Scan Scheduling of Multi-Function Phased Array Radars using Heuristic Techniques

Bill Gillespie, B. Eng, Royal Air Force
Evan Hughes, Ph. D., Cranfield University at RMCS
Mike Lewis, M. Sc., Cranfield University at RMCS

Key Words: multi-function radar, phased array radar, scheduling, optimization, airborne intercept, heuristic algorithm

SUMMARY & CONCLUSIONS

This paper describes a scan scheduler based on heuristic principles for MPRF airborne fire control radar. The design of the heuristic scheduler is based around a set of simple rules which work in conjunction with a priority framework to provide the required outputs to the system, in this case the next beam azimuth and elevation.

A new heuristic scheduling algorithm has been produced that incorporates the elements of the best published algorithms. The heuristic rule set incorporates a hierarchy of function priorities and an overload strategy that provides graceful degradation of surveillance tasks to service target tracking. Simulations were designed and executed to stress the full capabilities of the scheduler. Superior performance was demonstrated in all but the most extreme of overload conditions. In common with all priority-based systems the scheduler surveillance performance degraded under heavy load conditions. This scheduler mitigated this degradation by altering the surveillance volume under load.

A novel approach to prioritising beam position updates in heavy loading conditions was implemented such that the beam scanned from the boresight position first. This approach optimises search time by ensuring that the areas with high beam dilation compensation factors were updated at a lower rate.

Realistic parameters were used, based on an existing fighter radar system *(AN/APG-63)* and showed improvements when compared with that radar's existing mechanical system. The chosen PRFs and dwell times gave initial acceptable performance but the adaptability of the MFR allowed the variation of dwell time and data rate to show further performance gains.

The set of heuristics used in the scheduler showed emergent behaviour that took account of track loading and the search time available in an overload situation to cope automatically with loading and shedding of tasks. This non-programmed behaviour resulted in graceful degradation of system performance under overload conditions.

Although the graceful degradation is emergent behaviour, it can be deduced by formal reasoning and shown to be implied by the heuristic rules.

1. *INTRODUCTION*

Ever increasing demands are being placed on airborne radar as a primary sensor. The functionality of avionic equipment has increased as the price of silicon components has reduced. Military radar has followed this trend and the demands for greater functionality and flexibility in order to react to dynamic target behaviour, clutter and Electronic Counter Measure (ECM) environments has increased[1]. Functionality has further increased to encompass specialized functions such as mapping, non-cooperative target recognition and data fusion. The result is that the limits of mechanically scanned antennas have been reached due to the inherent inertia of a moving system. The logical progression has been to electronic beam steering and scanning driven by two considerations:

1. The need to maintain high scanning rates to cover large volumes or multiple ranges.
2. The need to interlace modes of operation between air to air and air to ground.

The electronic control of the beam position eliminates inertia and gives an almost instantaneous response. This inertia less response enables rapid interleaving of modes.

2. *SCHEDULER SPECIFICATION & REQUIREMENTS*

2.1 *Resource Management*

The control of airborne MFR has been described [2] as being divided into two parts: strategic decisions and scheduling. The strategic decisions identify the importance of the task whilst the scheduling arranges the tasks in time. The importance of a given task, such as search or track, may not hold under all circumstances and may have to be promoted under appropriate conditions. The scheduler must allocate time intervals such that the resources are utilised in the most efficient manner based on power and time constraints.

The precise choice of optimum resource allocation is the underlying feature of scheduler design however all schedules share common inputs to the tasks that they control:

1. Task priority.
2. Task update rate.
3. Load changes.
4. Overload situations

0-7803-8882-8/05/$20.00 (C) 2005 IEEE

Unlike rotating radar, the position of a target or update of a target can not be scheduled to take place on boresight where maximum antenna gain is available. The radar will be required to spend longer times in areas where extended dwell times are necessary to compensate for off boresight beam dilation.

The task scheduler must determine the optimum sequence of tasks taking into account beam dilation, task priority, waveform selection, time available to complete a task and the required beam position revisit frequency.

There are numerous ways to achieve the scheduling of tasks and many schedulers exist each using different methods in an attempt to achieve essentially the same goals.

2.2 *Specification*

The scheduler has been designed to control an antenna array of 3040 elements over a 1m diameter producing a nominal beam width of 3°. The coverage is a total of 861 beam positions in a field of view of ± 60° in azimuth and ± 30° in elevation. The surveillance volume is set to a maximum horizon of 130 km and is limited in high angle look up positions to 30 km.

