Guaranteeing Safety in Spatially Situated Agents

Robert C. Kohout and **James A. Hendler**
Department of Computer Science and
Institute for Advanced Computer Studies
University of Maryland
College Park, MD 20742
kohout,hendler@cs.umd.edu
phone: (301) 405-7027
fax: (301) 405-6707

David J. Musliner
Honeywell Technology Center
MN65-2200
3660 Technology Drive
Minneapolis, MN 55418
musliner@src.honeywell.com
phone: (612) 951-7599

Abstract

"Mission-critical" systems, which include such diverse applications as nuclear power plant controllers, "fly-by-wire" airplanes, medical care and monitoring systems, and autonomous mobile vehicles, are characterized by the fact that system failure is potentially catastrophic. The high cost of failure justifies the expenditure of considerable effort at design-time in order to guarantee the correctness of system behavior. This paper examines the problem of guaranteeing safety in a well studied class of robot motion problems known as the "asteroid avoidance problem." We establish necessary and sufficient conditions for ensuring safety in the simple version of this problem which occurs most frequently in the literature, as well as sufficient conditions for a more general and realistic case. In doing so, we establish functional relationships between the number, size and speed of obstacles, the robot's maximum speed and the conditions which must be maintained in order to ensure safety.

Introduction

Applications in which the failure of a system to perform correctly can result in catastrophe are known as *mission-critical* systems. The reliability requirements of such applications, which include nuclear power plant controllers, "fly-by-wire" airplanes, medical care and monitoring systems, and autonomous mobile vehicles, have motivated extensive research into the development of highly reliable software systems. Research into the development of systems-level support for mission-critical systems focuses upon "hard" real-time operating systems, which can *guarantee* that the system can deliver resources stipulated by some externally generated set of timing constraints. Similarly, the programming language community has developed technologies to ensure that programs will *always* behave correctly, with respect to some externally provided performance specification.

In contrast to the effort in the systems and programming languages communities, there is not a large body of research into the problem of *generating the specifications* which will ensure the correct and timely operation of a deployed mission-critical system. Determining a correct plan of action is the focus of AI Planning research. However, the high-variance time requirements of current techniques make it difficult to guarantee that they will produce a correct solution in time to actually use it. CIRCA (Musliner, Durfee, & Shin 1995) was developed to address this problem: by modeling the world as a finite set of situation-states, with well defined transitions between them, CIRCA is able to search the situation space "offline" (i.e. before the system is actually deployed), in an effort to find a closed set of safe states such that, for any possible combination of external events, it will always be possible for the control system to take an action that will keep the current situation-state within the closed set of safe states. When the situation space includes continuous dimensions, this technique can only be used if we can somehow discretize the continuous space. When the dimension is time, it is usually straightforward to meaningfully distinguish between times before and after a deadline, as well as some small set of "deadline approaching" intervals. However, when the dimensions are spatial, there is often no simple partitioning which will allow us to reason about a finite set of discrete states. This paper examines the problem of guaranteeing safety in a well-studied class of robot motion problems. By establishing sufficient conditions for ensuring safety, we provide the basis for automatic reasoning about maintaining safety in spatial domains.

The "Asteroid Avoidance Problem"

Consider one of the simplest natural problems in dynamic motion planning: how can we find a path for a robot, R, which travels from some initial location l_0 to some goal location l_G, while avoiding each of n obstacles, $O_1, \ldots, O_n$, where each of the O_i is moving at a known, constant velocity? We are making three simplifying assumptions that would rarely occur in a real-world application:

1. The trajectories of the obstacles are *known* to the system in advance.
2. The obstacles move linearly.

3. The speed of the obstacles is fixed.

(Reif & Sharir 1985) named this class of problems the *asteroid avoidance problem*, and they showed that, for the three-dimensional case, the problem is $\mathcal{PSPACE}$-hard when the velocity of the robot is bounded, and $\mathcal{NP}$-hard even when the robot's velocity is unbounded. (Canny & Reif 1987) showed that the 2-dimensional case is $\mathcal{NP}$-hard. This has strong implications for any mission-critical application, such as a fly-by-wire or autonomous vehicle system, for which the failure to avoid obstacles could be catastrophic: *in the general case*, we will only be able to find timely results for problems of very small size, even under the strong simplifying assumptions listed above. Additionally, one should note that these results are for the problem of finding a path *if it exists*; there is no guarantee that such a path will in fact exist.

