Deadlock-free traffic control with geometrical critical sections

Albert L. Schoute Peter J. Bouwens

Department of Computer Science
University of Twente
P.O.box 217, 7500 AE Enschede
E-mail: schoute@cs.utwente.nl

Abstract

Traffic control of vehicles on pre-planned paths may be based on critical sections where vehicles have to control their velocity mutually in order to avoid collisions. By analysing the geometry of critical sections carefully deadlock situations can be recognised and situations where vehicles may share a critical area under certain restrictions to speed-up their passage. Deadlock-freeness can be guaranteed by applying a variant of the bankers algorithm. The traffic control strategy is applied within the ‘Mobile Autonomous Robot Twente’-project in which mobile assembly robots will drive around in a ‘factory of the future’ guided by software control.

1 Introduction

Traffic control aims at guiding mobile objects towards their goals as fast as possible while avoiding mutual collisions between these objects. Collision avoidance requires either adjustment of the velocity (planning in time) or adjustment of the followed path (planning in space), or a combination of both. Planning in space and time simultaneously leads to the best solution but is also the most demanding with respect to processing power. Planning over a large space-time scope and with a large number of objects not only becomes intractable but also useless in view of uncertainty factors in real situations. Therefore planning is solved in practice often by choosing appropriate decompositions of the problem space.

In case of traffic control of many moving objects typically a path-velocity decomposition [Kant86] is applied on a global scale (i.e. a large space-time scope), while on a local scale small adjustments are still allowed to optimise the combined path-velocity profile of each object. The path-velocity decomposition means that trajectories of moving objects are determined by only considering stationary objects, which leads to a kind of global road map that remains fixed over time. Velocity is planned along the paths to avoid collisions between moving objects. Note that velocity planning at a global scale may consist of a simple "stop or go" scheme. More subtle velocity adjustment may take place on a local scale.

In this article we consider traffic control of vehicles that travel along prescribed paths. Given these paths critical sections can be recognised where vehicles may collide and have to control their velocity mutually to avoid such collisions. Well known complications with the exclusive occupation of critical sections such as deadlock have to be faced. As we will see deadlock causes may be quite subtle in a geometrical context. Although a huge amount of literature

Proceedings SION Conference Computing Science in the Netherlands, CSN94, 21-22 nov 1994, Utrecht
Ed. E. Backer, Published by Stichting Mathematisch Centrum, Amsterdam 1994, ISBN 90 6196 449 0

exists on robot path planning, traffic control based on critical sections within a fixed road map has to our knowledge not yet received much attention.

The traffic control strategy described has been inspired by application to mobile robots. In the MART project (Mobile Autonomous Robot Twente) [Tillema93] a mobile robot is being developed that - in the factory of the future - picks up components at part supply stations and assembles products while it is driving from one station to the other. In order to travel on the work floor, the mobile robots follow preplanned routes like roads in a city. In a city however, where roads are static, traffic lights or priority rules are used to control the traffic and to avoid collisions on for example crossings. In the factory of the future, the production must be flexible in order to make say mixers in one week and shavers in another week. Therefore, routes may only be fixed for relative short periods of time. No static physical provisions like rails or magnetic strips are acceptable. The road map on the work floor will be determined by software such that it can be adapted easily. Given the actual road map all critical sections have to be identified. To prevent mobile robots from colliding with each other, mobile robots have to claim and release critical sections by means of a central traffic controller process. The traffic controller will apply a dynamic algorithm to avoid deadlock.

The outline of this article is as follows. In section 2 collision situations are analysed. A deadlock avoidance strategy (bankers algorithm) is considered in section 3 within the context of geometrical critical sections. In section 4 more details are presented of the application to the MART project.

2 Analysis of mutual collision area's

Consider vehicle V_i travelling along some path P_i . A path determines the vehicles position vector according to a path parameter s. Let the position vector be given by $(x(s), y(s), \alpha(s))$ representing respectively the x, y-co-ordinates of some reference point of the vehicle and the orientation α of the vehicle, i.e. the rotation of some reference radial.

How can we characterise the occurrence of collisions between vehicles ? Although multiple vehicles could collide together, a collision is in principle a mutual event between two vehicles that try to occupy the same space. So we will consider vehicles always in pairs.
Let $L_i(s_i)$ denote the *locus* of vehicle V_i when following path P_i, i.e. the set of points occupied by the vehicle when in state s_i .The locus L_i is determined by the shape of the vehicle which is assumed to remain fixed.
The joint state or *configuration* (s_i, s_j) of two vehicles V_i and V_j is defined as collision state if $L_i(s_i) \cap L_j(s_j) \neq \emptyset$. Note that we generalise the notion of collision to all configurations where $L_i(s_i)$ and $L_j(s_j)$ overlap, irrespective of physical reality. Collision states can be made visible in a 2D configuration diagram with s_i and s_j as independent parameters.

