing the available radar time and the time demanded by each task. When tasks demand more time than the time provided by the radar, the radar is in the so-called overload condition. In this work, simulated radar data are used to compare the performance of TB-based scanning strategies. Several methods are proposed to mitigate the overload condition and their performance is assessed and compared to that of conventional Volume Coverage Pattern (VCP) used in the operational Weather Surveillance Radar- 1988 Doppler (WSR-88D).

1. INTRODUCTION

Continuous technology upgrades on the Weather Surveillance Radar-1988 Doppler (WSR-88D) over two decades have considerably benefited both research and operational communities. The National Weather Service has shown improvement in warning of severe weather after the installation of WSR-88D (Polger et al. 1994). WSR-88D surveils the atmosphere by mechanically rotating the antenna 360°in azimuth at a number of elevation angles. Scanning patterns are known as volume coverage patterns (VCP) and lead to update times from 4 to 6 min for convective storms in order to provide Doppler spectral moments with the required accuracy (ROC 2007). However, rapid updates are often desirable for understanding fast-evolving weather systems (e.g., Steadham et al. 2002). Although fast update times can be achieved by increasing the antenna rotation rate, the accuracy of spectral moments is usually degraded for the same spatial resolution because fewer samples are available in the dwell time. Therefore, a good compromise is to accomplish fast revisits over regions of interest maintaining data accuracy while nonhazardous regions can be covered at smaller revisit rates. However, this is not feasible with conventional radars.

Phased-array radar (PAR) technology was developed in the mid-1960s primarily for military use (Skolnik 2001). PARs are capable of steering the beam electronically on a pulse-by-pulse basis. This beam agility makes the PAR an ideal platform to simultaneously perform multiple functions such as surveillance, multitarget tracking, and weapon guidance. Although, for a multifunction radar, all tasks are competing for a finite radar resource. Therefore, it is important to solve this resource management problem; that is, how to allocate radar resources in an optimal way by executing competing tasks in sequence (e.g., Vannicola et al. 1993; Capraro et al. 2006; Haykin 2006; Gini and Rangaswamy 2008). One of such scheduling algorithms that is based on the concept of time balance (TB) was developed by Stafford (1990) and applied to PAR. Recently, the TB scheduling algorithm was extended for weather applications (Reinoso-Rondinel et al. 2009).

The PAR installed in the National Weather Radar Testbed (NWRT) in Norman, Oklahoma has been available to research communities since September 2003 (Forsyth et al. 2005). The PAR operates at S-band

* *Corresponding author address:* Ricardo Reinoso-Rondinel, The National Weather Center, 120 David L. Boren Blvd., Rm 5900, Norman, OK 73072-7307; e-mail: rein3@ou.edu

and is able to electronically steer in elevation and azimuth within any given 90°sector. Moreover, recent experiments with the NWRT PAR have demonstrated better and precise characterization of fast-evolving weather systems with faster updates than the WSR-88D (Zrnić et al. 2007; Heinselman et al. 2008). Thus, the NWRT PAR is expected to execute both, multiple weather tracking and a volumetric surveillance to collect data of interest and at the same time detect new weather developments.

In our previous work, we illustrated the effectiveness of the TB algorithm to schedule storm tracking and weather surveillance tasks (Reinoso-Rondinel et al. 2009). However, if functions request more radar resources than are available, the radar is in the so-called overload condition, which requires special consideration. Methods to mitigate the overload condition are introduced in section 2. In section 3, the standard TB algorithm (Reinoso-Rondinel et al. 2009) is extended using these overload mitigation techniques. To demonstrate the performance of these, simulations based on interpolation of real data to a finer time scale are conducted in section 4. Finally, a summary and conclusion are given in section 5.

2. RESOURCE UTILIZATION CONTROL FOR ADAPTIVE WEATHER SENSING

Conventional radars that use mechanical scanning (e.g., the WSR-88D) typically scan a volumetric region with update times of 4 to 5 min. As a result, the update time for each storm cell is the same and equal to the revisit time of the volumetric region. On the other hand, a PAR system is flexible enough so that it can revisit multiple storms at different rates to better capture the evolution of weather phenomena of interest.

In this work, two radar functions are considered: storm tracking and surveillance. These compete for radar time and need to be properly executed. For storm tracking, it is assumed that the information about the size and location of each storm to be scanned is known (e.g., from a storm tracking algorithm). Tracking of each storm cell is defined as a tracking task with its own update time. In contrast, the purpose of the surveillance (herein referred to as the surveillance task), is to scan the volumetric regions where no storms are identified to keep track of current storm cells and to ensure the detection of newly developing phenomena.

The parameters that define storm tracking and surveillance tasks were presented by Reinoso-Rondinel et al. (2009) and are reviewed next. Each task is defined by its task time (T) and update time (U). For tracking, task time is the total time needed to volumetrically scan the identified storm. The update time, however, may be set based on users needs. For each task, the occupancy (O) is defined as the ratio of task time and update time (TU^{-1}) (Manners 1990). In general, it is desirable that the total occupancy for all tasks adds up to 100% so that the radar resources are fully allocated, That is,

$$O_T + O_S = \sum_{i=1}^{N} O_i + O_S = 100\%, \qquad (1)$$

where O_T is the total requested tracking task occupancy, N is the number of storms being tracked, O_i = $T_i {U_i}^{-1}$ is the occupancy for the ith tracking task, and O_S is the surveillance task occupancy. The task time for surveillance (T_S) is the total time to scan nonstorm regions and the update time for surveillance (U_S) comes from O_S = $T_S {U_S}^{-1}$ assuming full use of the radar resources. If O_T is more than 100%, the radar is referred to as being overloaded. In this situation storm tracking tasks will be unavoidably delayed and the surveillance task, which usually runs with a lower priority, will not run until the overload condition disappears. This becomes a serious problem since it is desired that the surveillance task is executed within a time interval not too long so that it can keep track of current storm cells and achieve early detection of new cells. To maintain surveillance execution while handling the overload condition two approaches are presented next.

2.1. Adjusting task update times

This approach handles the overload condition by adjusting the requested update times for storm tracking tasks while setting a maximum surveillance update time. The update time for a conventional scan (U_C) is defined as the time to complete a full scan over a 90°sector. Thus, the maximum surveillance update time is given by U_C.

The first step to adjust task update times is to set the surveillance update time equal to U_C and then estimate the remaining occupancy based on equation (1) as

$${O_T}^* = 100\% - T_S {U_C}^{-1}, \qquad (2)$$

where ${O_T}^*$ is the total occupancy available for storm tracking tasks after allocating occupancy for surveillance ($T_S {U_C}^{-1}$). The estimation of ${O_T}^*$ leads to three different radar resource load conditions. If ${O_T}^* < O_T$ the radar is said to be overloaded, if ${O_T}^* = O_T$ it is fully loaded, otherwise it is under loaded. For the overload condition, it is necessary to decrease the total requested tracking task occupancies. A total occupancy correction factor (f) can be computed such that

$$f^{-1} O_T = {O_T}^*. \qquad (3)$$

Thus, the new occupancy distribution for N tracking and surveillance tasks is described as:

$$O_T{}^* + O_S{}^* = \sum_{i=1}^{N} T_i (fU_i)^{-1} + T_S {U_C}^{-1} = 100\%, \quad (4)$$

where the term fU_i is the adjusted update time for the ith tracking task, and $O_S{}^*$ is the surveillance occupancy corresponding to an update time of U_C. Note that $f \geq 1$ and the total sum of storm tracking and surveillance occupancies is equal to 100%, i.e., the radar is fully loaded.

For the non-overload condition, the update times for storm tracking tasks may or may not be adjusted. If the surveillance update time is calculated from equation (1), storm tracking update times should remain the same for a full load condition. On the other hand, if surveillance update time is set to be U_C and more storms revisits are desired, then update times for storm tracking tasks can be adjusted using equation (4), where in this case $f \leq 1$.

