Proceedings of the 1999 Summer Computer Simulation Conference (SCSC 1999).
July 1999. Chicago [SCS]. ISBN 1-56555-173-7

TRACES: TRAFFIC CONTROL ENGINEERING SYSTEM
A case-study on container terminal automation

Mark B. Duinkerken, Joseph J.M. Evers and Jaap A. Ottjes
Faculty of Mechanical Engineering and Marine Technology
Delft University of Technology
Mekelweg 2, 2628 CD Delft, The Netherlands
e-mail : m.b.duinkerken@wbmt.tudelft.nl

Keywords

Automatic control, Control systems, Discrete simulation, Transportation.

Abstract

In this study a control system to coordinate the traffic flow of Automated Guided Vehicles (AGVs) is presented. This control system, TRACES (Traffic Control Engineering System), coordinates a system of AGVs, which are autonomous transport units.
A prototype was built as a pilot-study for the TRACES system. The formulation of the problem and simulation environment in which TRACES was tested is discussed.
For this study of the system an existing automated container terminal has been chosen. Here about 50 AGVs operating in a relatively small area look after the transport of containers between the quay where ships are loaded and unloaded and the stack where containers are stored for longer or shorter periods. Conclusions from this research and future research and applications are presented.

Introduction

Automated Guided Vehicles (AGVs) are increasingly used for transportation of both people and goods. When the system only uses a few AGVs the coordination of the vehicles is relatively simple. As the number of AGVs, and thereby the complexity of the system, increases the need for a traffic control system also grows. Requirements which have to be satisfied by such a system are:

- monitoring of the physical safety: vehicles must not crash with each other or their environs
- interfacing with job-allocation
- flexibility: possibility of alternative routing when there are static or dynamic blockages
- verifiability: checking of possible deadlocks and theoretical capacity
- scalable and expandable through modular approach

In this paper a traffic control system which satisfies the above-mentioned requirements is introduced. The system is based on the semaphore concept that is well known from automation theory. The Traffic Control Engineering System, TRACES, can be used in three phases of a design process:

- design of a layout and the associated traffic control
- simulation of the designed model to verify the layout and to measure the performance
- implementation of the control of the actual transport system.

To prove the correctness of TRACES, a case-study is completed. The automated container terminal DSL at the Maasvlakte, Rotterdam is modelled and simulated using TRACES. It can be concluded that the performance of the simulated model matches the performance of the actual terminal. The model offers possibilities to investigate and eliminate bottlenecks on the DSL terminal. Furthermore, the insights gained with this model leads to building new, more complex terminal layouts with the help of TRACES.

TRACES

General description

The task of the system is to control the traffic, to instruct individual vehicles with respect to their destination (or mission) and routing, to direct conflicting common use of traffic infrastructure and to provide facilities for communication between a control centre and the vehicles. TRACES has been developed to provide such a traffic control system. Important requirements were robustness, the power to support high performance control, in order to cope with high traffic intensities on any scale and to take into account the vehicle characteristics.

In the set-up of TRACES it is assumed that vehicles only communicate with the local control centre (LCC) and that they drive along routes (coded in scripts), that are issued by the LCC. An LCC is part of a ‘local scene’, which is equipped with an apparatus for autonomous information processing. Vehicles are assumed to be able to maintain the appropriate distance from vehicles driving in front of them independently of the LCC. This implies that, in principle, vehicles may drive in line in the same ‘control area’. They must also be equipped with the facilities needed for independent data processing.

It should be noted that the control of distance from obstacles or preceding vehicles by the AGVs themselves is not a crucial condition for the use of TRACES. Instead of distance control by the AGVs, it is also possible to allow the distance control to proceed by means of TRACES by giving each control area of the route a capacity of 1. For longer distances this leads to much communication with the controller and to inefficient use of the infrastructure. In the example given in this paper the latter approach has been used.

Conventional traffic control is directly linked with the traffic infrastructure by traffic lights and situational traffic rules. In TRACES traffic control takes place entirely within a computer model, where, within some margin, synchronisation is preserved between the progress of the physical processes (i.e. the moves of the vehicles) and the evolution of the state of the computer model. The computer model is formulated in terms of abstract event processes. To stress the point that the physical reality and its representation as a computer model are different worlds, and to avoid confusion, the terminology of the traffic model is kept abstract.