2.3 *Scheduling requirements*

Many current scheduling algorithms are based on cyclic-executive buffers with a limited ability to alter the execution order. The resulting scheduler is sub-optimal since it does not make the most efficient use of radar resources and does not guarantee system performance. Scheduling must encompass prioritisation from mission parameters such as expected target range, track retention and tracking accuracy. In addition the scheduler must cope with overload situations with the ability to shed load, allowing for event and decision driven changes and cope gracefully with task time-outs when a process overruns. The scheduler needs to take account of the waveforms being used by the radar and arrange the schedule to obtain minimum time delay and use of radar energy. Dynamic reallocation of resources must be achieved and is dependent on accommodating the duty, occupancy and processor loading. By breaking the time line into small time slices, which are not initially associated with any function, the scheduler can allocate time to each waveform rather than have large areas dedicated to a given function. This method of allocation gives the scheduler the flexibility to change the order of functions with minimal wasted energy.

3. *SCHEDULER DESIGN*

3.1 *Heuristics*

Heuristics are defined as rules of thumb or sets of guidelines to follow, as opposed to an invariant procedure, so that the decisions made at any instant are based on the current radar environment and are influenced by previous decisions. It is this environmentally driven quality that gives the heuristic scheduler its apparent ability to learn the most appropriate route to the required results. On the other hand this may result in the scheduler having an unpredictable solution space which does not guarantee results.

3.2 *Scheduler Functions*

The functions incorporated in the scheduler simulation are

1. Target Confirmation and Track Initiation
2. Target Track Maintenance.
3. Search.

The scheduler generates detections and target tracks corresponding to the positions of the search beam during MFR operation. These tracks are then maintained over their lifetime by the scheduler track maintenance algorithms.

3.3 *Allocation of Priorities.*

Task	**Priority**
Confirmation Management	1
Sequential Calibration Management	2
Track Management	3
Search Management	4

Table 1: AMSAR Scheduler Priorities

A scheduler algorithm has been described by Powis et al [1]. The program is referred to as AMSAR. Tasks are allocated by priority according to the current radar operational mode as shown in Table 1. During "search only" modes of operation, neither the confirmation nor the track management subsystem is required to be scheduled. The scheme is fairly simple: confirmation attempts have the highest priority and should not be delayed by a lesser priority task. In a typical scenario with no targets present, the time will be devoted to search dwells with the occasional interruption for calibration tasks. When targets are present, confirmation attempts will be made as soon as possible after the search dwell that triggered them. In the AMSAR system the resource allocation frame is approximately 1-2 seconds and this defines the granularity at which percentage allocations to the types of tasks can be assessed. This timeframe is somewhat arbitrary, but is based upon the typical maximum timescale within which a track revisit would be desired. From the simple rules defined for AMSAR, the track updates would be performed at the beginning of a new resource allocation frame unless a higher priority task was pending.

Task	**Priority**
Target Confirmation	1
Target track updates	2
Boresight weighted search beam position	3
Longest overdue update search position	4

Table 2: Heuristic Scheduler Priorities

The heuristic scheduler discussed in this paper is also based on four levels of priority, although this could be expanded to encompass any nested priority system. The levels of priority on order of execution are shown in Table 2.

4. *HEURISTIC RULES*

Like the AMSAR scenario, the highest priority is given to target confirmation, which is scheduled to be performed immediately upon target detection, and cannot be over ruled by any other task priority. The resource allocation frame in the heuristic scheduler is very short and calculated after each successive search beam dwell. The track updates are given higher priority than search dwells but the two are interleaved since track updates hold rigidly to their update frequency whereas search dwells cycle through continuously from the oldest updated position. Track updates are calculated on a beam to beam basis.

The following twelve rules govern the adaptive search and track behaviour of the scheduler:

1. Every beam position must be searched on a regular basis at a frame rate dictated by the maximum closing velocity.
2. Where an excess of search time remains repeat searching until the frame time is exhausted.
3. Irrespective of whether a track update task exists at a given beam position, a search task must be performed.
4. Search tasks must be performed on the beam position with the longest elapsed time 'since the last update'

Compliance is ensured by the use of two time ordered queues, a search position queue and a track queue. The population of the search position queue is independent of the track queue.

