An Adaptive Update-Rate Control of a Phased Array Radar for Efficient Usage of Tracking Tasks

Sang Hoon Baek, Hyunchul Seok, Kyu Ho Park and Joohwan Chun
{baeksangh, hcseok, kpark}@core.kaist.ac.kr, chun@ee.kaist.ac.kr
KAIST, Daejeon, Korea

Abstract— **In multi-functional radar, task scheduling algorithm should be designed such that timing resource is efficiently utilized by functions such as surveillance and tracking, and its performance is maximized. In the target tracking, the tasks are required to be executed to consider the maneuvering motion, measurement condition and required tracking performance. Frequent execution of tracking tasks results in not only precise tracking, but also waste of timing resource which is shared with other functions. Therefore, to reduce the number of unnecessary observations, the tracking task is required to be executed only when the update is needed. In this paper, the innovation, position residual, in Kalman filter is used as reference value for adjusting update rate of tracking tasks. Using feedback controller, the update rate is allocated so that predicted observation is expected to be within specified error range. In addition, targets are classified into 7 priorities according to tactical characteristics, and target's priority is also used as reference value for calculating update rate. The simulation results show that the proposed method reduces the tracking error of the target on maneuvering movement compared to fixed update rate case.**

I. Introduction

Phased array antennas instantaneously direct a beam into a desired direction by electronically controlling the relative phases of the antenna array. By controlling a sequence of beam direction, single radar can perform several functions such as surveillance, tracking and weapon guidance. In target tracking, the radar chases the target movement by observing the target intermittently, determining the detection state through signal processing, and updating the track state from detection. As tracking tasks are executed more frequently and dwell time to the target is longer, the track quality tends to be improved due to more updates. However, since the radar time is a shared resource among several functions such as surveillance and weapon guidance, frequent execution of tracking tasks takes away the chances for the other task executions. Therefore, in the task scheduling, the goal is to minimize the time resource by utilizing tracking beam efficiently and to maximize the overall track quality with as fewer beams as possible.

This could be regarded as a problem of finding optimal update time to each target to fit with its mobility and measuring condition. For example, if the movement of a target object is on less predictable condition like maneuvering movement, or the measurement error is heavy, the update rate will be increased to prevent the track quality degradation. On the other hand, if target movement is predictable like quiescent motion, and also the measured data is reliable, the update rate will be decreased to potentially yield the observation time to other task execution with the cost of ignorable track quality degradation.

For efficient management of PAR, maintaining the prediction error within specified range is a reasonable solution to reduce the number of unnecessary observations. Previous approaches proposed a solution for them by selecting large update rate to the target with high target acceleration [3]. Other methods are proposed based on IMM (Interacting Multiple Models) that next update time is selected such that predicted error covariance in position is kept under the threshold value [2]. This approach is based on degree of adaptability of tracking model to the target motion, and the value is not affected by prediction error.

This paper describes a method of adjusting the update rates based on the innovation which is one of the variables in Kalman filter. Using feedback controller, the update rate is allocated so that predicted target observation expected to be within specified ellipsoid implying observation error range. In addition, we categorize targets into 7 priorities. When calculating the update rate, we use different base error boundaries according to target's priority. This adaptive approach using feedback controller which compensates the prediction error is suitable for update rate allocation since the target movement characteristic varies with time and the relation between the track update rate and its track quality is known as monotonically increasing relation, but their relation is not clearly defined.

In the simulation results, the update rate variation and the position error are shown and tracking performance is compared between adaptive update rate scheme and fixed update rate scheme.

II. Related Works

An adaptive update rate algorithm is an extension of traditional trackers with uniform update rates, and it is needed for efficient usage of tracking beam. Two typical types of variable update algorithms are proposed before. One is the variable update algorithm based on the predicted error covariance and the other is based on the innovation (position residual).

As the first approach, Watson proposed an algorithm to select update interval which makes predicted error covariance exceed a given threshold [1]. The predicted error covariance can reflect the uncertainty of a target, thus it can be used to control the update rate. The next measurement is scheduled when the predicted error covariance in position exceeds a given

978-1-4244-5812-7/10/$26.00 ©2010 IEEE

threshold, which can be expressed as follow

$$\overline{\mathrm{P}}_{k+T|k} \leq \overline{\mathrm{P}}_{th} \tag{1}$$

, where $\overline{\mathrm{P}}_{k+T|k}$ denotes the predicted error covariance, $\overline{\mathrm{P}}_{th}$ is the threshold, and T is the time between consecutive target updates. When calculating T, the non-diagonal elements are ignored, thus

$$Tr[\overline{\mathrm{P}}_{k+T|k}] \leq Tr[\overline{\mathrm{P}}_{th}] \tag{2}$$

, where Tr is the sum of its diagonal entries.