Semaphores

The concept the semaphore is borrowed from the automaton theory (Dijkstra 1968; Ben-Ari 1990). The semaphore is a non-negative integer variable, say ‘S’, with the interpretation of freely available capacity. Two operations are defined by this: ‘wait’ and ‘signal’. The operation ‘wait’ is activated whenever an actor (vehicle) reports itself for the use of a controlled resource, while ‘signal’ is executed whenever the actor releases this resource. The operations are defined as:

- wait (S) if $S > 0$: set $S := S-1$ and allocate 1 capacity unit of the resource ,
  if $S = 0$, block the entry to the resource;
- signal (S) if there are waiting actors admit one of these to the controlled resource,
  if there are no waiting actors, set $S := S + 1$.

An initial value, which can be interpreted as the capacity of the controlled resource, is given to semaphore S. In Figure 1, a semaphore is shown with a capacity of 2. In the pictures different wait- and signal occurences are shown.

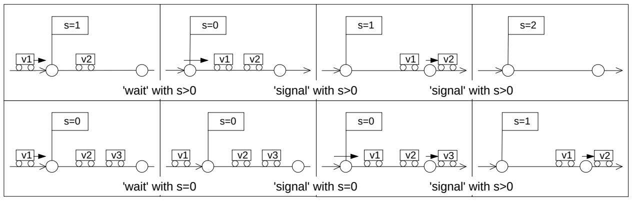

Figure 1 Example semaphore with capacity 2

It is important to observe that when more than 1 applicant is waiting on a busy semaphore (i.e. S = 0 and queue >1), the operation ‘signal’ will allocate 1 capacity unit to one of the applicants in the queue. However, this applicant must be additionally specified by the so-called queue discipline. Usually, the queue discipline is FIFO: ‘first-in-first-out’.

Several variants of the semaphore applications are proposed (Ben-Ari 1990) and also an application in the context of traffic control (Evers 1996a). In order to obtain the appropriate control of the synchronisation between the physical traffic processes and TRACES, several extensions are implemented. For the present study two extensions are relevant. First, the split-up of Dijkstra’s wait-operation into ‘reserve’ (i.e. request for semaphore capacity), followed by ‘watch’ (i.e. observe whether the application is granted, and if not, follow by waiting until it is granted). Secondly, the introduction of the data-type ‘ticket’ as a means to place a request for semaphore-capacity. A ‘ticket’ is used in four stages:

1. the creation, where also the semaphore that is involved is specified;
2. the request for capacity on this semaphore;
3. observing the result of the application;
4. the return of the capacity, comparable with Dijkstra’s signal-operation.

Scripts

Initially, the concept of semaphores was developed for use in information technology. There, co-ordination does not take into account physical dimensions and time aspects. However, in applying semaphores to physical processes like vehicle guidance these aspects are essential. This is one of the reasons why the concept of the semaphore has been extended by adding abilities to anticipate physical moves and to arrange termination. This is done by the use of a script.

The script describes a route between two points in the infrastructure. This route can be considered from both physical and logistics points of view.

Physical

Physically, a route is the path that a vehicle must travel in order to carry out a transport order. Nodes and corridors are used to define a path. A node is an indication of a position in the infrastructure, a corridor is a predefined link between two nodes. In the simplest case this link is a straight line, but it may also be used to describe more complex vehicle movements such as 90° bends, S-bends and lateral movement. Figure 2 shows an example of a simple layout.

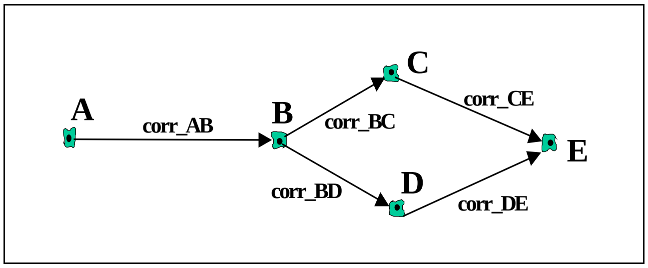

Figure 2 Example of a layout with 5 nodes and 5 corridors

Logistic

A script consists of a combination of physical and logistic concepts. A script describes the physical route between beginning and end points, and indicates when tickets for specific semaphores must be claimed and released. A number of alternative physical routes may be defined in one script, in which dynamically, during the execution of the script, a specific route is chosen on the basis of the available tickets. By indexing the objects for both the physical infrastructure (nodes and corridors) and the logistic control (semaphores), it is possible to describe complex, repeating patterns in a simple way.