5. Search beam positions must be prioritised every frame based on their displacement from the radar boresight.

Rule 5 implements Priority Level 3 in Table 2. The search beam scans from the centre of the field of view outwards. The interval to next update is weighted such that the longest interval is applied to the positions at the edge of the beam. The weighting is based on the beam dilation factor which is least on boresight and greatest at the edge of the beam. The beam dilation necessitates longer dwell times at large deflections compared with the boresight. Since the interval between updates on boresight is shorter than that between large deflections boresight positions will be visited more frequently, in a frame, than large deflections.

6. On target detection, in any given beam position, during a search task requires a target confirmation to be performed. This is to eliminate false alarms.
7. On a confirmed detection a track initiation must be performed, the track log updated and the track passed to the target tracker.

Rules 6 and 7 are nested within the *SEARCH* process. This is defined in Hoare's CSP notation [3] as:

$$SEARCH(search) ::=$$
$$\left(\begin{array}{l} search.bins = SET\_BEAM(search.az, search.el) \rightarrow \\ \mu X : tgt := CFAR(search.bins) \rightarrow \\ \left(\begin{array}{l} tgt.present \rightarrow CONFIRMATION(tgt) \rightarrow X \\ \mid \neg tgt.present \end{array} \right) \end{array} \right)$$

Upper case denotes a process, lower case denotes a variable.

CONFIRMATION is assumed to include all other functions for initiating a confirmed track. Notice the recursion to handle multiple targets in the beam.

8. Once a track has been formed, the target tracker must perform track maintenance tasks at intervals based on the dynamic behaviour of the target. A track is removed from the track list when its range exceeds the maximum range of the radar or it falls below 20km.

Rule 8 is applicable to all trackers and is not considered further here.

9. All track maintenance tasks must be performed each frame. The total time for track maintenance tasks will be calculated at the start of each frame and reserved to allow all tracks to be updated.

Rule 9 is an invariant and can be expressed as:

$$frame.time$$
$$\geq track_N.dwelltime + \sum_{n=1}^{N-1} track_{n+1}.duetime - track_n duetime$$

10. The total time available for search tasks will be the frame time not allocated to track maintenance.

Rule 10 is a simple invariant

$$searchtime = frametime - \sum_{n=1}^{N} track_n.dwelltime$$

It follows that

$$searchtime \geq \sum_{m=1}^{M} search_m.dwelltime$$

11. A track maintenance task must only be performed when it is due according to the frequency dictated by the data rate determined from knowledge of the target dynamics.

This now leads to a definition of *TRACK* that complies with rule 11.

$$TRACK(track_n) ::=$$
$$WAIT(track_n.duetime) \rightarrow$$
$$UPDATE\_TRACK(track_n)$$

12. Track maintenance tasks take priority over search tasks where conflict exists in 'due time.'

Although an execution time is allocated to each search task no direct use is made of it in the scheduler. It is used as a convenient means of time ordering the searches. The execution time of the search at the head of the queue is '*now*'. If the time interval between '*now*' and the next scheduled track update is less than the proposed search dwell time, a track update is scheduled. The search dwell is delayed until the track update has been completed.

This leads to rule 12 being expressed as

$$SCHEDULE\_OP(search_m, track_n) ::=$$
$$\left(\begin{array}{l} now + search_m.dwell\_time > track_n.duetime \rightarrow \\ TRACK(track_n) \rightarrow SCHEDULE\_OP(search_m, track_{n+1}) \\ \mid now + search_m.dwell\_time \leq track_n duetime \rightarrow \\ SEARCH(search_m) \rightarrow SCHEDULE\_OP(search_{m+1}, track_n) \end{array} \right)$$

It is clear from the recursion that a search operation may be delayed many times in favour of a series of track updates. This has a significant impact on the search behaviour when a large number of tracks have been generated.

5. *SCHEDULER PERFORMANCE UNDER LOAD*

In the design of the heuristic scheduler no use, apart from as an ordering mechanism, has been made of a search beam position scheduled update time. The next beam position update in the queue is executed immediately unless pre-empted by a track update. Track loading and shedding of tasks in an overload situation must be performed by the scheduler. The choice of heuristic resulted in behaviour emerging that automatically took account of the track loading and the search time available.

The following figures show a typical set of results obtained over 100 runs of a Monte Carlo simulation. The curves show the averaged results alongside 3.09 standard deviations giving an indication of the maximum excursions over the various runs.