Consider for example the case where P_i and P_j are defined by perpendicularly crossing straight lines on which disk-shaped vehicles with radius R_i and R_j move (see figure 1). Because of the disk-shape the orientation is irrelevant. Collisions occur when the distance between the vehicles centres (that are taken as reference point) is less than the sum of the

vehicles radii. As a result the *collision area* in the configuration diagram is circular with radius $r = R_i + R_j$.
Note that the collision area induces a *critical section* on each path defined by the critical states s_i respectively s_j for which some collision configuration (s_i, s_j) exist. More precise, a critical section $C_i(j)$ on path P_i induced by P_j is defined by:

$$C_i(j) = \{ s_i \mid (s_i, s_j) \text{ is collision configuration for some } s_j \}$$

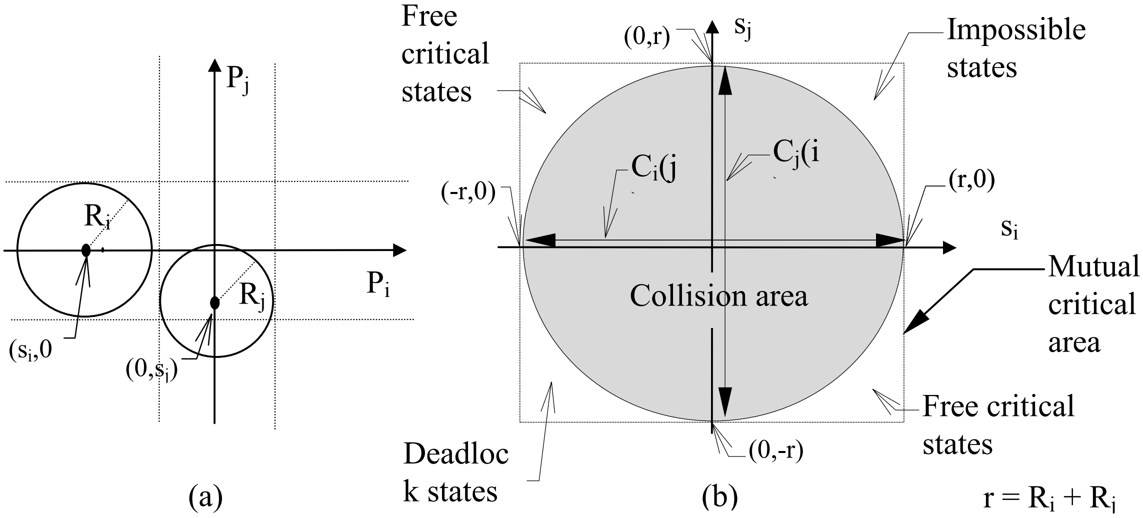


Figure 1. (a) Perpendicular crossing of disk-shaped vehicles
(b) Configuration diagram with mutual critical area $C_i(j) \otimes C_j(i)$

In the configuration diagram a rectangular area can be recognised enclosing the collision area that is given by the Cartesian product $C_i(j) \otimes C_j(i)$. This area is called a *mutual critical area* because it contains the configurations for which the vehicles are in mutually induced critical sections. The mutual critical area may contain collision-free states, for example in figure 1 the states outside the circular collision area. Some of these collision-free states may however lead to a collision-state under certain conditions. Suppose vehicles can only drive forward, which means that their path parameters are non-decreasing. The non-collision states in the lower-left corner of the mutual critical area of figure 1 are indicated as *deadlock states*. Increasing the path parameters s_i or s_j would inevitable lead to collision. It can only be avoided by halting both vehicles. As a result both vehicles are waiting for each other and are thus in a state of deadlock. Note that a vehicle enters the mutual critical area when it traverses a dashed line in figure 1a. It can be seen that both vehicles can enter the critical area without (yet) colliding. Avoidance of collision thus not only means avoiding collision states but also avoiding deadlock states.
Certain critical configurations could be impossible to reach due to physical constraints. In case of forward driving vehicles the upper-right corner of the critical area in figure 1 contains such *impossible non-collision states*. The upper-left corner and the lower-right corner contain configurations that could be reached without problems and are termed *free critical states*. These states correspond to "following situations", where one vehicle is already entering the critical area while the other is about to leave.
A safe strategy for collision avoidance would be to prohibit two vehicles of being in the mutual critical area. However, as will become apparent by other examples, traversing free