This paper focuses upon finding restrictions to the problem for which we can guarantee that some safety-preserving path exists, and for which the problem of computing the solution is tractable. For instance (Fujimura & Samet 1989) give an $O(n^2 logn)$ algorithm for solving the asteroid avoidance problem, under the assumption that the robot can move faster than all of the obstacles. (Reif & Sharir 1985) claim that under such an assumption, it is always possible to find such a path, *as long as the initial position of the robot is not in the "shadow" of any obstacle*, where the shadow of an obstacle is defined to be all those locations from which escape from that obstacle is impossible. The proofs that 1) a safety-preserving path always exists, and that 2) we have a relatively efficient way of finding it, are the two most important criteria for guaranteeing that under these conditions obstacles will always be avoided. We shall refer to these as the *existence* and *ability* criteria, respectively.

If we have the luxury of knowing well in advance what the initial positions and trajectories of the robot and obstacles are, so that we can compute the solution "offline", and use it at the appropriate time, the existence and ability criteria are all we need to satisfy to ensure the robot's safety. However, this situation rarely occurs in practice. Normally, the relevant data become available at some time, t_0, and we must have the appropriate solution by some later deadline, t_d, or the solution will be obsolete by the time we begin to execute it. We call the requirement that a solution be produced before it is obsolete the *timeliness* criteria. In the case where the speed of a robot is greater than that of any object, we can establish timeliness by expanding the shadow of each obstacle to account for the (worst-case) time required to compute a solution to the problem.

In this section, we will consider the following variant of the asteroid problem: we have a single robot, R, and a set of n obstacles $O_1, \ldots, O_n$ moving at constant speeds $v_1, \ldots, v_n$ along linear trajectories. The robot is capable of instantaneous, unbounded acceleration, up to some maximum speed V_r. Under what conditions can we guarantee that 1) a safety-preserving path exists which will allow the robot to avoid being hit by any obstacle and 2) we can compute the path in time to execute it? We shall model the robot as a single point and the obstacles as circles with diameters $d_1, \ldots, d_n$. Most path planning literature assumes we can bound the robot and obstacles by polygons. Reducing the robot to a point is a standard technique introduced in (Lozano-Pérez & Wesley 1979) : it turns out that a solution in the case where the robot is a convex polygon is equivalent to the case where the robot is a point and the sizes of the obstacles have been increased by the size of the robot. However, this technique can only be used when the obstacles do not rotate. Since a polygon that does not rotate can be bounded by a circle, and a polygon that does rotate can also be bounded by a (possibly larger) circle, centered at the center of rotation, we use circles to represent obstacles in order to simplify our presentation, and actually gain some generality. In this paper, we are concerned only with avoiding the obstacles: there is no goal position to which we are trying to move the robot. Finally, though the concepts we present do generalize to higher dimensions, we will limit our treatment to the two-dimensional case for ease of presentation.

The Threat Horizon

The central insight of this section is that, once we fix the number, speed and sizes of the obstacles, and the maximum speed of the robot, the obstacles can always be avoided, so long as they are each initially some minimum distance away from the robot. We call this distance the *threat horizon*, H. It should be obvious that, if we make H extremely large relative to the speeds of the obstacles, some safety-preserving path must exist. However, we would like to make H as small as possible. We also need to satisfy the ability criteria, i.e., it is not enough to know a path exists, we must be able to find it. We address these issues in the proof of the following theorem:

Theorem 1 *Let R be a point in a 2-dimensional Euclidean plane, which represents the location of a robot at time t_0. Assume that the robot can rotate and accelerate instantaneously, but is limited by a maximum speed V_r. Let $O_1, \ldots, O_n$ be a set of n circular obstacles with diameters $d_1, \ldots, d_n$ which move at known, constant velocities $v_1, \ldots, v_n$. Let V_o be the largest of the v_i. Let W be the sum of the widths of the obstacles, i.e., $W = \sum_{i=1}^{n} d_i$. If each of the obstacles is initially a distance greater than*

$$\frac{W(V_o + V_r)}{2V_r}$$

from R, then there exists a "safe harbor" point S such that none of the O_i will touch S at any time, and the robot can move from R to S without intersecting any of the O_i.