The series of curves is for heavy loading, ie., 1100 targets, Mach 5 closing 0.5/2 Dwell/Data rate, which allows the formation of tracks from the search detections. The result is a triplet of dwells for detection, confirmation and track initiation. Since the frame is fixed the result is a reduction in the time available for the search function. Each new track is assigned an opening or closing velocity and an associated range. This dictates the data rate for each track beam.

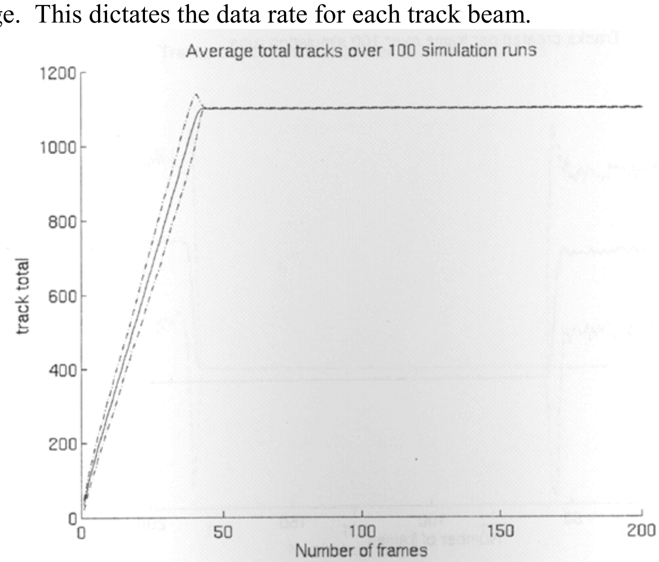

Fig 1. Total Target Tracks

Target loading of the scheduler has been simulated to show a rapid increase in target tracks and gradual tail off as the targets reach either the maximum horizon or the minimum close in range, Figure 1.

Figure 2 shows the build up rate of 16 tracks per frame over 70 frames. The frame length is 8.6 seconds and Figure 3 shows the search occupancy reducing from 100% to 65% during the loading of 125 simultaneous target tracks. This is in accordance with Rule 10:

$$searchtime = frametime - \sum_{n=1}^{N} track_n .dwelltime$$

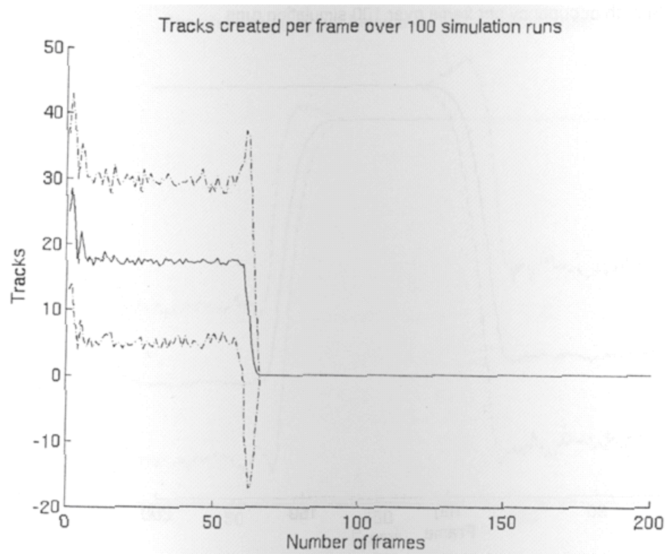

Fig 2. Tracks Created per Frame

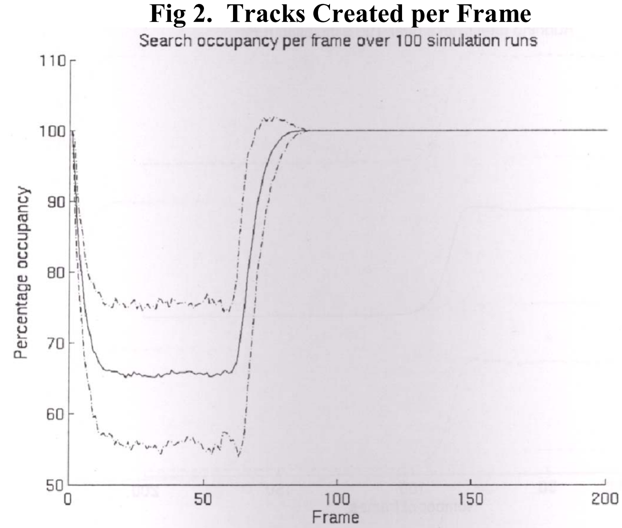

Fig 3. Search Occupancy, 125 Target Tracks

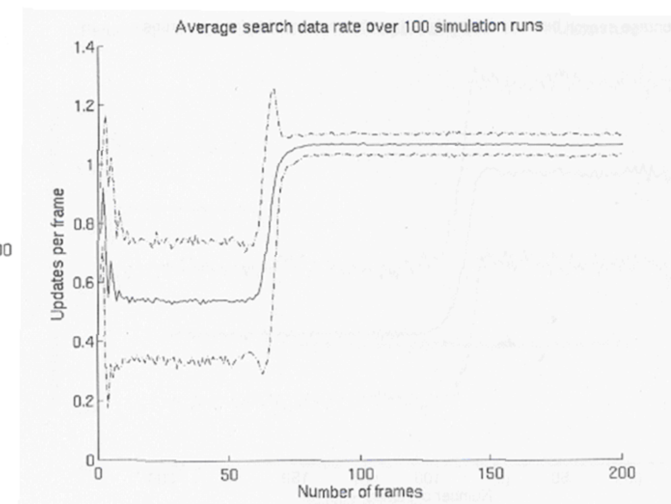

Fig 4. Updates per Frame

The Data Rate reduces to 0.55 under load and recovers to 1.1 when the load is removed. This is illustrated in Figure 4 and is again to be expected from Rule 10.