For the non-overload condition, two quality measures were proposed by Reinoso-Rondinel et al. (2009) to quantify the improved performance offered by PAR adaptive sensing from conventional scans (VCPs), given that the accuracy of storm data is maintained. In this work, we analyze performance based on one of them, the revisit improvement factor. The total revisit improvement factor (I) for N tracking tasks is defined by the ratio of the total number of revisits for N storms using adaptive scanning over conventional scanning during a period of time U_m. Mathematically:

$$I = \frac{\sum_{i=1}^{N} U_m {U_i}^{-1}}{N U_m {U_C}^{-1}}. \quad (5)$$

For N tracking tasks, equation (5) quantifies the gain in the number of revisits yielded by adaptive weather sensing. It is of interest to know how revisit improvement factors change when storm tracking update times are adjusted. These results will be presented in section 4.

2.2. Task prioritization

As it was mentioned before, the multiple tasks competing for radar resources should be executed as requested. Although, if there are not sufficient radar resources, one or more of these tasks may be executed late. In this scenario, users can designate some tasks as highly important and request their executions on time. Under this idea, each task can be associated a priority level. Miranda et al. (2006) presented this idea in the context of military radar applications. In this work, it is assumed that users may provide a priority level for each task. Let P_i be the priority level of the ith task such that $P_i \in \mathbb{Z}^+$ and the minimum priority level is equal to 1, for $i = 1, 2, .., N'$. Here, $N' = N + 1$ so there are N tracking tasks and one surveillance task with associated priority levels.

To mitigate the lateness of some important tasks under an overload condition, tasks with higher priority levels need to be executed on time, while others may experience delays. Priority levels can change with time depending on users' needs and on the characteristics of the observed weather phenomena. For example, if surveillance has not been executed for a time period of U_C, it must be forced by maximizing its priority level, so that storm identification and tracking can be effective. Note that user-defined update times are not adjusted as in the previous approach so the overload condition still holds. Next, the TB algorithm is extended to handle the two overload mitigation approaches described above.

3. ADAPTIVE TIME BALANCE SCHEDULING ALGORITHM

The first scheduling algorithm for weather sensing based on TB was presented by Reinoso-Rondinel et al. (2009). Such algorithm associates a time balance variable (T_B) to each tracking task, where a positive time balance indicates the task is late for execution at any given time. Surveillance is only executed when the time balances of all tracking tasks are negative. Therefore, there is no time balance variable associated with the surveillance task. In addition, the surveillance task is executed in fragments. The time needed to dwell on a fragment region is defined as task fragment time (T_F). Only after all fragments are scheduled, the surveillance task is said to be completed. In this work, some ideas behind the TB principles are extended. Surveillance can have an associated T_B, that is, surveillance must be executed based on its requested update time to ensure the identification and tracking of present and new storm cells. Also, T_F = T_S, so surveillance is not fragmented. To incorporate the approach described in section 2.1, requested update times can be adjusted based on equation (4) to mitigate overload conditions or to increase the revisit improvement factor. For the approach described in section 2.2, the adaptive TB algorithm receives user-defined priority levels for each task, that is, tasks are scheduled based on priority levels and then on time balances.

The process of acquiring and/or setting parameters for surveillance and tracking tasks is described in Figure 1. First, storm tracking parameters such number of cells (N), task time (T), update time (U), and priority level (P) for each cell are provided. Also, U_C is the maximum surveillance update time that will be used in the algorithm. Second, if a T_B is not associated with surveillance, the left branch of Figure 1 shows that surveillance is fragmented and update times are adjusted as suggested by approach 1 in section 2.1. Step e explains such approach and is shown in detail in Figure 2. If requested update times are to be controlled, the update time for surveillance is estimated based on equation (1) and its value determines whether or not radar resources are overloaded. For an overload condition, the update time correction factor is estimated based on equation (3), and update times are increased as shown in equation (4). For a non-overload condition, if the surveillance update time is set to U_C, then tracking update times are decreased using equation (4). Third, when a T_B is associated with surveillance, (shown in the right branch of Figure 1), surveillance is not longer fragmented and its task time is denoted by $T_{N'}$. Also, priority level assignments are evaluated as suggested by approach 2 in section 2.2. Step h explains such approach which is shown in detail in Figure 3. If radar resources are insufficient, surveillance update time is forced to be the same as for conventional radar, U_C. In addition, if surveillance T_B is positive, its priority is set to be larger than the maximum of all P_i. Otherwise, surveillance update time and surveillance priority are not modified. In summary, step 1 is the pre-scheduling process where parameters for storm tracking and surveillance tasks are defined and possibly adjusted.

The adaptive TB scheduling algorithm flow chart is shown in Figure 4. After the parameters for each task are acquired, priority levels and time balances for all tasks are evaluated. The output of this process is indicated by three branches in Figure 4. The left branch runs if there is at least one task of storm tracking or surveillance that has a positive T_B at the current priority level, (note that tasks are analyzed in decreasing order of priority). This task is scheduled next. The center branch runs if no storm tracking task has positive T_B and surveillance fragments need to be executed until a tracking task T_B becomes positive. The right branch runs if neither storm tracking nor surveillance tasks have a positive T_B, which means that all of them are on time. In this case, the radar is idle until the largest T_B becomes zero or positive. The T_B changes with time as the algorithm provided by Reinoso-Rondinel et al. (2009) for left and center branches. This procedure is repeated until no more tasks are to be scheduled.

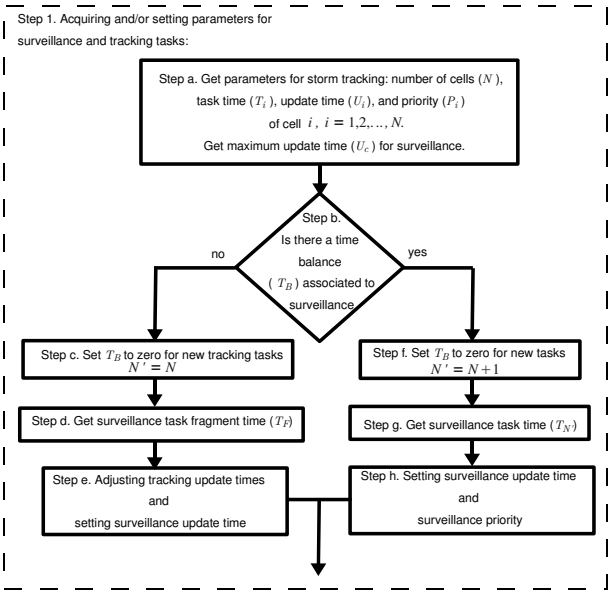

Figure 1: Step 1 of the adaptive TB scheduler algorithm. Parameters associated to storm tracking and surveillance tasks are provided or estimated. The option to associate a T_B with surveillance is shown in step b.

The general idea of the adaptive TB algorithm is to schedule each task as it was requested. Therefore, the TB algorithm schedules these competing tasks by balancing the available radar time and the requested update time of each tracking task. When radar resources are not enough, the algorithm can adaptively change requested update times or schedule tasks based on both priority levels and time balances.