Obviously, T can get its maximum when the equal mark holds. The threshold is usually selected to relative to the measurement error covariance

$$Tr[\overline{\mathrm{P}}_{k+T|k}] = \lambda Tr[\mathrm{R}_k] \tag{3}$$

, where R_k is the measurement error covariance, and λ is a positive scalar which can be controlled to balance the tracking precision and the system load.

The second type of variable update algorithm uses the position residual. The increasing residual of filter means the tracking precision is decreasing currently. Therefore, relative small update interval is required to keep the tracking performance. Based on this, Cohen got a formula of the update interval in [4]

$$T(k+1) = \frac{T(k)}{\sqrt{e_0(k)}} \tag{4}$$

, where

$$e_0(k) = |e(k)|/\sigma_k \tag{5}$$

, σ_k is the standard deviation of the measurement noise and $e(k)$ is the position residual. In addition, Cohen suggested smoothing the residual using a first order filter to get $e(k)$

$$e_s(k) = \alpha_r e(k) + (1-\alpha_r)e_s(k-1) \tag{6}$$

, where α_r is the smoothing constant and $e_s(k)$ then replace $e(k)$ in equation (5).

In [7], Coetzee proposed an adaptive update time calculation algorithm. The standard deviation s of the measurement noise is calculated over the entire trajectory. The residual error is normalized to remove the effect of the measurement noise

$$e = \frac{|y_n - x_n|}{s} \tag{7}$$

, where y_n is a measurement position at time n and x_n is a predicted position at time n. The update time is recalculated using the 'cube-root filter' as

$$t_{n+1} = \frac{t_n}{\sqrt[3]{e/\lambda}} \tag{8}$$

, where $\lambda = 2.25$ according to evaluations by [6]. The update time is rounded to a factor of the sampling interval T, where in this case $T = 1sec$.

$$t_{n+1} = round(\frac{t_{n+1}}{T}). \tag{9}$$

To limit the update time to an acceptable interval, t is restricted to design-specified minimum and maximum values. In this case, $1 \leq t_{n+1} \leq 4sec$.

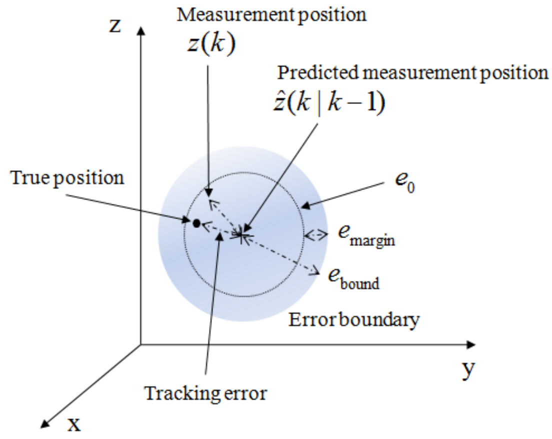


Fig. 1. Tracking error and error boundary

The adaptive update rate algorithms based on the predicted error covariance and the position residual are previously introduced in many papers [1][4][5][6][7][8]. But those approaches have fixed approaching value, thus are not capable of adjusting its update interval to fit with target characteristics and corresponding error level.

III. Algorithm for Adaptive Update Rate

In this section, we describe a method for determining the update interval for the next observation such that the prediction error is expected to be kept under the specified threshold as shown in Fig. 1. The update interval is controlled according to the position error of the target. In Fig 1, the position error is the difference between measurement position, $z(k)$ and predicted measurement position, $\hat{z}(k|k-1)$, and we control the position error resides in an error boundary, e_{bound}. For a maneuvering target, the measurement error increases so that the update interval should be small to prevent tracking losses. For a target with stable motion, the sampling interval is increased to reduce the number of observations.