Proof: Let Φ_i be the space occupied by obstacle O_i, from time t_0 to t_∞. Since the obstacles move along linear paths, Φ_i is comprised of all of the space between two parallel rays, separated by a width of d_i. Since the space which lies between two parallel lines has been named a *plank*, we shall call this region a *half-plank* of width d_i.

If each of the obstacles begins at a distance greater than $W(V_o + V_r)/2V_r$ from R, then for each obstacle O_i, there must exist a positive number $\hat{\epsilon}_i$, such that O_i begins *exactly* $W(V_o + V_r)/2V_r + \hat{\epsilon}_i$ from R. Let $\epsilon = 2V_r\hat{\epsilon}_i/(V_o+V_r)$. Then ϵ is positive, $\hat{\epsilon}_i = \epsilon(V_o+V_r)/2V_r$, and O_i begins a distance $(W + \epsilon)(V_o + V_r)/2V_r$ from R.

The earliest time that one of the obstacles could intersect the robot would be in the case that the obstacle O_n for which $\hat{\epsilon}_n$ is the smallest of the $\hat{\epsilon}_i$, travels at speed V_o and heads directly towards R, while the robot travels a straight-line path towards O_n, at its maximum speed V_r. In this case, O_n and the robot would collide at time $t_0 + ((W + \epsilon)(V_o + V_r)/2V_r(V_o + V_r))$, which is just $t_0 + (W + \epsilon)/2V_r$.

Now consider the region which comprises all of the points to which the robot could move by time $t_j > t_0$, while never moving at a speed greater than V_r. This area is just a circle, with radius $V_r(t_j - t_0)$. It follows that the area which comprises the locations to which the robot could travel before it could possibly be hit is a circle centered at R, with radius $V_r(t_0 + (W + \epsilon)/2V_r - t_0)$, which simplifies to $(W + \epsilon)/2$. We shall call the distance $(W + \epsilon)/2$ the *safety radius*, and the circle of this radius centered at R the *safety region*.

Now we need to show that the n half-planks, $\Phi_1, \ldots, \Phi_n$, cannot completely cover the safety region. To do this, we use the 2-dimensional version of Bang's solution to Tarski's "plank problem" (Bang 1951), which states[1]:

Theorem 2 (Bang) *If L is a convex body of minimal width l in a 2-dimensional Euclidean plane, and L is contained in the union of p planks of widths $h_1, \ldots, h_p$, then $h_1 + \cdots + h_p \geq l$.*

Clearly, the set of objects which can be covered by planks is a superset of the set of objects which can covered by half-planks. Since the safety region is a convex body of width $W + 2\epsilon$, by Bang's theorem, in order for the n half-planks to cover the safety region, $\sum_{i=1}^{n} d_i$ must be greater than or equal to $W + 2\epsilon$. But, by definition, $W = \sum_{i=1}^{n} d_i$, and ϵ is positive, so there must be some area within the safety region which is not covered by the Φ_i. This proves the existence of S. To see that the robot can move from R to S without being hit, one only need remember that the safety radius was defined so that it is possible for the robot to move anywhere in the safety region by the time the first obstacle reaches its perimeter. **Q.E.D.**

[1]This theorem generalizes to n-dimensions.

This proof satisfies the existence criteria. In order to compute the solution, we need to compute the intersection of the n half-planks with the safety region. If k is the number of intersections of the half-planks, this can be done in $O((k + n)logn)$ using a modification of (Bentley & Ottmann 1979) algorithm for reporting the intersection of line segments, as elaborated in (Melhorn 1984, pp. 154–160). (Chazelle & Edelsbrunner 1992) describe an $O(nlogn+k)$ algorithm which could also be modified to find safe harbors within the safety region. Either of these solutions satisfies the ability criterion.

In order to establish timeliness, for a fixed number of obstacles n, we need to know the *actual* worst-case time required to compute the solution. Call this time t_s. If we increase H by the maximum distance an obstacle can travel in t_s, then we ensure that the system will have sufficient time to compute and execute the solution. Thus, the threat horizon, H should be

$$V_o * t_s + W(V_o + V_r)/2V_r$$

in order to guarantee timeliness. Note that we can use a similar argument to account for robot rotation and acceleration times, in the more realistic cases where acceleration and rotation are not assumed to be instantaneous.