4. SIMULATION RESULTS

Data observed by a WSR-88D radar is used to simulate PAR observations. In this section, a case of multicell storms observed by the KTLX radar in Twin Lakes, Oklahoma on 22 April 2008 is used to demonstrate the adaptive TB scheduling algorithm for adaptive weather sensing with the goals of (1) providing fast update time for storms without compromising data quality and (2) mitigating the overload condition. Over a period of 70 minutes, a single storm cell split into two cells at approximately 0125 UTC and later the one on the north side further split and three cells were observed by the radar. Reflectivity fields from selected times at elevation of 0.5°are shown in Figure 5. The KTLX data was linearly interpolated in time at each elevation to simulate a more frequent weather data set every 15 s. In this way, we can simulate the process of adaptive TB scheduling that would be possible with the NWRT PAR. In this work,

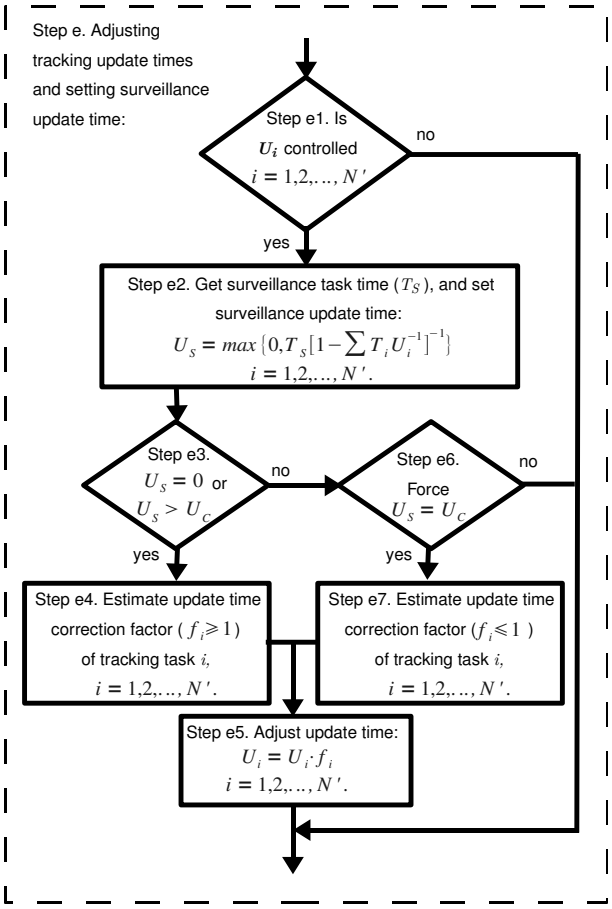

Figure 2: Step e of the adaptive TB algorithm. Tracking task update times are increased to mitigate the overload condition or reduced to increase the revisit improvement factor, if steps e1 and e3 are true or steps e1 and e6 are true, respectively.

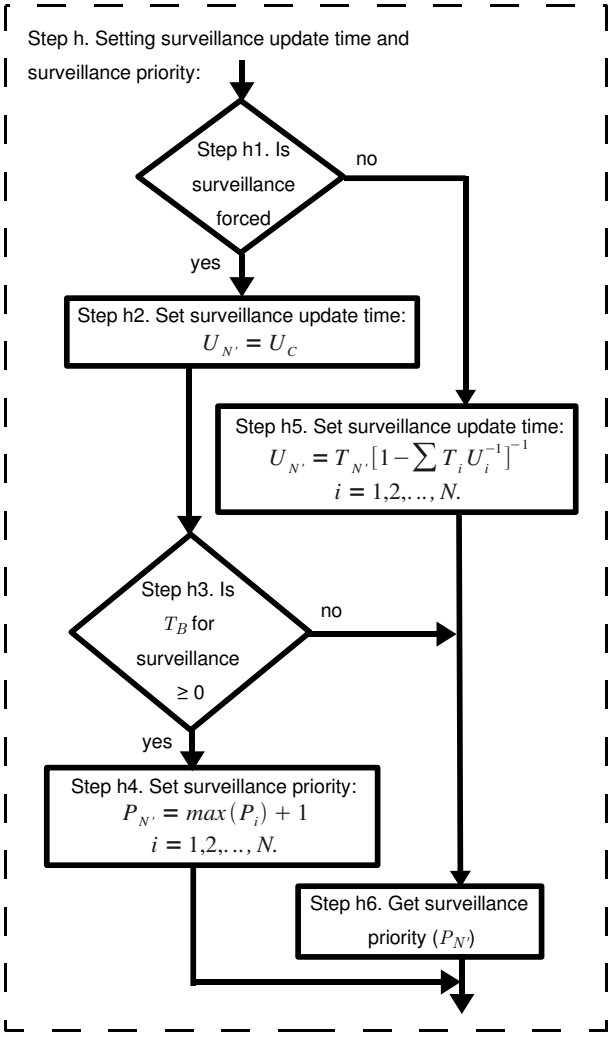

Figure 3: Step h of the adaptive TB algorithm. Surveillance update time is set the same as for conventional radar if step h1 is true. If step h3 is also true, surveillance priority is set to be larger than the maximum of all P_i.

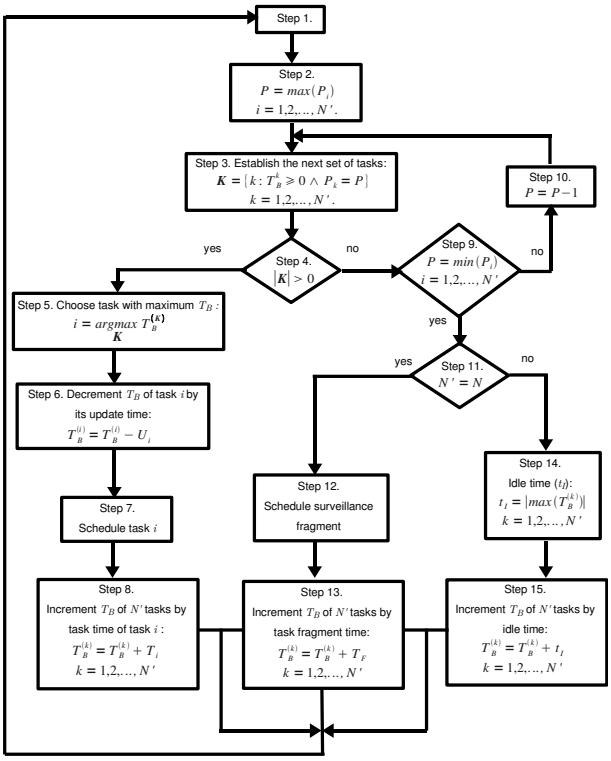

Figure 4: Adaptive Time Balance scheduler algorithm flow chart. The scheduler selects a tracking or surveillance task (left branch), or a surveillance fragment (center branch), or it remains idle (right branch) based on the evaluation of priorities and time balances. Parameters for tasks are updated on every iteration.

the storm cells are identified based on some of the concepts used in the storm cell identification and tracking algorithm (SCIT) (Johnston et al. 1998), but in a simpler manner. Figure 5 shows a 35 dBZ contour for each identified cell and their azimuthal extent limited by red, blue, and cyan lines for cells 1, 2, and 3, respectively. For

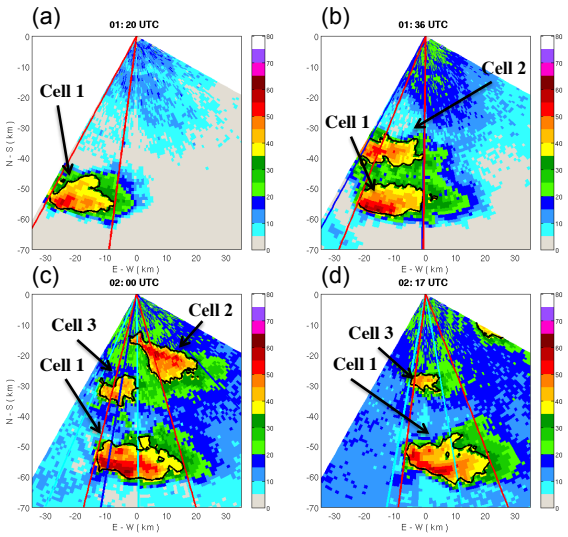

Figure 5: Reflectivity data of storm cells at 0.5°elevation angle in Central Oklahoma on 22 April 2008. (a) cell 1. (b) cells 1 and 2. (c) cells 1,2, and 3. (d) cells 1 and 3 (cell 2 left the 90°sector).

tracking tasks, the task time is the time needed to complete the volumetric scan of the identified storm using the same dwell times as in the WSR-88D VCP. Thus, the data quality is maintained for storm tracking tasks. For the surveillance task, the dwell time is fixed for all elevations at 9.2 ms. This is the same as the shortest dwell times used for detection in VCP 12 (i.e., long-PRT pulses of Batch mode). Surveillance task fragments are determined by grouping beam positions by elevation angles. Then, T_S is the sum of T_F over all the elevations in the scanning strategies. When there is a T_B associated with surveillance, $T_F = T_{N'}$; i.e., surveillance contains only one fragment. Task times for surveillance and cells 1, 2, and 3 are graphed in gray, red, blue, and cyan lines respectively in Figure 6.

