

# Allocating Sensing Resources with Incomplete Information

Robin R. Murphy

Center for Robotics and Intelligent Systems

Colorado School of Mines

Golden, CO 80401-1887

phone: (303) 273-3874 fax: (303) 273-3875

rmurphy@mines.edu

## Abstract

This paper presents work in progress on allocating sensing resources for an autonomous mobile robot in the presence of incomplete information. It provides a taxonomy of the types of contention for sensing resources in mobile robots and the categories of available information. Our assumptions about the structure of behaviors and the characterization of the scheduling problem suggest an iterative repair strategy, where an initial schedule is formed from each behavior's sensing requests and alternative logical sensors are substituted if needed.

## Introduction

This paper presents work in progress on allocating sensing resources for an autonomous mobile robot in the presence of incomplete information. This work is motivated by our larger research goal of *reactive sensing*. Reactive sensing is intended to allow sensing (or data interpretation) processes to act as autonomous specialists in maintaining the most robust and certain perception for the given behavior, state of the robot, and state of the world. In order to be useful, the reactive sensing mechanisms should be compatible with existing reactive and hybrid reactive/deliberative architectures rather than viewed in isolation from motor control.

One problem is how to provide adequate perception for all behaviors when there is contention for sensing resources. For example, consider a mobile robot with a camera fixed to the base. In order to look to the left, the robot base must turn to the left. In this case, the robot cannot follow the hall (head straight), and look to the left for a door simultaneously. But the robot may be able to use alternative sensing strategies (via a logical sensor mechanism (Henderson & Shilcrat 1984)) to follow the hall or look for the door, resolving the contention problem.

The allocation process is not static; that is, it cannot assume that decisions are carried out without interruption. The reaction of the robot to the environment implies the possibility of unanticipated changes in which behaviors are active. This means the sensing demands

will change accordingly. Likewise, the same behaviors can remain active but the sensors assigned to them can fail. Again if alternative sensing strategies are available, the robot can assign the new sensors and continue to execute the task(s). In either case, the robot is dynamically presented with the possibility of contention for resources and must resolve that contention in real-time or else have to suspend execution.

The generic allocation and interruption policies must be able to operate with incomplete information from at least two sources: the plan itself and unpredictable sensing failures. First, the robot may not have a plan or only a partial plan of what behaviors will be active over the relevant time frame. In a purely reactive architecture, only the set of possible behaviors to accomplish a task(s) is known in advance. The order and duration of their instantiation (and thereby, their sensing demands) is strictly a function of what is encountered in the environment. In a hybrid deliberative/reactive architecture, the planner may be able to project a desired sequence of behaviors but not necessarily the duration of each behavior. Furthermore, the plan may have to be repaired in response to unanticipated events (e.g., a hallway is completely blocked). Second, even if a plan exists and is valid for the time interval of interest, the availability of sensors may change at any time due to sensor malfunctions and/or changes in the environment.

Without complete information, "optimal" allocation of sensing resources may actually interfere with the robot's overall performance. Let optimal mean that each sensing process is producing the most certain perception possible (each behavior is assigned the best sensors for that job according to some global utility function). If the robot cannot project what the sequence of behaviors are and attempts to allocate resources optimally each time the set of active behaviors changes, then the robot is susceptible to thrashing (i.e., frequent and unproductive reallocation). Consider that behavior *A* operates continuously, and is given sensor *x*. Then when behavior *B* is activated, sensor *x* is "robbed" from *A*, given to *B*, and sensor *y* is allocated to *A*. When *B* is deactivated, *x* is returned

to  $A$ . The temporary improvement in sensing certainty for  $A$  and  $B$  versus adequate, but not optimal, sensing may not be worth hidden costs such as slew time of the sensors, delays in bootstrapping the perceptual algorithms, etc. Continuous optimality may result in the robot spending the majority of its time deliberating and bootstrapping sensing, forcing the robot's task execution to be effectively halted. Therefore, the allocation mechanism must be able to function under different levels of incomplete information; on one end of the continuum, the sequence and duration of the behaviors may be known *a priori*, on the other, the behaviors may be purely reactive.