critical states could be really advantageous. Unnecessary waiting before entering the critical section may be avoided. In this case we say that vehicles *share* a mutual critical area.
Figure 2 shows a configuration diagram if the paths are directed straight lines that cross with a sharp angle $\gamma < 90°$. Figure 3 shows the case with a wide angle $\gamma > 90°$. Figure 3 can be derived from figure 2 by reversing the direction of one of the vehicles, which means mirroring the configuration diagram with respect to either s_i or s_j .
In case of rectangular vehicles the orientation must be taken into account. It is natural to assume the orientation to be directed along the trajectory, in this case a straight line. If vehicles are able to turn in arbitrary ways, collision analysis can become complicated. A worst-case approach would be to model the vehicle over straight path segments with fixed orientation by a rectangle, but over path segments with changing orientation by an enclosing disk.

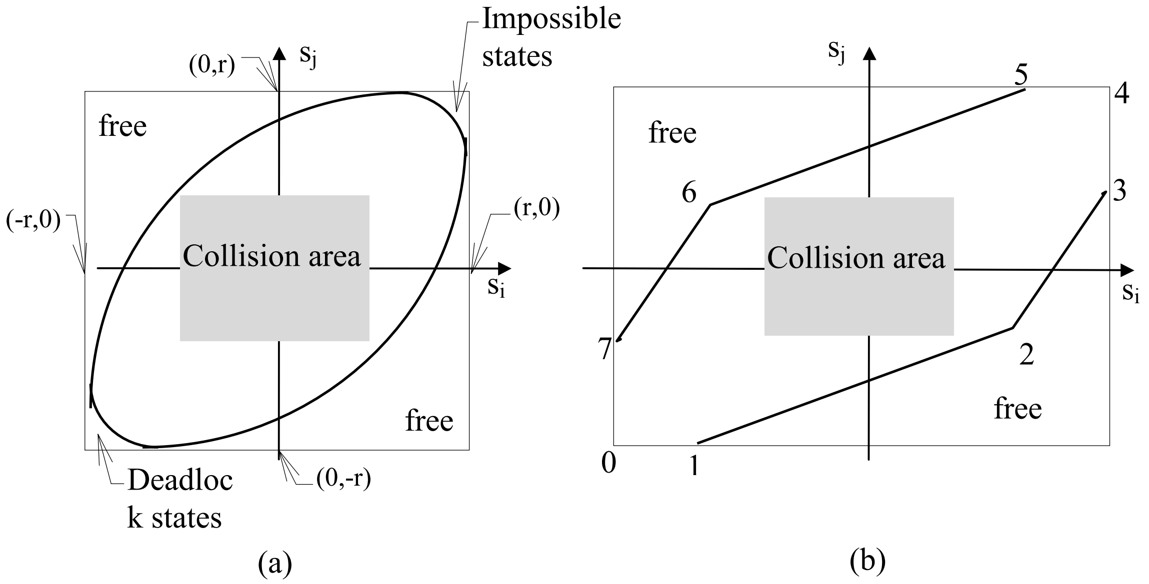
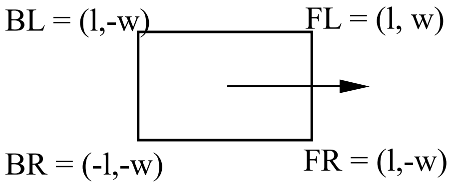


Figure 2.
(a) Configuration diagram of disk shaped vehicles crossing at sharp angle γ with $r = (R_i + R_j) / \sin\gamma$.
(b) Configuration diagram of rectangular vehicles crossing at sharp angle γ. Assume the corners of the vehicles given by:



The marked points on the boundary of the collision area correspond to the cases where corners of vehicles coincide as follows:

0: FR_i meets FL_j if $(s_i,s_j) = (-r_i, -r_j)$
1: BR_i meets FL_j if $(s_i,s_j) = (-r_i + l_i, -r_j)$
2: BR_i meets FR_j if $(s_i,s_j) = (q_i, -q_j)$
3: BL_i meets FR_j if $(s_i,s_j) = (r_i, r_j - l_j)$
4: BL_i meets BR_j if $(s_i,s_j) = (r_i, r_j)$
5: FL_i meets BR_j if $(s_i,s_j) = (r_i - l_i, r_j)$
6: FL_i meets BL_j if $(s_i,s_j) = (-q_i, q_j)$
7: FR_i meets BL_j if $(s_i,s_j) = (-r_i, -r_j + l_j)$

with $r_i = (w_j + w_i \cos\gamma) / \sin\gamma + l_i$, $r_j = (w_i + w_j \cos\gamma) / \sin\gamma + l_j$,
$q_i = (w_j - w_i \cos\gamma) / \sin\gamma + l_i$, $q_j = (w_i - w_j \cos\gamma) / \sin\gamma + l_j$.