In order to investigate the impact of update times on the scheduling algorithm, three different periods were selected with different update time requirements. Period 1 (0120-0140 UTC) contains cell 1 only, and then cells 1 and 2. Period 2 (0140-0215 UTC) contains cells 1, 2, and 3. Period 3 (0215-0230 UTC) contains both cells, 1 and 2, and then only cell 1. For each period, if more than one storm cell is present, update times for all tracking tasks are identical. Tracking update times are set as

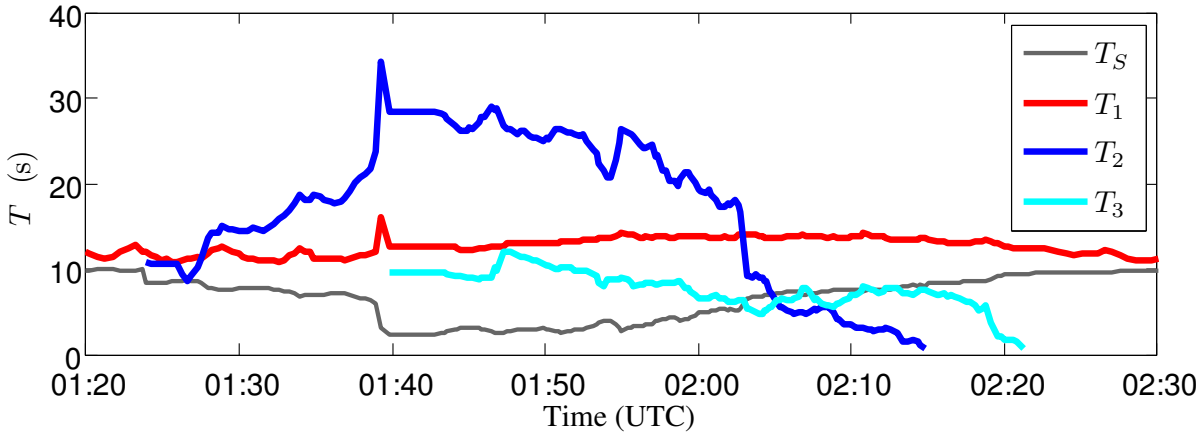

Figure 6: Task times for surveillance and cells 1, 2, and 3 are denoted by gray, red, blue, and cyan lines, respectively.

follows. For period 1, U is 30 s and after cell 2 appears, U increases to 40 s for both of them. For period 2, U increases to 45 s for all cells. For period 3, U decreases to 30 s. These requested update times may lead to an overload condition and to evaluate the adaptive TB algorithm, the original TB algorithm, and two variations are analyzed: (1) the adaptive TB algorithm with update times adjustment, and (2) the adaptive TB algorithm with task prioritization.

4.1. Original TB algorithm

The performance of the scheduling algorithm is assessed by comparing the requested and actual update times. The improvement factor is calculated using a sliding window with size given by the acquisition time (Reinoso-Rondinel et al. 2009). Figure 7 illustrates non-overload and overload conditions obtained with the original TB algorithm where task priority levels are always the same and requested update times are not adjusted. Update times, requested occupancy along with envelope of T_B, and the revisit improvement factor are shown on top, middle, and bottom panels of Figure 7. For tracking tasks, requested update times for each period are shown by the black dashed thick line on the top panel of Figure 7. The update time for the surveillance task can be estimated from equation (1) and it is plotted with a gray dashed line. In addition, the update time for a 90°sector of KTLX, U_C, is represented by the black solid line. Actual update times obtained from the original TB algorithm for cells 1, 2, and 3, are graphed in red, blue, and cyan lines respectively. For periods 1 and 3, requested and actual update times for storm tracking tasks agree well. However, for the interval of 0140 to 0155 UTC in period 2, actual update times are larger than requested. The reason for tasks being scheduled late is that the requested U of 45 s is shorter than sum of task times for surveillance and cells 1, 2, and 3 ($O_T > 100\%$). Between, 0155 and 0205 UTC actual update times become smaller than requested because storm tracking tasks are executed earlier than requested to compensate for the delays during the overload period. Note that the surveillance update time becomes extremely large during and immediately after the overload condition.

Total requested occupancy (right y-axis) and $\widetilde{T_B}$ (left y-axis) are shown in the middle panel of Figure 7. Total requested occupancy is the sum of the ratios between T and U for each task and is indicated by the black dashed thick line. As it can be seen, in the interval from 0140 to 0155 UTC of period 2, the total requested occupancy is higher than 100%; i.e., radar resources are overloaded. The envelope of T_B ($\widetilde{T_B}$), proposed by Reinoso-Rondinel et al. (2009), is the set of the values of T_B every time a task is executed. $\widetilde{T_B}$ can be used to observe whether a task is executed on time. $\widetilde{T_B}$ for cells 1, 2, and 3 are indicated by red, blue, and cyan lines, respectively. The overload condition is reflected on the $\widetilde{T_B}$ for tracking tasks. For the time interval from 0140 to 0155 UTC, every $\widetilde{T_B}$ increased with time. During this interval, only tracking tasks were executed; i.e, surveillance was critically delayed. Between 0155 and 0205 UTC, each $\widetilde{T_B}$ decreased rapidly because tracking tasks were scheduled in time intervals smaller than the requested U, this is referred to as the "catch up" period. After 0205 UTC, $\widetilde{T_B}$ behaves normally as expected for a non-overload condition.