Our approach is to treat sensing resource allocation for autonomous mobile robots as a *scheduling* problem. The set of currently instantiated behaviors and their associated sensing demands are viewed as representing a plan, paraphrasing Fox's definition (Fox 1994): *a sequence of behaviors that achieve one or more tasks and satisfy a set of domain constraints*. The allocation mechanism serves as a scheduler, again paraphrasing (Fox 1994), *selecting among alternative sensing strategies and assigns resources and times for each sensing activity so that the assignments obey the temporal restrictions of activities and the capacity limitations of a set of shared resources*.

The remainder of the paper is laid out as follows. Section briefly summarizes related work in planning and sensing for autonomous mobile robots. The assumptions about the organization of sensing in reactive and hybrid architectures are presented in Section . A taxonomy of the types of contention for sensing resources is presented in Section , followed by a list of the categories of available information for scheduling in Section . Our proposed intelligent scheduler is described in Section , and the ramifications for planning for acting is discussed in Section . The paper concludes with a summary and future work (Section ).

## Related Work

Efforts in integrating planning and sensing within an architecture for autonomous mobile robots have been recently reported by (Noreils & Chatila 1995) and (Chen & Trivedi 1995). (Noreils & Chatila 1995) concentrate on monitoring the robot's execution and repairing plans. The architecture of (Chen & Trivedi 1995) focuses on the impact of sensing on planning, especially how to take advantage of new information. In contrast, our efforts concentrate on the inverse problem: how to plan for sensing? Furthermore, we wish to avoid the computational overhead and time delays of interleaving planning and execution at the reactive layer incumbent in the (Chen & Trivedi 1995) approach unless a) a failure occurs (mandating a new plan) or b) an opportunity arises for significant gains in sensing.

The specific problem of contention has either been ignored, acknowledged but not implemented (Simmons, Lin, & Fedor 1990), or handled by sequencing

the control of the sensor (for example, behavior 1 turns the camera to the left and senses, notifies behavior 2 that the camera is free, which then turns the camera to the right and senses, then behavior 1 turns it to the left again, etc.) as in (Kaelbling 1987). More satisfactory schemes which also facilitate sharing of resources, permit the substitution of alternative sensors, and streamline control are desirable.

Aspects of our approach bear a superficial resemblance to the CIRCA architecture (Musliner, Durfee, & Shin 1993). That system is primarily concerned with guaranteeing safe real-time execution of behaviors. Sensing influences the frequency of execution of Test-Action Pairs (TAPs). If a sensing resource is not available or adequate, CIRCA directs the robot to substitute another sensing resource or slow the robot in order to reduce the sensing demands. No discussion is given of how an appropriate substitute sensing source is identified, nor of the impact that the selection process would have on the TAPs schedule (e.g., would all activity cease while the sensing allocation was determined?). The architecture is also heavily dependent on the assumption that the robot's interactions with the environment can be modeled, allowing all states to be known. As noted in the Introduction, the sensing resource allocation is complicated by unpredictable sensing failures. Instead, CIRCA and our work can be taken as complementary, one concentrates on scheduling sensing and acting, while the other schedules sensing in such a way as to make sensing and acting possible.

## Organization of Sensing

In order to discuss the impact of incomplete information on allocating sensing resources, an overview of the assumptions about how sensing is organized is now presented.

There is no generic reactive or hybrid deliberative/reactive architecture. However, representative systems have enough attributes in common that we propose a basic architecture. The generic system has a *task planner* which is responsible for setting the robot's goals in the form of one or more tasks, specifying the behaviors needed to achieve those tasks, and monitoring their *execution*. In a purely reactive system, the task planner is the set of activation conditions for the behaviors, while in a hybrid architecture, the task planner is a true planner.