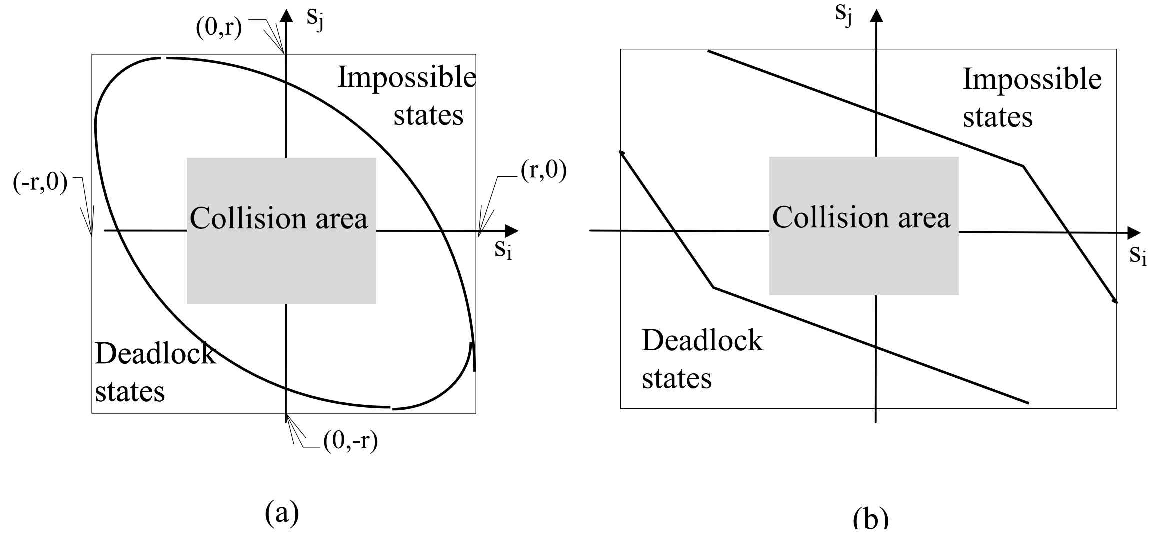
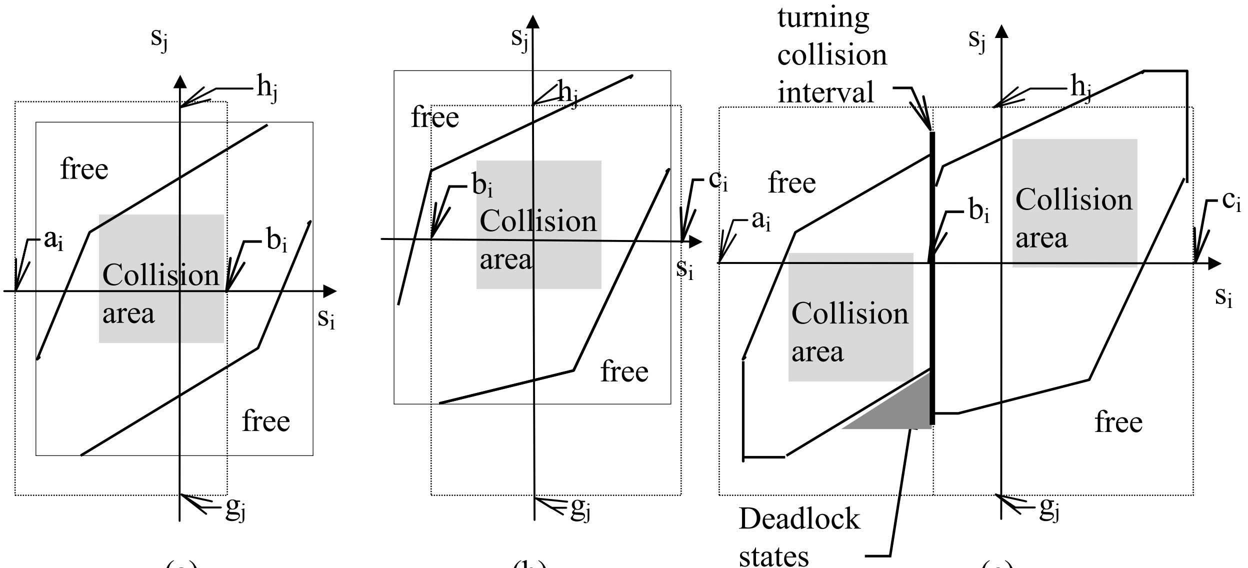


Figure 3. Similar as figure 2 only with wide angle γ .