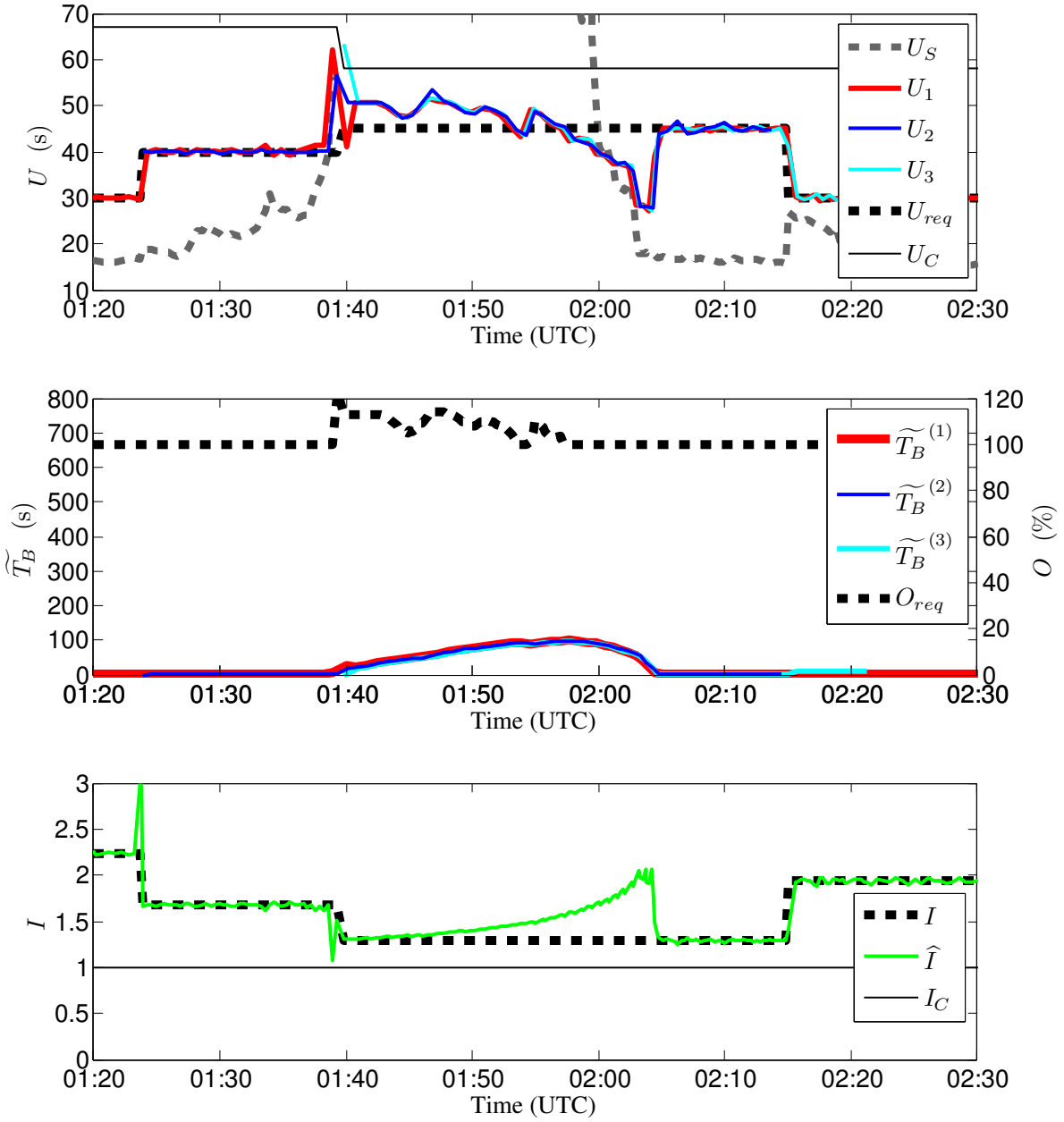

Figure 7: Performance of the original TB algorithm. (Top) Update times for surveillance and each storm tracking task. Thick, dashed black, and gray lines are the requested update time for each cell and surveillance, respectively. Actual update times for cell 1, 2, and 3, are represented by red, blue, and cyan lines. The update time for WSR-88D, U_C, is indicated by a black solid line. (Middle) Envelop of time balance, $\widetilde{T_B}$ on the left y-axis, for cells 1, 2, and 3, indicated by red, blue, and cyan lines. The total requested occupancy, O on the right y-axis, represented by black dashed thick line. (Bottom) Total revisit improvement factor. Theoretical, estimated, and conventional revisit improvement factors are indicated by dashed black, green, and solid black lines, respectively.

Finally, the revisit improvement factor can be used to quantify how frequently storms are scanned compared to the WSR-88D. Both the theoretical and estimated revisit improvement factors are presented in the bottom panel of Figure 7. The theoretical (I) and estimated ($\widehat{I}$) revisit improvement factors are denoted by the black dashed thick and green lines, respectively. In addition, $I_C = 1$ means no gain on the number of revisits from adaptive sensing over conventional scanning (denoted by the black solid line). The theoretical and estimated improvement factors are consistent except when overloading occurs. When the overload condition begins, $\widehat{I}$ shows a constant increment compared to I. This is due to the early execution of tracking tasks. Although, that occurs later on; these events are included in and bias the estimation of I. It is important to note that during the entire simulation, each storm is revisited more frequently compared to the KTLX standard scan, while the same data accuracy is maintained.

4.2. Approach 1: Adjusting task update times

The 1^{st} variation of the adaptive TB scheduling algorithm is presented. Here, task priority levels are always the same and the update times requested by users are adjusted as suggested in section 2.1. To simplify the discussion, approach 1 is treated as two cases: 1a and 1b. In approach 1a, requested update times can be incremented to mitigate the overload condition. In approach 1b, requested update times can be decremented to increase the revisit improvement factor. Figures 8 and 9 illustrate non-overload and overload conditions obtained with approaches 1a and 1b, respectively. Update times, requested occupancy along with envelope T_B, and the revisit improvement factor are shown on top, middle, and bottom panels of Figures 8 and 9 for approaches 1a and 1b, respectively. Both requested and actual tracking update times, surveillance update time, and U_C are plotted using the same line styles as in Figure 7. These can be seen on top panels of Figures 8 and 9 for approaches 1a and 1b, respectively. For approach 1a, requested and actual update time results for periods 1 and 3 are very similar with those results obtained from the original TB algorithm; however, this is not the case for period 2. The original update time defined by the user (U_{user}) is indicated by a dashed black thin line on the top panel of Figure 8 (same for all tracking tasks). In this period, U_{user} is increased so that surveillance update time is U_C and there is not overloading. As a consequence, actual and requested update times agree well while surveillance is also executed during period 2. Note that tracking tasks are executed on time, but actual update times are not smaller than requested. On the other hand, for approach 1b, requested and actual update time results for period 2 are very similar with those results obtained from approach 1a. However, results from approach 1b and the original TB algorithm are different for periods 1 and 3. In these periods, U_{user} is decreased so that surveillance update time is no less than U_C. Note that the total requested occupancy is not changed but redistributed to increase the revisit improvement factor of storm tracking tasks.

The total requested occupancy and $\widetilde{T_B}$ are plotted using the same line styles as in Figure 7. These can be seen on middle panels of Figures 8 and 9 for approaches 1a and 1b, respectively. For both approaches, the first difference from the original TB algorithm in period 2, is that, the total requested occupancy shows a non-overloaded condition because U_{user} was increased. Second difference is that $\widetilde{T_B}$ for each tracking task is flat; i.e., storm tracking tasks were executed on time.

Finally, theoretical, estimated, and conventional revisit improvement factors are plotted using the same line styles as in Figure 7. These can be seen on bottom panels of Figures 8 and 9 for approaches 1a and 1b, respectively. For approach 1a, since actual and requested update times agree, theoretical and estimated revisit improvement factors agree as well during the three periods, in contrast with the original TB algorithm. For approach 1b, theoretical and estimated revisit improvement factors agree well because no execution delays on tracking tasks were shown. This makes a difference from the original TB algorithm since U_{user} was decremented based on equation (4) and then placed in equation (5). Thus, the theoretical revisit improvement factor of approach 1b is larger than the original TB algorithm in periods 1 and 3. Note that the estimated revisit improvement factor from approaches 1a and 1b are very close to 1 in the interval 0140-0155 UTC of period 2, which means no much gain on the number of storm revisits was achieved.