It is assumed that the task planner specifies a list of one or more concurrent *tasks* and the behaviors to accomplish them. This list of tasks is known as the *plan*. For example, a plan may represent directives for the robot to 1) go to end of hall while 2) avoiding obstacles. A task is accomplished via one or more *behaviors*, which may operate sequentially and/or concurrently. Going to the end of the hall may be accomplished via a *navigate-hall()* behavior. The planner also selects the behaviors to accomplish each task. This selection

of the behaviors may be simple, a task may have an associated collection of behaviors, or it may involve true planning.

It is assumed that a behavior is decomposed into a *motor schema* and a *perceptual schema*, following (Arbib 1981). The motor schema component represents the physical pattern of activity, while the perceptual schema defines the perceptual process needed to support the actions of the motor schema. The motor and perceptual schemas act as autonomous, independent agents; essentially logical behaviors and sensors (Henderson & Shilcrat 1984). When the behavior is instantiated, the schemas themselves determine the most appropriate activity and perception for the circumstances, not the planner. This distribution of planning effort allows reactive and hybrid architectures to be highly modular and reduces computational and representational complexity. It is important for a sensing resource allocation mechanism not to subvert this organization.

## Types of Contention for Sensing Resources

A sensing resource consists of control over the sensor position (e.g., pointing), control over hardware parameters (e.g., autofocus, light sensitivity, firing frequency, etc.), and receipt of the output. In a behavioral system, more than one behavior can share the output from the same sensor and process the output using behavior-specific algorithms if the algorithms expect the sensor to be at the same position with the same parameters.

Contention for sensing resources stems from three sources:

1. *No contention.* This arises when all perceptual schemas have access to their preferred sensing resources during their execution. This could be the result of each behavior having a perceptual schema with a dedicated sensor, or multiple perceptual schemas being able to share a sensor. This is the simplest case and is included both for completeness and to emphasize that the allocation mechanism should be effective for both complex and simple situations.
2. *Competition by two or more perceptual schemas for same resource.* In this case, multiple perceptual schemas prefer the same resource, which cannot be shared. For example, two schemas may want control of the same camera but wish to point it in different directions at the same time.
3. *Competition by two or more perceptual schemas with a motor schema for the same resource (effector).* This case represents the example in the Introduction of a camera mounted to a robot base, where the sensing interferes with the direction of travel. It should be emphasized that adding sensing effectors (put the camera on a panning mast) does not solve

this problem for all situations. Eventually, a scenario equivalent of being able to turn your head only so far before you must turn your body will surface.

It is distinct from the previous case of competition between multiple perceptual schemas in that it adds the challenge of how do motor and perceptual schemas they communicate, negotiate, and/or be supervised.

The goal of sensing resource contention is to find a set of adequate substitutions which permit all behaviors to continue to execute concurrently. Otherwise, the plan, which assumes concurrent execution, must be declared as failed. The solution implemented in (Kaelbling 1987) of changing the order of execution (i.e., each behavior take a turn at controlling the sensing resource for) of behaviors could not arise from the scheduler, but must reflect a change by the task planner. This is to insure that changing the order of the tasks to satisfy sensing does not violate some other domain constraint. Consideration of domain constraints are by definition a function of the planner, not the scheduler.

## Types of Incomplete Information

The levels of knowledge associated with various behaviors will impact a scheduler. At this point in time three levels have been identified. In order to attempt to make this discussion clearer, the following terminology is introduced. The set of behaviors which are active at the same time is denoted by  $\mathcal{B}$ .  $\mathcal{B}$  is distinct from the set of all possible behaviors; not all behaviors will be instantiated at the same time, all the time. Robot execution can then be viewed as sequence of  $i$  sets of active behaviors,  $\mathcal{B}_{1,I_1}, \mathcal{B}_{2,I_2}, \mathcal{B}_{i,I_j} \dots, \mathcal{B}_{n,I_m}$ , each of which has a particular start time and duration denoted with the subscript  $I_j$ , for the  $j$ th duration interval. Note also that resources must be allocated to satisfy the current  $\mathcal{B}_{i,I_j}$  for duration  $I_j$  in order for the plan to continue to be executed.