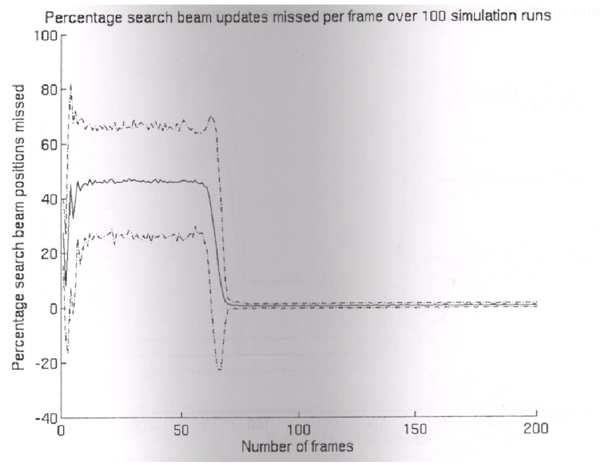

Fig 5. Percentage Search Positions Missed per Frame

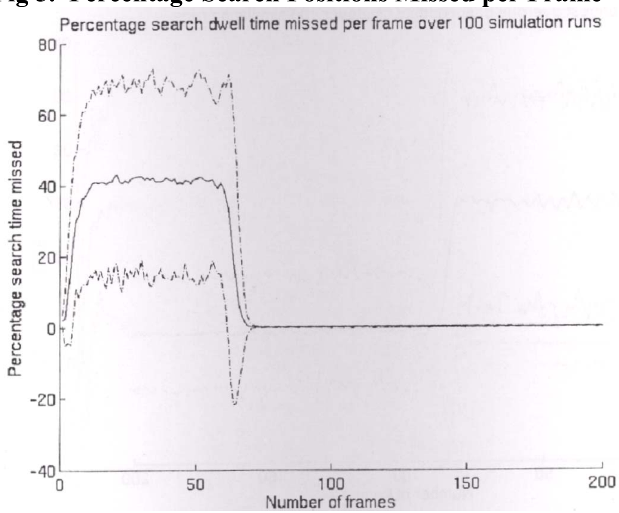

Fig 6. Percentage Search Dwell Time Missed per Frame

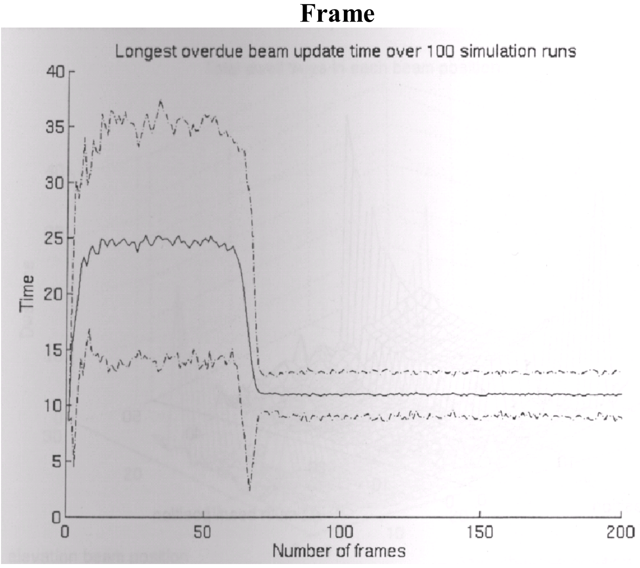

Fig 7. Longest Overdue Update Time

Figure 5 shows the percentage of search positions missed per frame under the full load conditions. This amounts to 44% and also accounts for 40% of the search dwell time as shown in Figure 6.

The longest overdue update is illustrated in Figure 7 and is 24 seconds.

5. *EMERGENT BEHAVIOUR*

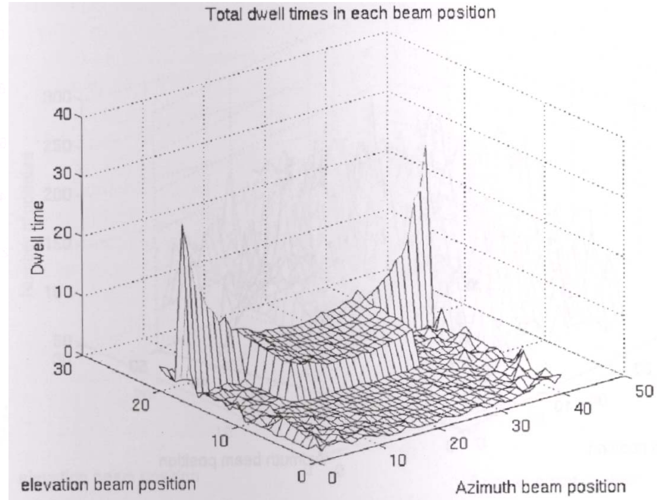

Fig 8. Total Dwell Times in Each Beam Position

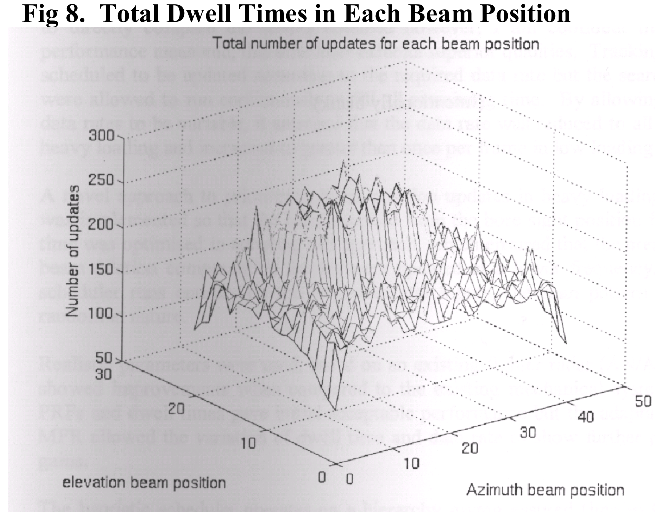

Fig 9. Total Number of Updates per Frame

Figure 8 illustrates the distribution of Total Dwell Times across the beam positions. The detailed results were interesting in that they showed that the prioritisation placed on updating from the central beam position outwards resulted in the data rate reducing further at the extreme limits of scan.

In essence under high loading, the number of updates that were given to the long dwell times at the edges of the search pattern were gradually tapered, as shown in Figure 9, giving a tunnel vision approach. This method ensures that the limited time the scheduler has is used in the most time and gain efficient way, concentrating on small dwell times nearest the maximum gain region of the array. This emergent behaviour dynamically allocates the most efficient use of the time available for search under load and when the load is reduced, automatically fills the entire search volume again.

This is to be expected from Rules 10 and 12 which prioritise short dwell times. Since short dwell times and early visitation are allocated to beam positions at the centre of the field view then they tunnel vision effect is to be expected. In particular since the frame time is fixed and

$$searchtime = frametime - \sum_{n=1}^{N} track_n.dwelltime$$

an increase in the number of tracks will cause a reduction in available search time. The reduced available search time will mean that the higher priority central positions will exhaust the available search time per frame thus prevent the lower priority edge of beam positions being visited.