The Necessity of the Threat Horizon

In the previous subsection, we showed that constraining obstacles to begin their travels outside of the threat horizon H was sufficient to ensure the safety of the robot. In this subsection, we show that the bound is tight. Note that the size of the safety region depends only upon the sizes of the obstacles, while H also depends upon the ratio of V_o to V_r. This leads to the following theorem:

Theorem 3 *Let R be a point in a 2-dimensional Euclidean plane, which represents the location of a robot at time t_0. Assume that the robot can rotate and accelerate instantaneously, but is capable of only a maximum speed V_r. Let $O_1, \ldots, O_n$ be a set of n circular obstacles with diameters $d_1, \ldots, d_n$, which move at known, constant velocities $v_1, \ldots, v_n$. Let $W = \sum_{i=1}^{n} d_i$, and let δ be a positive constant, such that $\delta > 2WV_r/V_o$. If at time t_0 obstacles are allowed to start a distance*

$$(W - \delta)(V_o + V_r)/2V_r$$

from the robot, then there exist configurations for which it is not possible for the robot to avoid collision.

Proof: It suffices to show a single configuration for which it is not possible to avoid the obstacles. Let all the obstacles O_i have the same diameter D. Assume that the O_i start in the configuration depicted in Figure 1, where the obstacles are just touching (i.e. for $i < n$ the distance from the center of O_i to the center of O_{i+1} is D), and they are all traveling at speed V_o, along parallel courses as indicted in the figure. Let l_t

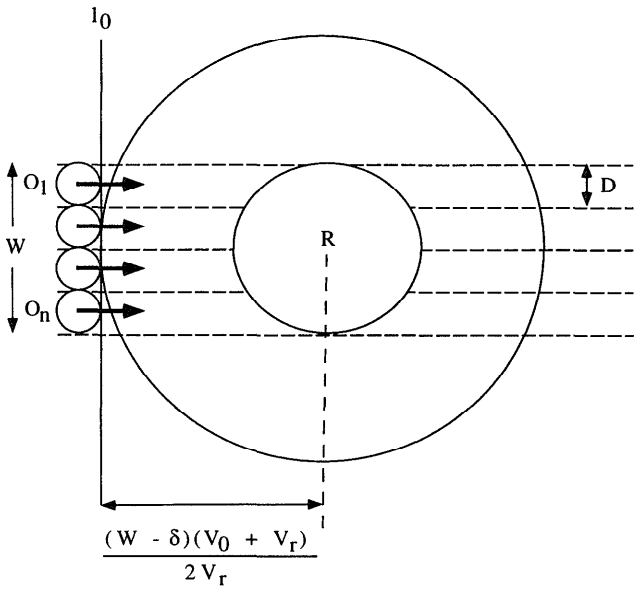

Figure 1: Necessity of the Threat Horizon

be the line which is tangent to all of the obstacles at time t, and which is on the same side of the obstacles as it is at time t_0, when it is on the side of the obstacles nearest the robot. Let the distance from R to the line l_0 be $(W-\delta)(V_o+V_r)/2V_r$. Clearly, all of the obstacles are initially at least this distance from R at time t_0. The robot can travel the distance $(W-\delta)/2$ in time $(W-\delta)/2V_r$. When it has done so, the line $l_{(W-\delta)/2V_r}$ will be a distance $(W-\delta)/2$ away from the (original) point R. If the obstacles can travel the distance W before the robot can move $\delta/2$, then they will completely traverse the circle of radius $W/2$ centered at R before the robot is able to have moved outside of this circle. That is, if $W/V_o < \delta/2V_r$, the robot will be hit by at least one obstacle. Since, by definition $\delta > 2WV_r/V_o$, this completes the proof.

In cases where V_o is large relative to W and V_r, δ approaches arbitrarily close to 0. Consequently, $H = W(V_o+V_r)/2V_r$ is the minimum distance for which we can guarantee that a safety-preserving path exists, in the general case.