First, we model the relation between the time interval and tracking error from unpredictable target movement. When the tracking target is on maneuvering movement, the target has acceleration in accordance with its maneuvering movement. In this case, the acceleration is considered as track error with constant velocity track model where the tracker considers the current target position and velocity. When the acceleration at time k results in the certain tracking error, the relation between them is represented as

$$\frac{1}{2}a_k T_k^2 = e_k \tag{10}$$

, where a_k is the acceleration at time k, T_k is the update interval at time k, and e_k is the position error between predicted position and measurement position.

Similarly, estimate of track error at next update is represented as

$$\frac{1}{2}a_{k+1}T_{k+1}^2 = e_{k+1}. \tag{11}$$

Our objective is to obtain the next time interval which makes e_{k+1} approach to a specified position error e_0, which is smaller than e_{bound} in Fig. 1. To obtain T_{k+1}, let

$$\frac{\frac{1}{2}a_{k+1}T_{k+1}^2}{\frac{1}{2}a_kT_k^2} = \frac{e_0}{e_k} \tag{12}$$

and assuming that the target maneuvering is not varied much between two observations, so let

$$a_k \approx a_{k+1}. \tag{13}$$

Solving the equation (12), we can obtain

$$T_{k+1} \approx T_k\sqrt{\frac{e_0}{e_k}}. \tag{14}$$

According to (14), the next update interval which is proportional to the inverse square root of the position error is allocated.

Second, the next step is to find approaching position error e_0. From the equation (14), the update interval is allocated in order that the position error is approaching to e_0, which should be selected such that the track error is expected not to exceed a threshold position error e_{bound}. To limit the track error e_0 has the smaller value than e_{bound}, we set the space between e_0 and e_{bound} denoted as e_{margin}.

The magnitude of e_{margin} can be set considering the measurement error. For instance, as the measurement error is low, the true position can be located closely to the measurement position and estimated position with high probability. On the other hand, under the condition where measurement error is high, the position error will be large. So, e_{margin} is selected proportional to the normalized standard deviation of current measurement error.

We define

$$\sigma_{norm} = \sqrt{\sigma_x^2 + \sigma_y^2 + \sigma_z^2} \tag{15}$$

, where σ_x, σ_y, σ_z is standard deviation of measurement error of $x,\ y,\ z$, respectively.

Let R_k^c, measurement error covariance matrix in Cartesian coordinate, as

$$R_k^c = \begin{bmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_z^2 \end{bmatrix} \tag{16}$$

and R_p^c, measurement error covariance matrix in radar coordinate, as

$$R_p^c = \begin{bmatrix} \sigma_r^2\sigma_\phi^2 & 0 & 0 \\ 0 & \sigma_r^2 & 0 \\ 0 & 0 & \sigma_r^2\sigma_\theta^2 \end{bmatrix} \tag{17}$$

, then the relation between both measurement error covariance is

$$R_k^c = DR_p^cD^T \tag{18}$$

, where D is the translation matrix from radar coordinates to Cartesian coordinates described as [9]

$$D = \begin{bmatrix} \cos\phi_k & \sin\phi_k\cos\theta_k & -\sin\phi_k\sin\theta_k \\ -\sin\phi_k & \cos\phi_k\cos\theta_k & -\cos\phi_k\sin\theta_k \\ \sin\theta_k & 0 & \cos\theta_k \end{bmatrix} \tag{19}$$

TABLE I

THE TRACKING PERFORMANCE WITH RESPECT TO WEIGHT C

Weight c	0.1	0.2	0.3	0.4	0.5
The number of beams	52	57	68	75	74
Mean square error (m)	44.75	43.35	43.13	38.24	38.57
Satisfied interval (%)	91.89	92.93	95.06	95.86	95.66

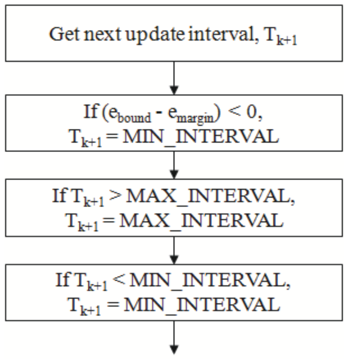


Fig. 2. Update interval limitation

.

If we substitute σ_x, σ_y, σ_z into σ_r, σ_θ, σ_ϕ in equation (15), we can get

$$\sigma_{norm} = \sqrt{\sigma_x^2 + \sigma_y^2 + \sigma_z^2} \approx \sqrt{\sigma_r^2 + \sigma_r^2\sigma_\theta^2 + \sigma_r^2\sigma_\phi^2}. \tag{20}$$

Then the relation between σ_{norm} and e_{margin} should be defined. In this paper, we assume that the their relation is on linear for simplicity instead of establishing the exact relation between them and leaving it as a further work.