It should be noted that in case of a sharp angle γ the vehicles move more or less in similar directions, whereas in case of a wide angle vehicles move more or less in opposite directions. The corresponding collision area's are of different types. In case of a sharp angle crossing there may be a large area of free critical states that can be traversed. Such a mutual critical area may be shared. In case of a wide angle crossing this large area corresponds to deadlock states because the vehicles are approaching each other from opposite directions. Now, the mutual critical area can not be shared.
In the extreme, when $\gamma = 0°$ the mutual critical area is sharable because vehicles can drive behind each other in the same direction, and when $\gamma = 180°$ the mutual critical area is unsharable because vehicles drive in opposite direction towards each other.

Suppose, the vehicles path is no more a single straight line but consists of a broken line, i.e. multiple adjacent straight line segments. The collision analysis can be obtained by considering first each of the line segments separately as unbounded lines. For each pair of line segments the configuration diagram can be derived. Second, only the parts of the configuration diagrams over the relevant path parameter ranges are combined.
As an example let path P_i be a broken line through points a_i, b_i, c_i and P_j through g_j, h_j. The mutual critical areas of line combinations $a_ib_i \leftrightarrow g_jh_j$ and $b_ic_i \leftrightarrow g_jh_j$ are determined separately, whereafter the relevant parts are combined as shown in figure 4.
The vehicle V_i travelling on path P_i will make a turn at the connection point b_i . A conservative bound with respect to collisions due to turning could be obtained by modelling a turning vehicle by its enclosing disk. Turning of vehicle V_i at point b_i induces a collision interval on path P_j as is indicated in figure 4.
Turning could lead to deadlock if one vehicle is blocked before colliding at such a position that the other vehicle can not turn anymore to proceed. In figure 4 such configurations are indicated as deadlock states.



(a) (b) (c)

Figure 4. Combination of mutual critical area's of adjacent straight path segments: (a) line segment a_ib_i with line segment g_jh_j , (b) line segment b_ic_i and line segment g_jh_j , (c) joined (broken) line $a_ib_ic_i$ and line segment g_jh_j .

Such turning phenomena could be computed in great detail, however, simple bounds on the collision and deadlock area's may be already satisfactory. Two approaches are possible in this respect: rotation zones could be estimated and added to the configuration diagrams [Bouwens94] or turning effects could be avoided completely by modelling vehicles always by their enclosing disk.

The critical area analysis so far has been based on straight path segments in order to simplify the computations. Real vehicles will follow smooth trajectories for which similar collision diagrams could be derived by analytical or numerical means. Again, one should judge whether the computational effort is worth the advantage of having sharp bounds on collision and deadlock area's.
A computational attractive approach would be to model a vehicles path as lying between two broken lines as boundaries[1]. The locus of the vehicle could be assumed to be included by these limiting cases, informally written as $\underline{L}_i(s) \le L_i(s) \le \overline{L}_i(s)$. Bounds on collision and deadlock area's for the trajectory boundaries could be combined in a conservative manner such that the bounds are valid also for any included trajectory.

3 Deadlock avoidance

Mutual critical area's of vehicles come into view if their paths intersect, coincide or are so close that a collision could happen. A distinction has been made between the actual collision area and the mutual critical area. If for example paths coincide, the mutual critical area extends along the whole path whereas the collision area contains the actual collision states.

---

[1] If for example a path is defined by spline interpolation the trajectory is known to lie within the convex hull of the supporting vertices.

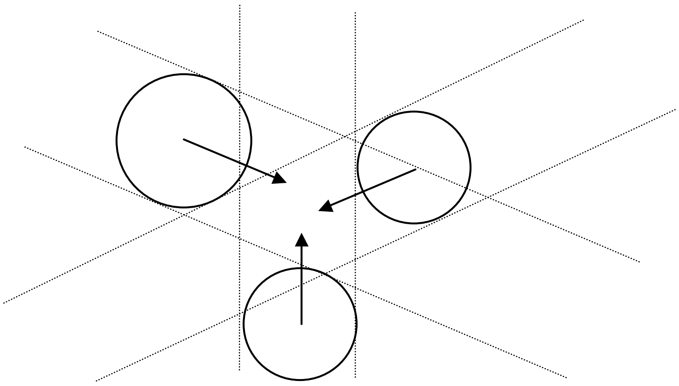


Figure 5. Deadlock because of cyclic blocking at critical sections.