4.3. Approach 2: Task prioritization

The 2^{nd} variation of the adaptive TB scheduling algorithm is presented next. Here, task priority levels are different but the requested update times are not adjusted as it was done in the previous variation (section 2.2). As mentioned in section 2, it is very important that the time between surveillance executions is not long. In fact, a surveillance update time no larger than U_C is desired. However, if $O_T > 100\%$, radar resources would not be enough to execute tasks as requested. Thus, an assigned priority level to each task and a T_B associated with surveillance task are proposed. For the surveillance

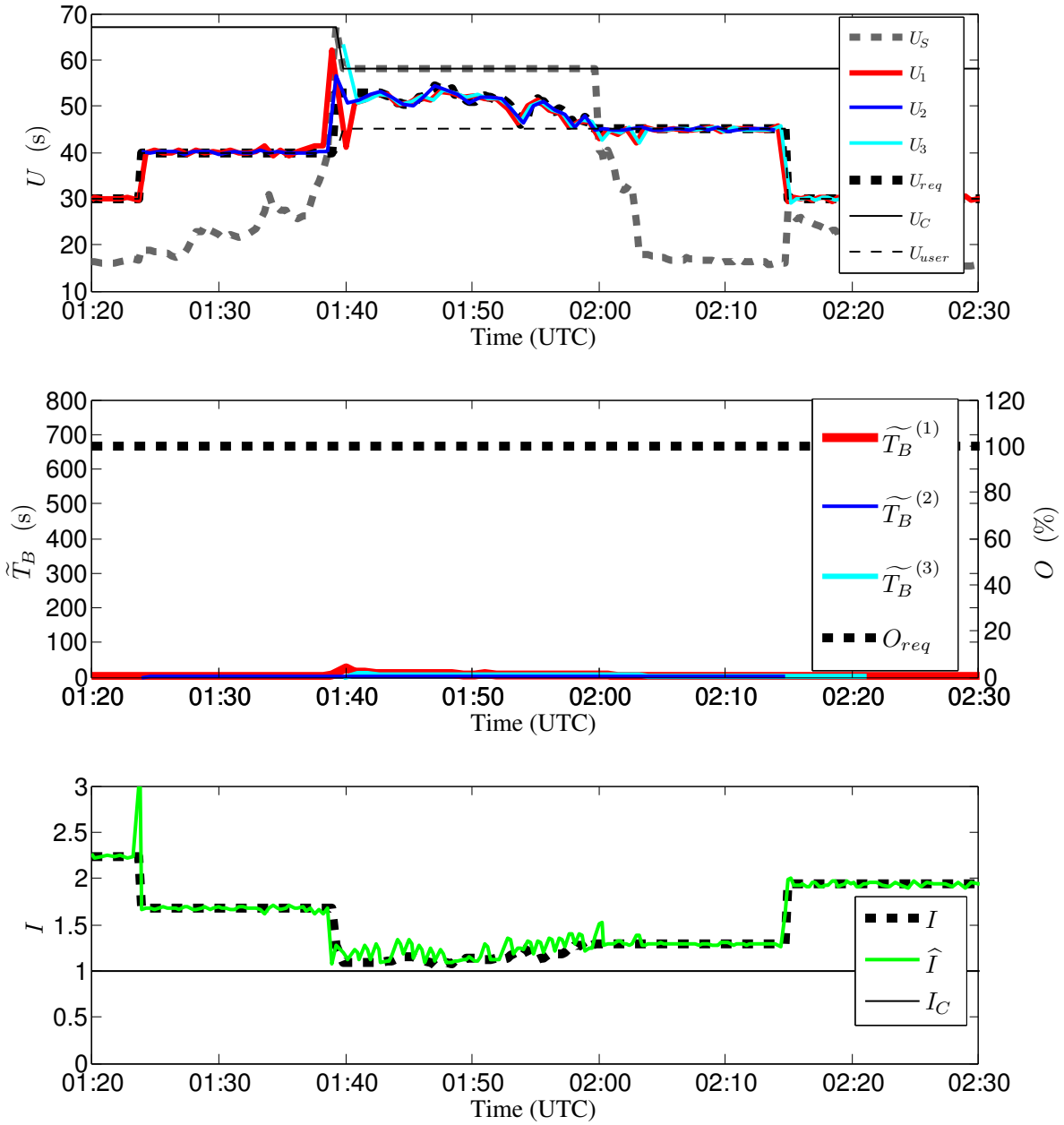

Figure 8: Performance of approach 1a. (Top) Update times for surveillance and each storm tracking task. Thick, dashed black, and gray lines are the requested update time for each cell and surveillance, respectively. Actual update times for cell 1, 2, and 3, are represented by red, blue, and cyan lines. The update time for WSR-88D, U_C, and the user update time, U_{user}, are indicated by black solid and dashed thin lines, respectively. (Middle) Envelop of time balance, $\widetilde{T_B}$ on the left y-axis, for cells 1, 2, and 3, indicated by red, blue, and cyan lines. The total requested occupancy, O on the right y-axis, represented by black dashed thick line. (Bottom) Total revisit improvement factor. Theoretical, estimated, and conventional revisit improvement factors are indicated by dashed black, green, and solid black lines, respectively.