Then,

$$e_{margin} = c\sigma_{norm} \tag{21}$$

, where c is weight for adjusting the distance of e_{margin}. Large c can limit the tracking error under the threshold value, but it leads to lots of update. Thus c is needed to be chosen considering allowable tracking error over the threshold value and the number of beams to use.

In this paper, c is determined experimentally. The simulated results is shown in Table. I under 3-dimensional turning movement scenario. The reasonable choice of c is 0.4 which makes both the number of beams and prediction error having compromising value. If $(e_{bound} - e_{margin}) < 0$, next time interval is replaced by the pre-defined minimum time interval.

Update interval allocated in accordance with (14) should be checked in order to prevent tracking quality degradation. First, the T_{k+1} is guaranteed maximum update interval to cope with sudden maneuvering of targets and is also guaranteed minimum update interval to radar time waste by preventing that a radar is dedicated to tracking of the target. Thus, if the time interval T_{k+1} is smaller than minimum threshold by (14), pre-defined minimum value is allocated. On the other hand, T_{k+1} is larger than maximum threshold, pre-defined maximum value is allocated as described in Fig. 2.

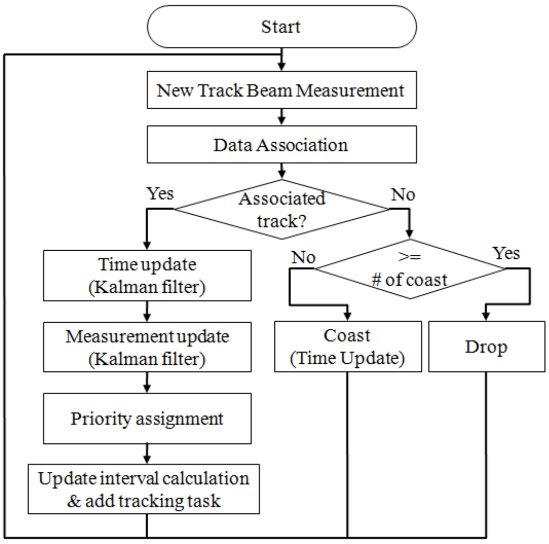


Fig. 3. Tracking sequence

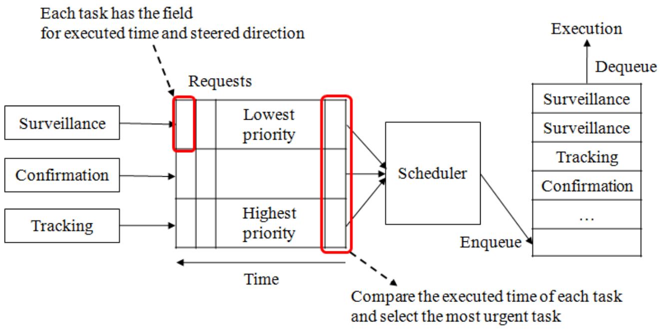


Fig. 4. Multi-level queue model for scheduler

Since targets with different priority level have different miss penalty, they have different level of required tracking performance. Therefore, the potentially threatening targets should be tracked with least miss rate by allocating high update rate. In tracking of high priority targets, if there exists conflict with another task execution, execution of the lower priority task should be delayed for higher priority task execution if necessary.

In our approach, the targets are evaluated and classified into 7 levels, and the allowable tracking error is selected depending on the priority level. According to the target priority level n, $e_{bound,n}$ is selected. Since the target importance is varied with time considering its characteristics, whenever the target update is occurred, target priority evaluation process is executed, the error boundary $e_{bound,n}$ is selected, and new time interval reflecting target priority is obtained.

IV. Implementation

The role of the tracker is to monitor consecutive updates from the radar system and to determine whether those sequences of detections belong to the same targets, while rejecting detections which are regarded as false alarms. In addition, the tracker can use the sequence of detections to estimate the current speed and direction of the target. The overall sequence of tracking is shown in Fig. 3. We implemented the tracker by using extended Kalman filter and included the function of beam scheduling in the tracker.

The scheduler part consists of task allocation part and priority allocation part. Task allocation is for scheduling radar functions such as surveillance, confirmation, and tracking based on the task priority. Priority allocation is for classifying targets and assigning suitable priority to each target based on target characteristics for tracking tasks.