As we have seen two vehicles can under some condition traverse a sharable mutual critical area without introducing collision or deadlock. The restriction is that only so-called free critical states are traversed. This corresponds for example to “weaving” of vehicles at a crossing, “merging” at a junction or “following” on a common path.

Apart from excluding mutual deadlock in critical area’s, another kind of deadlock could occur if three or more vehicles are blocked at critical sections and are caught in a cyclic waiting relation. A simple example is given in figure 5.
Cyclic waiting relations can be avoided by means of the bankers algorithm. Different variants of this algorithm exist dependent upon the fact whether single or multiple instances of resources are claimed [Silberschatz91]. If one considers mutual critical area’s as resources and allows in some cases vehicles to be simultaneously within these area’s, one would be inclined to adopt the multiple instance variant. However, a mutual critical area can not be used independently by more than one vehicle. Moreover mutual critical area’s have been defined as shared resources just between pairs of vehicles.
In fact one has to consider a mutual critical area between two vehicles as a unique resource type that may induce a wait-for relationship between these vehicles. One vehicle, say V_i, may enter the corresponding critical section unconditionally. The vehicle obtains the actual allocation of the mutual critical area. This fact can be denoted by an *allocation edge* in a so-called resource allocation graph (see figure 6). Suppose the other vehicle V_j wants to enter the corresponding critical section too. It either has to wait immediately or may enter under the condition that it follows the former vehicle. Both cases can be interpreted similarly as a follow request. This fact can be denoted as a *(follow)request edge* in the resource allocation graph as shown in figure 6.

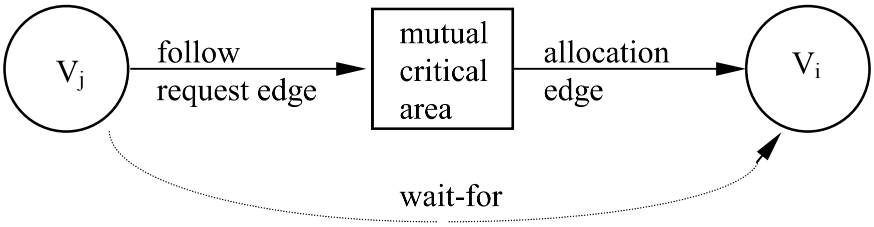


Figure 6. Example of edges in the resource allocation graph when vehicle V_j “follows” vehicle V_i.

Proceedings SION Conference Computing Science in the Netherlands, CSN94, 21-22 nov 1994, Utrecht
Ed. E. Backer, Published by Stichting Mathematisch Centrum, Amsterdam 1994, ISBN 90 6196 449 0

Whether V_j is already conditionally advancing or not, in both cases we say that there exists a wait-for relationship between V_j and V_i. One could detect deadlock by means of the presence of a cycle in the resource allocation graph (or in the associated *wait-for* graph if only wait-for relations between vehicles are represented).

The bankers algorithm requires that the maximum demands on resources are known beforehand. In case of a single instance resource type *claim edges* may be introduced in the resource allocation graph to indicate options on resources that are possibly requested in the future (see [Silberschatz91, section 6.4.2]). Allocation of a resource requires the conversion of a request edge into an allocation edge, that is a change of direction of the edge in the graph. The bankers algorithm garantees deadlock freedom by allowing edge direction changes only if they do not produce a cycle in the resource allocation graph. Also the addition of claim edges may never introduce a cycle. To be sure that this will never happen claim edges actually have to be added by a process before it does any request. In our case this means that before a cluster of connected critical sections on a vehicles path is entered all sections have to be claimed. Only when a vehicle has no allocations outstanding claims may be added. Claims may be removed at all times. If for example a vehicle can take alternative routes within a cluster of connected critical sections, claims on some sections can be dropped if it is decided not to traverse these sections.

If more than one vehicle may travel on a path, the whole path itself is a critical section. This means that if a path may be any route in some road net, i.e. vehicles may travel freely, all path sections are critical and the whole net forms one cluster. In other words claims on all sections have to be placed a priori in the resource allocation graph.

On the other hand a path long enough to contain all vehicles in the system can never contribute to a wait-for cycle between vehicles (at least if the sections of the path do not induce collisions internally). It has been stated that claim edges may be added if this will not generate cycles for sure. As a result claims need to be put forward only for a sufficient long stretch of the future path. Claims for critical sections that are far ahead of this stretch may be postponed without danger. This will reduce the number of claim edges in the resource allocation graph and hence will improve the efficiency of the bankers algorithm.