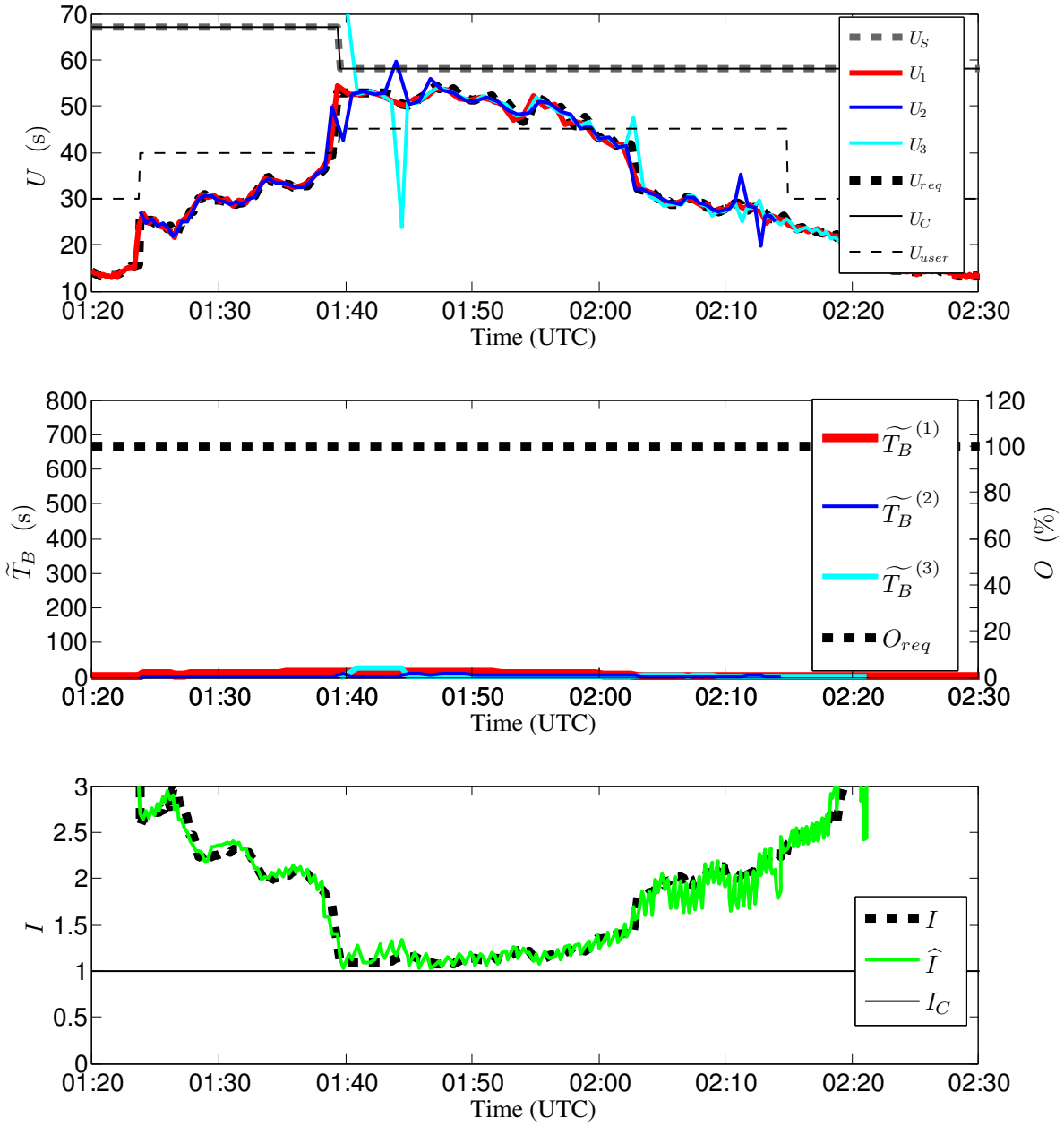

Figure 9: Performance of approach 1b. (Top) Update times for surveillance and each storm tracking task. Thick, dashed black, and gray lines are the requested update time for each cell and surveillance, respectively. Actual update times for cell 1, 2, and 3, are represented by red, blue, and cyan lines. The update time for WSR-88D, U_C, and the user update time, U_{user}, are indicated by black solid and dashed thin lines, respectively. (Middle) Envelop of time balance, $\widetilde{T_B}$ on the left y-axis, for cells 1, 2, and 3, indicated by red, blue, and cyan lines. The total requested occupancy, O on the right y-axis, represented by black dashed thick line. (Bottom) Total revisit improvement factor. Theoretical, estimated, and conventional revisit improvement factors are indicated by dashed black, green, and solid black lines, respectively.

task, the priority level is initially set to 1 for all periods. Priorities of tracking tasks are assigned as follows. For period 1, the priority of cell 1 is P_1 = 2. When cell 2 appears, P_1 = 3 and P_2 = 2 for cell 1 and 2, respectively. For period 2, P_1 = 4, P_2 = 3, and P_3 = 2 for cells 1, 2, and 3, respectively. For period 3, P_1 = 3 and P_2 = 2 for cells 1 and 3, respectively. When cell 2 leaves the 90°sector, P_1 = 2 for cell 1. In this version of the algorithm, there is a T_B associated with surveillance and the surveillance task is no longer fragmented. Further, its update time is estimated from equation (1); however, if it is larger than U_C, it is forced to U_C. Here, the adaptive TB algorithm chooses the next task based on priority levels and then time balances.

Figure 10 illustrates non-overload and overload conditions with the adaptive TB scheduling when task prioritization is applied. Update times, requested occupancy along with envelope T_B, and the revisit improvement factor are shown on top, middle, and bottom panels of Figure 10. Both requested and actual tracking update times, surveillance update time, and U_C are plotted using the same line styles as in Figure 7. In addition, actual update time for surveillance is plotted by the gray solid line. These graphs can be seen on the top panel of Figure 10. Results are divided in two for discussion. (1) For periods 1 and 3, two reasons explain the oscillatory behavior of actual update times around the requested ones. First, the priority level for surveillance is the minimum of all P_i, i.e., late surveillance executions are caused by executions of tasks with higher priorities (cells 1 and 2). Second, the requested update time for storm tracking tasks is larger than for the surveillance task. As a consequence, late executions for cells 1 and 2 occur since faster updates are required by surveillance. Because the idea of the adaptive TB algorithm is to balance the time between every executed task, a late task may be forced to be executed resulting in smaller actual update times, which explains the oscillations. (2) For period 2, radar resources are overloaded and according to section 2.2 some tasks may be executed on time while others do not. Requested and actual update times for surveillance and cells 1 and 2 are in fair agreement. This is achieved at the expense of a significant delay of cell 3. The reason for this is that, relative high priorities were associated with cells 1 and 2 compared to the priority of cell 3. Also, the update time for surveillance was set to U_C and its execution was forced by maximizing its priority level when this task was due. As a consequence, there was no time to execute cell 3. However, when the total requested occupancy decreases to 100%, actual update times for cell 3 become smaller than requested; i.e., its number of executions increases because its time balance leads to a“catch up” period. The main difference from the original TB algorithm during the overload condition is that all tasks except cell 3 were executed on time, while in the original TB algorithm, all tracking tasks were executed late while the surveillance task was not executed at all.

The total requested occupancy and $\widetilde{T_B}$ are plotted using the same line styles as in Figure 7. In addition, $\widetilde{T_B}$ for surveillance is plotted by the gray solid line and these graphs can be seen on the middle panel of Figure 10. In period 2, the total requested occupancy reflects the overload condition. For all three periods, $\widetilde{T_B}$ for surveillance and cells 1 and 2 indicate that resources requested by these can be provided by the radar. However, the increase of $\widetilde{T_B}$ for cell 3 in period 2 means that it was not executed on time. Thus, the main difference with respect to the original TB algorithm in period 2 is that executions for surveillance and cells 1 and 2 are balanced, while in the original TB algorithm, the execution of tasks are unbalanced. Theoretical, estimated, and conventional revisit improvement factors are plotted on the bottom panel of Figure 10 using the same line styles as in Figure 7. Estimated revisit improvement factors, obtained from approach 2 and the original TB algorithm, show similar results for all periods. The estimated revisit improvement factor is biased as described before but note that in the first part of period 2, the estimated is now smaller than the theoretical. This is caused by the significantly delayed execution of cell 3.

Concisely, two approaches were presented to handle the overload condition. In approach 1, tasks were considered equally important and the total requested occupancy was decreased to achieve a non-overload condition by increasing requested tracking update times. Thus, tracking and surveillance tasks are executed on time. On the other hand, approach 2 did not increase requested update times, that is, the overload condition remains. Instead, priority levels were assigned to storm tracking and surveillance tasks. As a consequence, the adaptive TB algorithm chooses the next task based on priority levels and then time balances; i.e., important tasks are executed on time while low priority tasks are not.

5. SUMMARY AND CONCLUSIONS

Conventional radars usually need to trade between update times and data quality. PAR are suitable to perform radar functions in an adaptive sensing manner and possibly overcome such trade-off. However, overload conditions may occur when there are no sufficient radar resources as requested. Therefore, an extended algorithm was presented to mitigate such condition. Two dif-