The Dynamic Asteroid Avoidance Problem

In the previous section, we presented a necessary and sufficient criterion for guaranteeing safety in the asteroid avoidance problem. In doing so, we have established a functional relationship between the number, size and speed of the obstacles, the maximum speed of the robot and the distance which obstacles must initially be from a robot in order to ensure that the robot will never collide with any of the obstacles. We have also shown that, in those cases where we can guarantee safety, there is a simple and efficient means of finding the safety-preserving path. The problem we have addressed thus far makes the same simplifying assumption as in made in, e.g., (Fujimura & Samet 1989; Kant & Zucker 1986; Reif & Sharir 1985) : the position and the trajectories of the obstacles are known *prior* to execution time. While this formulation has proven challenging, it is overly optimistic. Normally, we can expect the existence, location and trajectories of obstacles to become known during execution, perhaps while the robot is already in the process of avoiding previously detected obstacles. In this section, we examine the problem of guaranteeing safety when the location of obstacles must be sensed at execution time, which we have named the "dynamic" asteroid problem. Using Theorem 1, we develop a sufficient condition for ensuring the existence of a safety-preserving path in this problem.

Obstacles with Uniform Velocity

Consider the asteroids problem described above, where we know there are at most n circular obstacles $O_1, \ldots, O_n$, traveling along linear trajectories at constant speeds. For simplicity, we will assume that all of the obstacles are of the same diameter, D. In addition, assume that all of the O_i move at the same speed, V_o. Unlike the previous section, we do not assume that we know the location of the obstacles in advance. Instead, the obstacles are allowed to appear, one or more at a time, up to a maximum of n obstacles. We wish to determine a safety horizon, H, such that we know that a safety-preserving path exists as long as all of the obstacles initially appear at a distance of at least $\bar{H}$ from the robot. The following is a corollary of Theorem 1 above:

Corollary 1 *Let H be $W(V_o+V_r)/2V_r$, where $W = nD$, V_r is the maximum speed of the robot, and V_o is the (uniform) speed of the obstacles. If each of the obstacles O_i appears at some time $t_{a(i)} \geq t_0$, at a distance greater than $\bar{H} = nH + W$ from the position of the robot at that time, then there exists a collision-free path from the starting point R to some point S such that none of the O_i will touch S at any time.*

Proof (by induction on m, the number of obstacles which have already appeared[2]): The base case is handled by Theorem 1, since $nH + W \geq H$, and we can assume that $t_{a(1)} = t_0$.

Assume, by induction, that after the first $m < n$ of the obstacles have appeared, there exists a safety preserving path to some point S_m which is safe from all of the obstacles $O_1, \ldots, O_m$. Let R_x be the location of the robot at time t_x. If at time $t_{a(m+1)}$ all of the obstacles $O_1, \ldots, O_m$ are farther than H from $R_{a(m+1)}$, then

[2]This is not induction on n, the maximum number of obstacles which can appear.

once again Theorem 1 holds, and a safety preserving path exists from $R_{a(m+1)}$ to some point S_{m+1}. If one (or more) of the O_i is within H of the robot, then the robot can wait at S_m until those obstacles have traveled at least a distance H from S_m. In the worst case, this time is $(H + D)/V_o$. Of course, another obstacle could move to within H of S_m in this time. Since there are only $m < n$ obstacles, the longest we would have to wait would be $m(H + D)/V_o$ before we can be assured that all of the obstacles are at least H from S_m. In this worst-case, the most recent obstacle, O_{m+1} will still be a distance greater than $(nH+W)-m(H+D) = n(H + D) - m(H + D) = (n - m)(H + D)$ from S_m, and since $m < n$, we know that this distance is greater than H. Thus we know that at some time, all of the $m + 1$ obstacles will be outside of the safety region, and so Theorem 1 applies. Thus there exists a (linear) path from S_m to some new point S_{m+1}. **Q.E.D.**

Since by definition $W = nD$, it follows that $\bar{H} = n^2D((V_o + V_r)/V_r + 1)$, and thus this corollary gives an $O(n^2)$ upper bound for the threat horizon in the dynamic asteroids problem. It is also easy to see that $\bar{H}$ is sufficient to guarantee safety so long as there are never more than n obstacles within H of the robot R at any single time, even if many more than n obstacles appear over time.

Allowing the speed of Obstacles to Range from V_l to V_o

There is a straightforward generalization of the above theorem in cases where the obstacles are constrained to have constant positive velocities ranging from a lower bound of V_l to a maximum speed of V_o.