4 Application to mobile robot traffic in the MART project

Traffic control in the MART project is done partly by off-line processing and partly by on-line traffic control. Given a road-map, the off-line part computes the potential collision places. The result is stored in a revised road-map that is used by the traffic control process of the on-line part. This process is contained in the high level control software by which multiple mobile robots are guided globally to fulfil their production orders. Although actually only one mobile robot is built, the high-level control software has been designed to handle multiple robots. Beside the real mobile robot, virtual robots will be simulated. An impression of the real mobile robot that has been built is given in the picture below.

The off-line processing steps are shown in figure 8. Finding critical sections is the first step of the road-map analysis. The road map contains a description of all possible paths given as sequences of points. Paths are uni-directional. If a path should be bi-directional then it must be mentioned twice in the road map, for every direction once. To find all critical sections, all line segments of a path are compared with all line segments of other paths. Critical sections are found by computing the intersection of and the angle between the (extended) line segments.

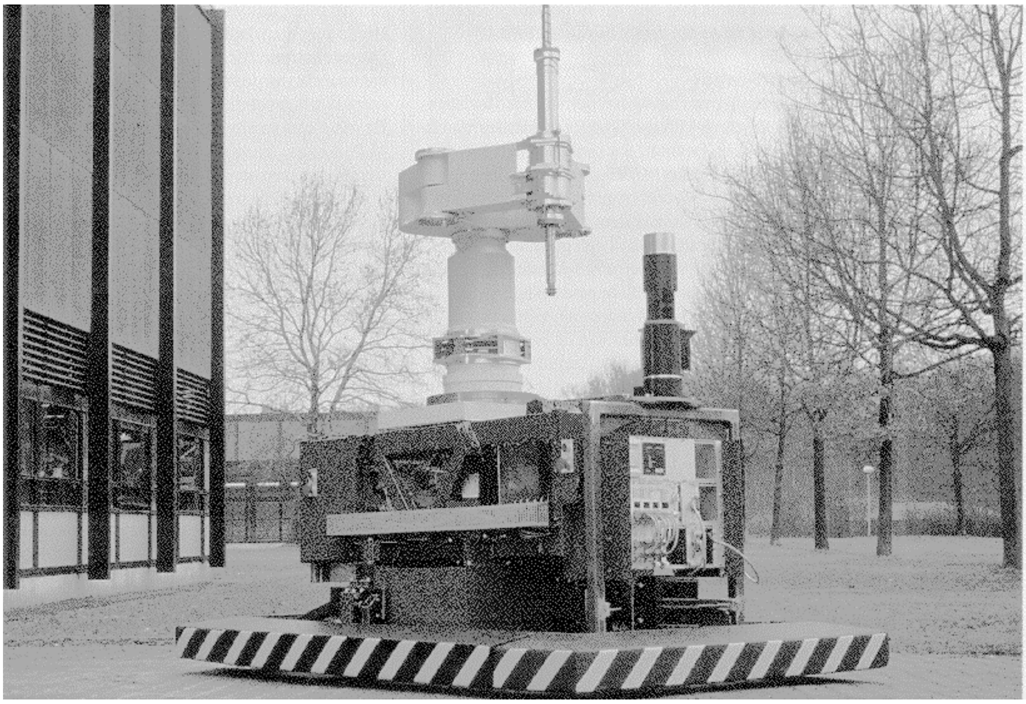


Figure 7. The MART mobile robot in its present state

The number of critical sections is reduced by joining corresponding sections on adjacent segments of the same path.

The second step of the road-map analysis is the insertion of entry and exit points of critical sections that were found in the road map by a merging operation. At the same time critical sections are clustered according to the fact whether they overlap (either directly or indirectly). Cluster information is used by the on-line traffic controller when it applies the bankers algorithm. Entering a cluster means that a claim option must be taken on all critical sections that may be travelled.

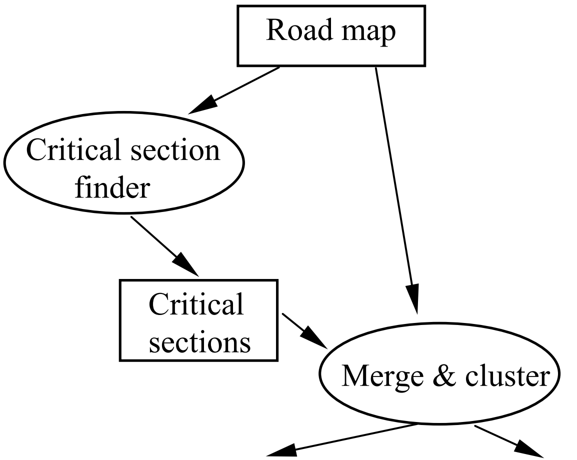


Figure 8. The off-line part of traffic control: road map analysis.