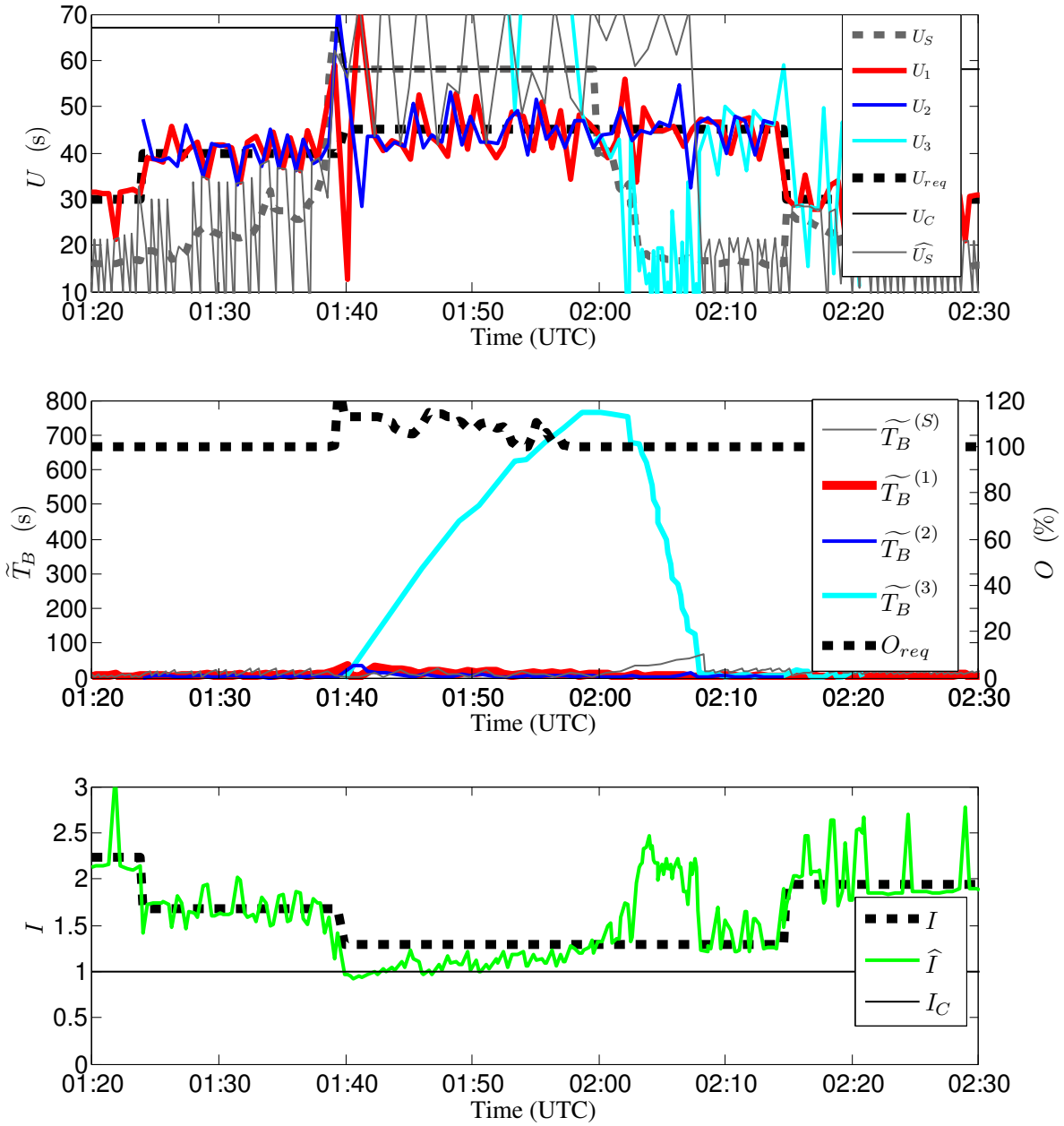

Figure 10: Performance of approach 2. (Top) Update times for surveillance and each storm tracking task. Thick, dashed black, and gray lines are the requested update time for each cell and surveillance, respectively. Actual update times for cell 1, 2, and 3, are represented by red, blue, and cyan lines. The update time for WSR-88D, U_C, and the actual surveillance update time, $\widehat{U_S}$, are indicated by black solid and gray thin lines, respectively. (Middle) Envelop of time balance, $\widetilde{T_B}$ on the left y-axis, for surveillance and cells 1, 2, and 3, indicated by gray, red, blue, and cyan lines. The total requested occupancy, O on the right y-axis, represented by black dashed thick line. (Bottom) Total revisit improvement factor. Theoretical, estimated, and conventional revisit improvement factors are indicated by dashed black, green, and solid black lines, respectively.

ferent approaches were shown to mitigate the reduced performance of the original TB scheduling algorithm during overload conditions.

Reflectivity data taken from KTLX was used to test the adaptive TB scheduling algorithm for the execution of three storm tracking tasks on the NWRT PAR. An overload condition caused by these tasks was mitigated by two approaches. (1) adjustment of task update times and (2) task prioritization. The first approach assumes that requested storm tracking update times can be relaxed so the radar can provide enough resources. The second approach tries to satisfy storm tracking update times as originally requested. Both approaches handle the overload condition but users may determine which one best satisfies their needs.

In summary, an adaptive TB scheduling algorithm like the one introduced here should be a vital component in a multifunction PAR system. As illustrated in this work, such algorithm should be designed to adjust requested task parameters to better organize radar resources for weather sensing. It is foreseen that in such adaptive weather sensing context, users of weather products will benefit from better data interpretation leading to significant improvements on weather warnings and forecasts.

ACKNOWLEDGEMENT

This work was primarily supported by NOAA/NSSL under Cooperative Agreement NA17RJ1227. Part of this work was supported by DOD, EPSCoR Grant N00014-06-1-0590.

References

Capraro, G. T., A. Farina, H. Griffiths, and M. C. Wicks, 2006: Knowledge-based radar signal and data processing: A tutorial review. *IEEE Sig. Proc. Mag.*, **23**, 18–29.

Forsyth, D. E., J. F. Kimpel, D. S. Zrnic, R. Ferek, J. F. Heimmer, T. McNellis, J. E. Crain, A. M. Shapiro, R. J. Vogt, and W. Benner, 2005: Progress report on the National Weather Radar Testbed (phased-array) becomes operational. *21st Int. Conf. on Interactive Information and Processing Systems (IIPS) for Meteorology, Oceanography, and Hydrology*, San Diego, CA, Amer. Meteor. Soc., CD-ROM, 19.5.

Gini, F. and M. Rangaswamy, 2008: *Knowledge-Based Radar Detection, Tracking, and Classification*. John Wiley and Sons, 265 pp.

Haykin, S., 2006: Cognitive radar: A way of the future. *IEEE Sig. Proc. Mag.*, **23**, 30–40.

Heinselman, P. L., D. L. Priegnitz, K. L. Manross, T. M. Smith, and R. W. Adams, 2008: Rapid sampling of severe storms by the National Weather Radar Testbed phased-array radar. *Wea. Forecasting*, **23**, 808–824.

Johnston, J. T., P. L. MacKeen, A. Witt, E. D. Mitchell, G. J. Stumpf, M. . D. Eilts, and K. W. Thomas, 1998: The storm cell identification and tracking algorithm: An enhanced WSR-88D algorithm. *Wea. Forecasting*, **13**, 263–276.

Manners, D. M., 1990: ART an adaptive radar testbed. *IEE Colloquium on Real-Time Management of Adaptive Radar Systems*, London, UK, Siemens Plessey Radar, 1-7.

Miranda, S., C. J. Baker, K. Woodbridge, and H. D. Griffiths, 2006: Knowledge-based resource management for multifunction radar: A look at scheduling and task prioritization. *IEEE Sig. Proc. Mag.*, **23**, 66–76.

Polger, P. D., B. S. Goldsmith, and R. C. Bocchierri, 1994: National Weather Service warning performance based on the WSR-88D. *Bull. Amer. Meteor. Soc.*, **75**, 203–214.

Reinoso-Rondinel, R., T.-Y. Yu, and S. Torres, 2009: Multifunction phased-array radar: Time balance scheduler for adaptive weather sensing. *J. Atmos. Oceanic Technol.*, **[submitted]**.

ROC, 2007: WSR-88D system specification. WSR-88D Radar Operations Center Rep. OWY55, 164 pp. [Available from NOAA FOIA Office, Public Reference Facility (OFA56), 1315 East West Hwy. (SSMC3), Room 10730, Silver Spring, MD 20910].

Skolnik, M. I., 2001: *An Introduction to Radar Systems*. McGraw-Hill, 772 pp.

Stafford, W. K., 1990: Real time control of multifunction electronically scanned adaptive radar (MESAR). *IEE Colloquium on Real Time Management of Adaptive Radar Systems*, London, UK, Siemens Plessey Radar, 1-5.

Steadham, R. M., R. A. Brown, and V. T. Wood, 2002: Prospects for faster and denser WSR-88D scanning strategies. *18th Int. Conf. on Interactive Information and Processing Systems (IIPS) for Meteorology, Oceanography, and Hydrology*, Orlando, FL, Amer. Meteor. Soc., J3.16.

Vannicola, V. C., L. K. Slaski, and G. J. Genello, 1993: Knowledge-based resource allocation for multifunction radars. *SPIE Signal and Data Processing of Small Targets*, **1954**, 410–425.

Zrnić, D. S., J. F. Kimpel, D. E. Forsyth, A. Shapiro, G. Crain, R. Ferek, J. Heimmer, W. Benner, T. J. McNellis, and R. J. Vogt, 2007: Agile beam phased-array radar for weather observations. *Bull. Amer. Meteor. Soc.*, **88**, 1753–1766.