The levels from most informed to least are:

1. *Can project the sequence of active behaviors,  $\mathcal{B}_{1,I_1}, \mathcal{B}_{2,I_2}, \dots, \mathcal{B}_{n,I_m}$ , and the duration of time the robot will spend executing each set.* In this case all  $n$  sets are known as well as how long the system will reside in execution of each set. This situation would arise when the robot was traversing a known area using a predefined collection of behaviors; for example, using an abstract navigation behavior to go down a hall of known length or average traversal time. This is the ideal case, where the scheduler has a time horizon that it can optimize over. However, this projection may be disrupted by reality, in particular, a highly cluttered hall way or a sensing failure. Therefore the scheduler must be able to handle dynamic changes even for this case.
2. *Can project the sequence of active behaviors,  $\mathcal{B}_{1,I_1}, \mathcal{B}_{2,I_2}, \dots, \mathcal{B}_{n,I_m}$ , but not the duration of time*

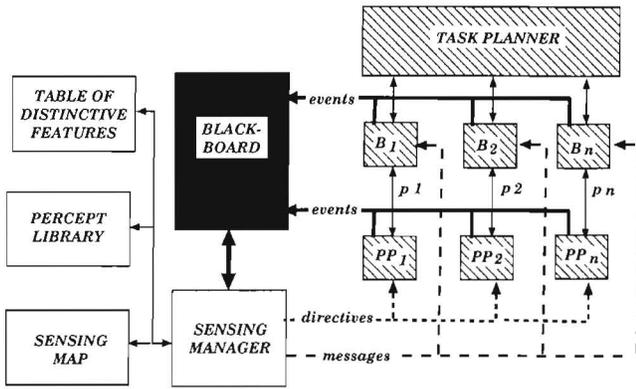


Figure 1: Layout of Sensing Manager

the robot will spend executing each set. All  $n$  sets are known but not the associated duration,  $I_m$ , of each. This would be the case where the robot was using a predefined collection of behaviors to navigating an area not stored in memory (i.e., going down a new hall).

3. *Cannot predict the sequence of active behaviors or the duration.* Neither  $n$  nor  $I_m$  is known. In this event, the robot would be operating almost at a purely reactive level. It also includes knowing only a subset of each  $B_i$ , where the robot might know that avoid-obstacle will always be active, but nothing about other behaviors.

In order to be efficient and prevent thrashing, the scheduler must be able to exploit any look-ahead information. For example, Level 3 implies that the scheduler has to catch up; that it can only allocate resources for the current situation but that situation may change momentarily (or even during the allocation process). Level 1 offers an almost perfect look-ahead. But designing a scheduler that works well for Level 1 information may not work well for Level 2 information. Therefore, these levels serve as canonical test cases for designing and evaluating the scheduler.

### Intelligent Scheduler Approach

Our approach to the problem of allocating sensing resources within the organization of a reactive or hybrid architecture is to have a global sensing manager which serves an intelligent scheduler. The choice of the intelligent scheduling algorithm(s) depends on the characteristics of the scheduling problem, which is defined below in Section . In response to these characteristics, we have formulated the iterative repair strategy presented in Section .

### Sensing Manager

The global sensing manager, shown in Fig. 1, is treated as an additional module in a reactive or hybrid architecture. Its purpose is to dynamically coordinate and

control sensing allocation for all active behaviors on the global level. The manager is intended to preserve the behavioral structure yet it needs to have global data structures, such as a sensing map representing the current area of operation for each sensors. Therefore, it is contained in a separate unit. The manager takes requests for sensors and allocates resources. Besides intelligent scheduling, the sensing manager also performs diagnostic reasoning to identify sensing failures. The activities of the sensing manager are beyond the scope of this paper.