Proceedings SION Conference Computing Science in the Netherlands, CSN94, 21-22 nov 1994, Utrecht
Ed. E. Backer, Published by Stichting Mathematisch Centrum, Amsterdam 1994, ISBN 90 6196 449 0

A mobile robot that according to the revised road map arrives at an entry point of a critical section must make a request to enter. It will send a message to the traffic controller with its own identification and the concerning critical section. The traffic controller after executing the bankers algorithm can decide in the following ways:

1. The request is granted unconditionally, which means that the critical section is allocated and may be passed. It is planned to let the mobile robots put their requests just early enough such that they can proceed at full speed if the critical section is given free.
2. The request is granted conditionally which means that one or more vehicles must be "followed" while passing the section. This will be the case if the critical section is not free but of a sharable type.
3. The request is not (yet) granted and remains pending. This will be the case if the critical section is neither free nor sharable or if entering may lead to deadlock.

Following is made the responsibility of the mobile robots themselves. By exchanging messages a mobile robot can inquire the position of the robots going ahead. It can determine its distance to the other mobile robots and adjust its velocity. It must always be able to stop in time if a predecessor stops for any reason. Following could also be implemented differently, for example by using range sensors. The MART is equipped with a laser scanner able to detect obstacles within a distance of 2.5 meters and a radial range of 180 degrees. A 'following' algorithm has not yet been worked out, but will be based on both the exchange of software messages and the physical range sensing capability.

The off-line road map analysis and the on-line traffic control have been implemented and tested by simulation [Bouwens94]. Presently it is embedded in the MART control system. The control software runs on a network of transputers, part of which is situated "on board" of the mobile robot and part of which is situated in the host system "on shore". Communication between both subnetworks is realised by a wireless radio link.

5 Conclusion

Analysis of mutual collision situations between two vehicles that follow prescribed paths leads to the notion of a mutual critical area within the joint configuration space of both vehicles. This area includes the collision states, but may also contain non-collision states. Dependent upon the geometry of the paths some of the non-collision states may be deadlock states that have to be avoided, and some may be 'free critical states', the traversal of which could be really advantageous. In this case vehicles will share the mutual critical area and unnecessary waiting will be avoided. This corresponds for example to "weaving" of vehicles at a crossing, "merging" at a junction or "following" on a common path.

Apart from mutual deadlock between two vehicles, deadlock could arise between more vehicles due to cyclic blocking at critical sections. Cyclic waiting can be avoided dynamically by the bankers algorithm. Mutual critical area's have to be considered as the unique resources that are either (1) allocated if free and allowed by the bankers algorithm, in which case a vehicle can pass it freely, or (2) granted conditionally if not free but sharable, in which case a vehicle may already advance while following the vehicle that possesses the allocation, or (3) not granted in which case the request remains pending and the vehicle has to wait.

The bankers algorithm requires that future claims are known before any allocation request is done. This implies that claim options have to be taken on all critical sections of a cluster when a cluster of connected sections is entered. The complexity of the bankers algorithm is reduced if critical sections can be divided in geometrically independent clusters. Because the number of vehicles places a limit on the maximum length of a cyclic waiting chain, future claims only have to be registered over a limited stretch ahead. This argument may also reduce the number of future claims and reduce the complexity of the bankers algorithm.

The traffic control strategy has been applied to the mobile robots in the MART project. Road map analysis is done by off-line pre-processing. Critical sections are identified by comparing line segments of a path with all line segments of other paths. Entry and exit points of critical sections are added to the road map. A central on-line traffic controller process does handle entry requests and exit messages of the mobile robots. Access to critical sections is granted according to the bankers algorithm as described. The off-line road map analysis and the on-line traffic control algorithm have been implemented. The traffic control system is yet to be integrated in the MART control software, but has already been tested by simulation.

References

[Bouwens94] P.J. Bouwens, Traffic control of mobile robots in the MART project, Master Thesis, Department of Computer Science, University of Twente, Report SPA 94-05, March 1994.

[Kant86] K. Kant and S.W. Zucker, Towards efficient trajectory planning; the path-velocity decomposition, The International Journal of Robotics Research, Vol.5 No.3, 1986.

[Siberschatz93] A. Silberschatz , J. Peterson and P. Galvin, Operating system concepts, 3rd Edition, Addison-Wesley 1991.

[Tillema93] MART: An overview of the Mobile Autonomous Robot Twente project, University of Twente, Memoranda Informatica 93-12, edited by H.G. Tillema, 1993, ISSN 0924-3755.