Corollary 2 *Let $O_1, \ldots, O_n$ be n circular obstacles of fixed diameter D, each constrained to move at a constant velocity, V_i, such that $\forall i\ V_l \leq V_i \leq V_o$, where V_l and V_o are fixed positive constants. Let $W = nD$, and H be $W(V_o + V_r)/2V_r$, where V_r is the maximum speed of the robot. If each of the obstacles O_i appears at some time $t_{a(i)} \geq t_0$, at a distance greater than*

$$\bar{H} = \frac{V_o(nH + W)}{V_l}$$

from the position of the robot at that time, then there exists a collision-free path from the starting point R to some point S such that none of the O_i will touch S at any time.

Proof (by induction on m, the number of obstacles which have already appeared): The base case is again handled by Theorem 1, since that theorem applies whenever obstacle velocities are constrained by some maximum, V_o, and obstacles occur outside the threat horizon H. Since $V_o \geq V_l$, it follows that, for all values of n, $V_o n/V_l \geq 1$, and thus $\bar{H} > H$.

The inductive step is very similar to that for Corollary 1: Assume, by induction, that after the first $m < n$ of the obstacles have appeared, there exists a safety preserving path to some point S_m which is safe from all of the obstacles $O_1, \ldots, O_m$. Let R_x be the location of the robot at time t_x. If at time $t_{a(m+1)}$ all of the obstacles $O_1, \ldots, O_m$ are farther than H from $R_{a(m+1)}$, then once again Theorem 1 holds, and a safety preserving path exists from $R_{a(m+1)}$ to some point S_{m+1}. If one (or more) of the O_i is within H of the robot, then the robot can wait at S_m until those obstacles have traveled at least a distance H from S_m.

In the following, we have to change the earlier proof to account for the fact that slower moving obstacles may remain near the robot for longer periods of time: In the worst case, this time is $(H + D)/V_l$ (note that $(H + D)/V_l \leq (H + D)/V_o$, which was the worst case in the previous proof). Again, another obstacle could move to within H of S_m in this time. Since there are only $m < n$ obstacles, the longest we would have to wait would be $m(H + D)/V_l$ before we can be assured that all of the obstacles are at least H from S_m. In this worst-case, the most recent obstacle, O_{m+1} can travel a distance of up to

$$V_o * m(H + D)/V_l$$

(if it happens to have the maximum speed, V_o). In any case, then, this will still be a distance greater than $V_o * (nH+W)/V_l - V_o * m(H+D)/V_l = (n-m)V_o(H+D)/V_l$ from S_m, and since $n > m$, and $V_o \geq V_l$, we know that this distance is greater than H, and Theorem 1 applies. Thus there exists a (linear) path from S_m to some new point S_{m+1}, which is safe with respect to all of the currently visible obstacles. **Q.E.D.**

Note that this threat horizon grows with the ratio of the maximum speed to the minimum speed of obstacles, V_o/V_l. If this number is large, the horizon becomes prohibitive. Intuitively, slower moving obstacles should be easier to avoid, but in the context of this result, allowing the speed of obstacles to approach zero will make the threat horizon approach infinity. This is a clear indication that the bound is not tight. Even in the case where $V_o = V_l$, we choose H to account for the case when all n obstacles are a distance H from the obstacle, but we choose $\bar{H} = nH + W$ to account for the times when the n obstacles are "evenly" spaced. But when all the O_i are visible H alone is sufficient, and when the obstacles are evenly spaced, so that exactly one is within H or the robot at any given time, then a horizon of $H' = \bar{H}/n = H + D$ will suffice. With this insight, it is possible to reduce $\bar{H}$ by a factor of 4, but the resulting horizon is still $O(n^2)$. We are currently trying to prove the conjecture that a threat horizon which is linear in the number of obstacles exists.

Goals of Achievement

In mission-critical applications, we can distinguish two components to the planning problem:

1. **Achieving Goals** - in most applications, the agent will be charged with satisfying (or perhaps optimizing) some goal function, where the successful

achievement of a goal has some positive utility, and the failure to achieve a goal is not considered catastrophic (i.e. the utility in negative, but small compared to the cost incurred by a failure to maintain safety).

2. **Maintaining Safety** - avoiding catastrophic failure is the primary consideration of MC control systems. For our purposes, all catastrophes are equivalent, in the sense that none is considered more or less desirable than any other. Since the cost of failure is extremely high, we seek problem solutions which will guarantee that the agent remains appropriately distant from any and all threats.