In task allocation, the scheduler allocates a sequence of tasks to execute. Three types of functions are considered, which are tracking, confirmation, and surveillance task. Each radar task priority is determined according to how each task needs regularity. For instance, tracking beam is directed periodically to update targets at the given time, and surveillance beam needs less regularity if satisfying the condition that surveillance tasks are executed more than enough a certain rate. As shown in Fig. 4, tracking task has the highest priority in our implementation.

In our scheduler model, the tasks are released based on the multi-level queue model. Each element in the queue has information about release time, beam direction, and target location information. Each queue in the multi-level queue has the same type of tasks, and the tasks with high priority are stored into higher level queue. And in each queue, each element is sorted by release time. The distributor scans the first element of each queue, and selects one task with earliest release time among them. If the execution time is overlapped, the task with higher priority is selected. The selected task is enqueued into probing queue, and the first element in probing queue is dequeued for task execution.

The proposed algorithm controls the release time of tracking tasks. After target update, the executed tracking task gets new release time and the beam direction to the estimated position at the release time by updating interval allocation phase. The element is compared with other elements in the tracking queue to be sorted by release time and is put in proper position. If the release time is overlapped with other track execution, the task whose target has higher priority is located ahead. When the target is determined to be terminated in the track maintenance, the tracking element is removed from the tracking queue.

Target priority allocation is based on decision tree in [10][11]. The approaches were compared under the same initial conditions and the same tactical characteristics in respect of targets and environment. The required information to assign a priority is provided by a tracking algorithm. In our implementation, we defined the total 7 levels of priority.

V. Simulation Results

To evaluate the adaptive update rate algorithm we proposed, we made a simulator consisting of generator, scheduler and tracker, and evaluator. To make the input data of tracker, generator is used to replace transmitter, receiver, and signal

TABLE II

MEASUREMENT ERROR STANDARD DEVIATION

	value
range	17.3 (m)
range rate	3.40 (m/s)
azimuth	0.17 (degree)
elevation	0.17 (degree)

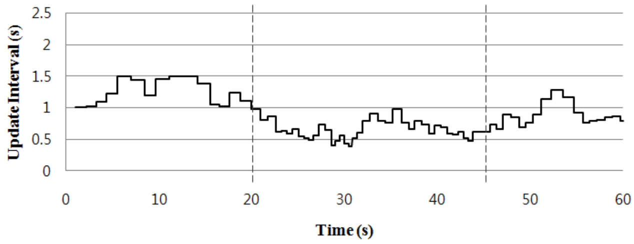
processor parts. The generator makes traces of targets which take certain types of movements.

Two kinds of maneuvering movements are considered, one is 3-dimensional turning (3DTR) and another is constant acceleration (CA) to model the possible movement of the targets. In 3DTR scenario, the target maneuvering occurs between 20 and 45 seconds by changing its direction with 4G acceleration. In CA scenario, the target maneuvering occurs between 20 and 40 seconds by acceleration to its traveling direction with 3G acceleration. Both targets are located upon 1000m height. As the measurement error, zero-mean white Gaussian noise is added as shown in Table. II.

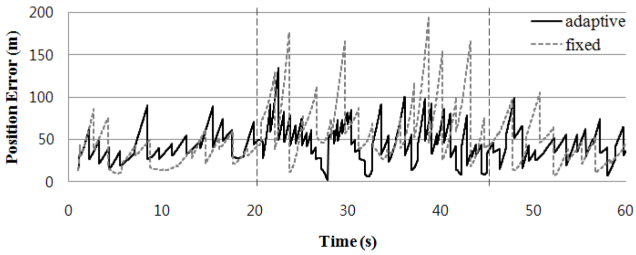
The scheduler and tracker work on generated target data under a certain beam scheduling policy based on adaptive update interval or fixed update interval for comparison. For evaluating a tracking accuracy of beam scheduling algorithm, the evaluator generates the tracking error and position error by comparing true state and filtered state.