### Characteristics of the Scheduling Problem

We define the scheduling of sensing resources for an autonomous mobile robot as:

- *It is not a dispatch problem.* The scheduler does not change or control the order of execution of the behaviors; instead it schedules resources to permit the execution of the behaviors in the order prescribed by the task planner.
- *It is a multi-agent planning problem.* The perceptual schema for each behavior represents a separate agent. Furthermore, we assume that agent has limited intelligence; the perceptual schema can construct a partial preference ordering on its different methods for sensing the percept.
- *The agents are uncooperative.* Each task wants to maximize its own goals, in effect, each behavior wants the best sensing for its purpose.
- *Search is combinatorially small.* Unlike job shop scheduling and other traditional scheduling applications which may attempt to schedule thousands of activities, a robot is expected to have 10 or less behaviors active at a time.
- *The potential for resource contention is high.* Although a robot may have few concurrent behaviors, it will probably have even fewer sensors.
- *The time devoted to scheduling is limited.* The robot must operate in real time, therefore it can't deliberate indefinitely. The allocation mechanism must be an anytime algorithm.
- *The agents may or may not be able to project sensing needs into the future.* As noted earlier, future sensing demands may or may not be known, and the robot may encounter unpredictable sensing problems. Therefore, the system needs to be able to repair a set of sensing resource allocations.
- *A task or behavior cannot be removed from the plan in order to make scheduling resources possible.* If a task or behavior was removed, the plan itself has failed and must be repaired.

### Iterative Repair Strategy

The proposed resource allocation algorithm for our sensing manager uses a *repair* methodology rather than

a *constructive* methodology. The algorithm takes advantage of the fact that the set of active behaviors defines a set of active perceptual schemas. The incumbent sensing demands from these schemas can be considered to form a schedule. If there is no contention, the schedule is sufficient and resources can be allocated accordingly. If contention occurs, the schedule must be repaired. In order to prevent thrashing and to capture an anytime flavor, repairs are done iteratively.

The key structure is the perceptual schema. The perceptual schema serves as an intelligent agent which can reason about all possible ways of accomplishing a perception. Another way of stating this is that a perceptual schema contains references to all relevant logical sensors for attaining that percept. At instantiation, a schema can arrange its logical sensors (or sub-schemas) into a preference ordering based on the needs of the motor schema and the state of the environment.

If the preferred sensor can be allocated to each perceptual schema, then there is no contention. Otherwise, one or more perceptual schemas must use a less desirable (but adequate) sensor. This means the original schedule must be repaired. If the preference is not significant, the scheduler behaves conservatively; it minimizes disturbances to the existing allocation.

The repair strategy is iterative, where each iteration is an attempt to get a better solution (Zweben *et al.* 1994). First, the scheduler uses the MIN-CONFLICT algorithm (Johnson & Minton 1994) to select an alternative resource from the schema making the most recent request. The scheduler simply considers the resources on the list, in order of preference. This is quick, does not disturb the other schemas, and does not need to consider time projections. If MIN-CONFLICT fails, the scheduler can then iteratively relax the restriction that only the latest request be examined for alternatives. Now the preference ordering of alternatives from other perceptual schemas are considered in order of request, and substitutions are made that minimize some global measure of disturbance. The MIN-CONFLICT will find a suitable schedule or declare the plan failed; however, the schedule may be further refined. Therefore, if time permits, the second step is for the scheduler to consider any available time projections. Essentially it decides if the MIN-CONFLICT solution is best for the projected demands. If not, the scheduler repairs (improves) the schedule with some other repair algorithm. This iteration imparts an anytime flavor to the scheduler.