In these domains, the constraints imposed by the second component absolutely dominate the influences of the first. So, for example, a mission-critical system will not attempt to achieve a non-critical goal, no matter what its utility is, unless it can assure itself that it is possible to maintain safety. This dominance allows us to effectively decouple the two components, and to consider the problem of maintaining safety independently of the influences of goals-of-achievement. This has allowed us to develop and implement a simulation in which a robot can achieve goals while avoiding moving obstacles. (Reif & Sharir 1985) present a search-based solution to the asteroids problem which is exponential in the number of moving obstacles. The high-variance time requirements of this algorithm make it unsuitable for ensuring that obstacles are avoided, but we are using a version of it to determine non-critical, safety-preserving paths to goals, while using a much faster algorithm based upon the results above to ensure safety. Using the Maruti hard real-time operating system (Saksena, da Silva, & Agrawala 1993), we are able to guarantee processing time to the safety-critical routines, and allow the search to use the processor time that is left over when the critical routines have finished. If the search algorithm is able to find a path to a goal quickly, then the system can use it without compromising safety. Otherwise, the lower-level competences for finding and reaching a safe-harbor will ensure that the robot remains safe while it searches for a way to achieve its goals.

Conclusions

It is easy to see the limitations of this work in its current form. However, without similar, albeit more comprehensive, results, we cannot deploy mission-critical systems in spatially-situated domains. The time and expense involved in developing hard real-time operating systems running provably correct software is wasted if the system specifications are not sufficient to ensure that catastrophe will be avoided. Presumably, the application domains will have sufficient restrictions and regularities to allow the development of provably correct behavioral competences. We believe the techniques introduced in this paper provide a basis for reasoning about safety maintenance in spatial domains in particular, and continuous domains in general.

Acknowledgements

This research was supported in part by grants from NSF(IRI-9306580), ONR (N00014-J-91-1451), AFOSR (F49620-93-1-0065), the ARPA/Rome Laboratory Planning Initiative (F30602-93-C-0039), the ARPA I3 Initiative (N00014-94-10907) and ARPA contract DAST-95-C0037. Dr. Hendler is also affiliated with the UM Institute for Systems Research (NSF Grant NSF EEC 94-02384). Thanks to V.S. Subrahmanian for additional support.

References

Bang, T. 1951. A solution of the "plank problem". In *Proceedings of the American Mathematical Society*, volume 2, 990–993.

Bentley, J. L., and Ottmann, T. A. 1979. Algorithms for reporting and counting geometric intersections. *IEEE Transactions on Computers* C-28(9):643–647.

Canny, J., and Reif, J. 1987. New lower bound techniques for robot motion planning problems. In *Proceedings of the 28th IEEE Symposium of Foundations of Computer Science*, 49–60.

Chazelle, B., and Edelsbrunner, H. 1992. An optimal algorithm for intersecting line segments in the plane. *Journal of the Association for Computing Machinery* 39(1):1 – 54.

Fujimura, K., and Samet, H. 1989. A hierarchical strategy for path planning among moving obstacles. *IEEE Transactions on Robots and Automation* 5(1):61–69.

Kant, K., and Zucker, S. 1986. Towards efficient trajectory planning: Path velocity decomposition. *International Journal of Robotics Research* 5:72–89.

Lozano-Pérez, T., and Wesley, M. 1979. An algorithm for planning collision-free paths among polyhedral obstacles. *Communications of the ACM* 22(10):560–570.

Melhorn, K. 1984. *Multi-dimensional Searching and Computational Geometry*, volume 3. New York, Berlin, Heidelberg: Springer-Verlag.

Musliner, D. J.; Durfee, E. H.; and Shin, K. G. 1995. World modeling for the dynamic construction of real-time control plans. *Artificial Intelligence* 74(1):83–127.

Reif, J., and Sharir, M. 1985. Motion planning in the presense of moving obstacles. In *Proceedings of the 25th IEEE Symposium of Foundations of Computer Science*, 144–154.

Saksena, M.; da Silva, J.; and Agrawala, A. 1993. Design and implementation of Maruti. Technical Report CS-TR-3181, University of Maryland Department of Computer Science.