6. *SEARCH STARVATION UNDER HEAVY TRACK LOADING*

In discussing the definition of the process *SCHEDULE_OP* it was observed that a search operation may be delayed many times in favour of a series of track updates and that this has a significant impact on the search behaviour when a large number of tracks have been generated. This will now be examined in a more formal manner.

From the definition of *SCHEDULE_OP* the precondition for a track update to be selected in preference to a search is

$$now + search_m.dwell\_time > track_n.duetime$$

The precondition for a sequence of track updates to delay a search is

$$\forall n \mid n : \mathbb{C}\in \{k \cdots l\}, \exists m \mid m : \mathbb{C}\{1 \cdots M\}$$
$$\bullet\, now_n + search_m.dwell\_time > track_n.duetime$$

The resultant trace (sequence of events) of *SCHEDULE_OP* will be:

$$\left\langle \begin{array}{l} now_k + search_m.dwell\_time > track_k.duetime, \\ TRACK(track_k), \\ now_{k+1} + search_m.dwell\_time > track_{k+1}.duetime, \\ TRACK(track_{k+1}), \cdots, \\ now_l + search_m.dwell\_time > track_l.duetime, \\ TRACK(track_l) \end{array} \right\rangle$$

Since the sequence of searches and thus their prioritisation is based on

$$search_m.dwell\_time \le search_{m+1}.dwell\_time$$

and

$$search_m.dwell\_time \le search_{m+1}.dwell\_time \Rightarrow$$
$$search_m.beam\_angle \le search_{m+1}.beam\_angle$$

From the above it can be inferred that under heavy track loading the greatest likelihood is that extremes of scan angles will experience starvation by being given a very low update rate.

It has been observed that under extreme loading conditions (11000 target tracks formed), the search occupancy is reduced without limit and whilst search tracks are always scheduled, the search data rate of the array reduces to almost zero resulting in boresight only updates.

It is significant that although the graceful degradation of the system that is observed under heavy track loading is emergent behaviour, i.e., not programmed in, it can be deduced by formal reasoning. The ability to express the heuristics in symbolic form permits manipulation of the rules and performance of proofs of behaviour such that a high confidence may be placed in the behaviour of the scheduler.

7. *HEURISTIC ALGORITHM SCHEDULER STRENGTHS*

During the operation of the MFR, the scheduler adapts to the load encountered by the track maintenance system by adjusting the search occupancy to fill the time that is not required for tracking. This allows a greater than unity update rate for search beam positions under low track loading conditions. The key heuristic, Rule 2, causes the search position queue to be cycled until the frame time is exhausted.

Track formation is made immediately following a detection and confirmation. This is unlike a mechanically scanned antenna radar which may adopt a 3 from 4 detection strategy requiring a minimum of 3 antenna sweeps.

Under heavy track loading, which naturally forces a reduction in the search occupancy, the algorithm adjusts the priority of the search beam positions to concentrate on the boresight directions and, based on the beam dilation compensation factor, will fan out to the extreme beam positions. This provides high efficiency in the early stages of each frame so that dwell times are at a minimum and the most effective use is made of the limited occupancy by concentrating on the threat area ahead of the aircraft.

By concentrating on the boresight positions under load but maintaining the heuristic of updating the oldest beam position, every element of the array will be illuminated at some stage in a sequence of frames. The outcome is that the overall data rate, particularly in respect of search, is reduced under loading and recovers when the loading is removed.

The ability of an MFR to form large numbers of beam positions per second allows the dwell times to be reduced and the data rate to be increased. This was demonstrated by performing a half dwell double data rate simulation which showed the ability of the scheduler to cope with high beam switching rates. The increased data rate resulted in up to tenfold increases in the performance of the MFR. This could be increased still further up to the maximum switching speed of the array producing significant performance gains.

REFERENCES

1. L. Powis, et al., "Adaptive Radar Control of multifunction radars using artificial intelligence", *Radar 92*, Brighton, UK, IEE International Conference, pp 426-429
2. D. Stromberg, P. Grahn, "Scheduling of tasks in phased array radar", Proc., IEEE International Symposium on Phased Array Systems and Technology, 1996, pp 318-321
3. C. A. R. Hoare, "Communicating Sequential Processes", Prentice-Hall International, London, 1985