Fig. 5 shows the results with 3-dimensional turning objects. The update interval under an adaptive control is shown in Fig. 5(a). According to this figure, the update interval starts with 1 seconds, but it increase to maximum update interval during the target is on straight-line motion. And the update interval decreases on maneuvering after 20 seconds and keeps the time interval during turning movement. The graph shows that the adaptive update interval control method can adjust update interval according to the motion of the target. Fig. 5(b) shows the position error including adaptive and fixed update interval methods. Compared to the fixed update interval method, the adaptive update interval method lower the tracking error when the target is on maneuvering movement between 20 and 45. In the simulation, the e_{bound} is set to 80 meters. It cannot guarantee the track error is maintained below e_{bound} all the times, but it tries to lower the tracking error below the threshold with high probability. In case of the adaptive update interval method, the position error has the value under the 80-meter boundary in 96.53% of interval, while the fixed update interval method shows that 83.91% of interval is under the boundary.

The simulation results for constant acceleration with e_{bound} of 80m is shown in Fig. 6. In this scenario, the update interval is decreased to prevent prediction error from increasing between 20 and 40 seconds, shown in Fig. 6(a). The prediction error is well maintained under the given threshold when the adaptive update interval control is applied, shown in Fig. 6(b).



(a) Update interval



(b) Tracking error

Fig. 5. 3DTR movement with 80-meter boundary

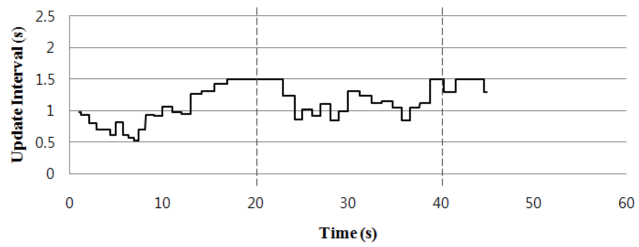
TABLE III

ERROR BOUNDARY, MINIMUM AND MAXIMUM UPDATE INTERVAL ACCORDING TO TARGET PRIORITY

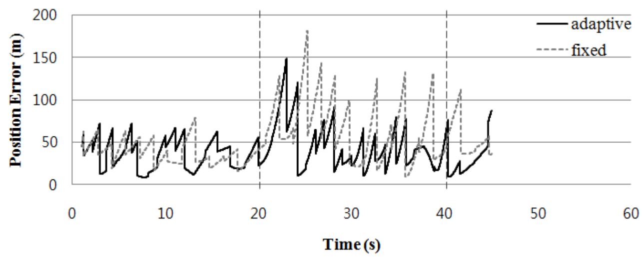
Target priority	0	1	2	3	4	5	6
e_{bound} (m)	250	180	130	100	80	60	40
Maximum (s)	5.0	4.0	3.0	2.25	1.5	1.25	1.0
Minimum (s)	0.8	0.8	0.8	0.4	0.25	0.18	0.13

In this evaluation, the adaptive update method shows 95.2% of interval is under the 80-meter boundary, which shows higher accuracy than that of the fixed update method. In case of fixed update method, the position error is lower in 85.06% of interval.

We also evaluated the priority allocation algorithm under 3DTR movement. The target movement used for test is 3DTR model with maneuvering at 20-36(s), 180(deg) turning, and 4G acceleration. At first, it moves away from and then comes up to a radar after turning. In this experiment, the error boundary, minimum and maximum update intervals for each target priority are set as shown in Table. III. According to the target movement, the priority varies as shown in Fig. 7. The update interval is allocated at each time considering the boundary corresponding to the target priority, shown in Fig. 8(a), and the prediction error is shown in Fig. 8(b) including the error boundary according to the target priority. When the target is maneuvering, the update interval is decreased when the target start maneuvering at 20 sec. The update interval is kept low after 32 sec because the target priority increases. In most time interval, the prediction error is maintained below the error boundary corresponding to the target priority.