### Ramifications for Acting

Sensing resource allocation is extremely important to scheduling or planning actions. First, acting reliably is not possible without reliable perception. Second, the robot must wait for contention for resources to be resolved. The robot cannot execute a behavior without the perceptual component instantiated. There-

fore, certain actions have to wait until contention is resolved. This raises the issue of how and when to interrupt other behaviors which cannot continue without the missing behavior. Fortunately, the proposed iterative repair strategy is an anytime algorithm. By using it, the worst that will happen is that the robot thrashes because it resolves contention only temporarily. Third, the possibility of thrashing emphasizes the need for meta-behaviors which allow the robot to predict sequences of behaviors (and associated perceptual demands) and remember the duration of each for specific applications.

### Summary and Future Work

A task planner produces a plan consisting of one or more concurrent behaviors to accomplish a set of tasks. Each behavior needs access to sensing resources. As a result, contention for sensing resources may arise. We are developing a sensing manager which uses intelligent scheduling to satisfy the sensing demands resulting from a plan. The intelligent scheduler must operate in the presence of incomplete information, both about what the projected sequence of sensing demands are and the duration of those demands.

Our assumptions about the structure of behaviors and the characterization of the scheduling problem suggest that an iterative repair strategy is appropriate. The strategy first uses a MIN-CONFLICT algorithm to attempt to locally repair the conflict, then iteratively relaxes the constraints until a suitable schedule is found. If time permits, the schedule is then refined using time projections, if available. The strategy is expected to be quick, reduce thrashing, and work reasonably well under the spectrum of incomplete information.

We are currently building a simulator to test our iterative repair strategy, and then will transfer the algorithms to an implementation on one of our mobile robots.

### Acknowledgments

This work is being supported in part by NSF Grant IRI-932038, ARPA Grant AO#B460, and NASA/JSC contract NAS.9-19040.

### References

- Arbib, M. 1981. Perceptual structures and distributed motor control. In Brooks, E., ed., *Handbook of Physiology- The Nervous System II*. Bethesda, MD: American Physiological Society. 1449- 1465.
- Chen, C., and Trivedi, M. 1995. Task planning and action coordination in integrated sensor-based robots. *IEEE Transactions on Systems, Man, and Cybernetics* 25(4):569- 591.
- Fox, M. S. 1994. Isis: A retrospective. In Zweben, M., and Fox, M., eds., *Intelligent Scheduling*. San Francisco: Morgan Kaufmann Publishers. 3- 28.

- Henderson, T., and Shilcrat, E. 1984. Logical sensor systems. *Journal of Robotic Systems* 1(2):169-193.
- Johnson, M., and Minton, S. 1994. Analyzing a heuristic strategy for constraint-satisfaction and scheduling. In Zweben, M., and Fox, M., eds., *Intelligent Scheduling*. San Francisco: Morgan Kaufmann Publishers. 241-256.
- Kaebling, L. P. 1987. An architecture for intelligent reactive systems. In Georgeff, M. P., and Lansky, A. L., eds., *Reasoning About Actions and Plans*. San Francisco: Morgan Kaufmann Publishers. 395-410.
- Musliner, D.; Durfee, E.; and Shin, K. 1993. Circa: A cooperative intelligent real-time control architecture. *IEEE Transactions on Systems, Man, and Cybernetics* 23(6):1561-1573.
- Noreils, F. R., and Chatila, R. 1995. Plan execution monitoring and control architecture for mobile robots. *IEEE Transactions on Robotics and Automation* 11(2):255-266.
- Simmons, R.; Lin, L.; and Fedor, C. 1990. Autonomous task control for mobile robots. In *Proceedings of the 5th IEEE International Symposium on Intelligent Control*, 663-668.
- Zweben, M.; Daun, B.; Davis, E.; and Deale, M. 1994. Scheduling and rescheduling with iterative repair. In Zweben, M., and Fox, M., eds., *Intelligent Scheduling*. San Francisco: Morgan Kaufmann Publishers. 241-256.