(a) Update interval



(b) Tracking error

Fig. 6. CA movement with 80-meter boundary

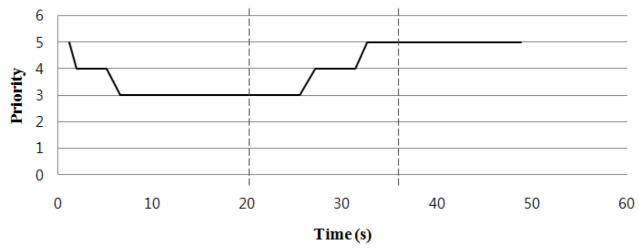
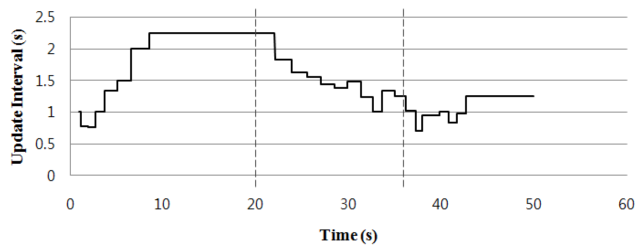
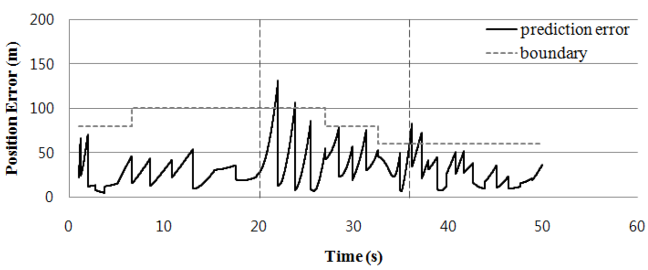


Fig. 7. Priority variation of target on 3DTR movement



(a) Update interval



(b) Tracking error

Fig. 8. 3DTR movement considering priority

VI. Conclusion

In this paper, the update rate allocation method based on innovation is proposed. First, we designed a new algorithm for calculating next adaptive interval for tracking targets, which is derived from the method of limiting the position error within a given threshold. The simulation results show that track error of the maneuvering targets is reduced compared to fixed update rate case. The amount of interval under 80-meter error boundary is increased by 11.3% when we used proposed adaptive update rate method. Second, we proposed a method of calculating the update rate based on the target's priority. By allocating the small error boundaries to high priority targets, we can allocate higher update rate to them. In military purpose of radar scheduling, target's threatening level should be essentially considered since the miss cost of each target would be different. The simulation results show that both the priority of target and update interval are dynamically changed during tracking task.

Acknowledgment

This work has been supported by Samsung Thales with whose permission this paper is published.

References

[1] G.A. Watson and W.D. Blair, *Tracking Performance of a Phased Array Radar with Revisit Time Controlled Using the IMM Algorithm*, IEEE National Radar Conference, 1994, p160-165.

[2] A. Yanbe, M. Ito, S. Tsujimichi, and Y. Kosuge, *Target Tracking with Adaptive Sampling Intervals Using a Phased Array Radar*, Electronics and Communications in Japan, 2002, Vol.85 Issue.10, p44-54.

[3] G. Van Keuk, *Adaptive Computer Controlled Target Tracking with Phased Array Radar*, IEEE International Radar Conference, 1975, p429-434.

[4] S.A. Cohen, *Adaptive Variable Update Rate Algorithm for Tracking Targets with a Phased Array Radar*, IEEE Proceedings, Vol.133, Issue.3, pp.277-280, 1986.

[5] D.J. Wilkin, I. Harrison, and M.S. Woolfson, *Target Tracking Algorithms for Phased Array Radar*, IEEE Proceedings, Vol.138, Issue.3, pp.255-262, 1991.

[6] H.J. Shin, S.N. Hong, and D.H. Hong, *Adaptive-update-rate Target Tracking for Phased Array Radar*, IEEE Proceedings - Radar, Sonar Navigation, Vol.42, no.3, pp.137-143, 1995.

[7] S.L. Coetzee, K. Woodbridge, and C.J. Baker, *Multifunction Radar Resource Management using Tracking Optimization*, Proceedings of the International Conference on Radar, pp.578-583, 2003.

[8] C. Ting, H. Zi-shu, and T. Ting, *An IMM-based Adaptive-update-rate Target Tracking Algorithm for Phased-array Radar*, Intelligent Signal Processing and Communication Systems, pp.854-857, 2007.

[9] Y. Kosuge, H. Kameda, S. Mano, and M. Kondou, *A Cartesian Coordinate Conversion Algorithm for Radar Tracking with Range Rate Measurement*, Electronics and Communications in Japan, Part 1, Vol.80, No.4, pp.51-61, 1997.

[10] F. Barbaresco, *Intelligent Multi-mission Radar Resources Management*, 2008 IEEE Radar Conference Tutorial, 2008.

[11] S. Miranda, C. Baker, K. Woodbridge, and H. Griffiths *Knowledge-based Resource Management for Multifunction Radar: a Look at Scheduling and Task Prioritization*, IEEE Signal Processing Magazine, Vol.23, Issue.1, pp.66-76, Jan. 2006.