As an example, a script is defined in the layout shown in Figure 2. Let S1, S2 and S3 be semaphores with capacity 1. The script, given in Figure 3, describes the route between the nodes A and E. Depending upon the availability of a semaphore (S2), the AGV executing this script chooses between two subscripts, defining different physical routes.

```
Script syntax       Description

check A             position is node A
ask S1              wait semaphore S1
do corr_AB          drive corridor from A to B
free S1             signal semaphore S1
check B             position is node B
select              select subscript :
  if S2
    ask S2            if S2 available, then go
    do corr_BC        from B to E via node C,
    do corr_CE
    free S2
  endif
  else                else go via node D
    ask S3
    do corr_BD
    do corr_DE
    free S3
  endelse
endselect
check E             position is node E
```

Figure 3 Example of a script

Capacity allocation by using tickets

In the example of Figure 3, the waits and signals on semaphores are mentioned in the script. As mentioned before, in TRACES, these actions are realized with the help of so-called tickets. In this paragraph 4 examples for the use of tickets are given.

Physical capacity of infrastructure

The physical monitoring of the infrastructure is achieved by giving each corridor a capacity of 1 (Note: distance monitoring by vehicles does make this rule obsolete). A vehicle that intends to enter a corridor must first have obtained a ticket. This ticket is returned when it leaves the corridor. If two corridors coincide for part of their length a common semaphore must be defined.

Prevention of deadlock

The prevention of deadlock (and gridlock) is important. The standard example of deadlock in traffic situations is a crossing that can be approached from four directions. To prevent four vehicles from each claiming part of the crossing and thus blocking it, a ticket is defined on a higher logistic level. This ticket is necessary for every vehicle that will use the crossing and has a maximum capacity of 3. It is easy to see that with this extra ticket deadlock can be prevented. Extra tickets can be defined at every level. Tickets to prevent deadlock can be defined for crossings, the entries and exits to stacks and even for entire terminals.

A less trivial example of a deadlock is given in Figure 4. Both vehicles needs space to make a 90° turn. The space needed for this manoeuvre is shown, none of them can proceed.

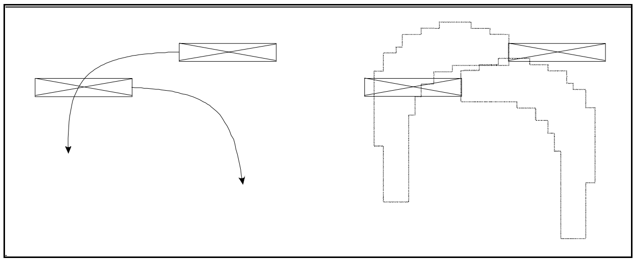

Figure 4 Deadlock example with only two vehicles

Selection of alternative routes

A special type of ticket that is defined within TRACES is the selection ticket. This ticket is composed from several single tickets. On the basis of the current availability of tickets, a choice is made between the alternatives in which each alternative has a different route described by a different sub-script.
Because it is possible to determine which physical route is to be followed only during the execution of the script, the routing has become dynamic. This ensures that the movements at the terminal are streamlined. Moreover, temporary blocking of the infrastructure, for example by a vehicle that has broken down, can be accommodated.

Higher level control

In addition to the vehicles that approach the LCC for tickets, a control system can function on a higher level which claims and releases the same tickets. This permits external intervention in the current traffic situation. A practical application of this control, for example, is the sequence monitoring when AGVs must reach the quay crane in a fixed order.

CASE-STUDY : DSL MAASVLAKTE

Introduction

The objective of this case study is to investigate the usability of TRACES in an existing, complex system. At the Delta Sealand terminal (DSL) of ECT, on the Maasvlakte, about 50 AGVs carry out the transport of containers between the stack and the quay cranes. High demands are made on this system with regard to capacity, throughput speed and reliability. The main purpose of the reproduction of this situation on the basis of TRACES-concepts is to validate the principles on which TRACES is based.

Layout

The stack-quay traffic follows a circular pattern. Typically, the vehicles travel along the entire length of the ship and turn back along the stack. In the quay area a fixed traffic lane is reserved for each quay crane. In the stack-area the traffic for two quay-cranes is combined to form a single traffic lane.

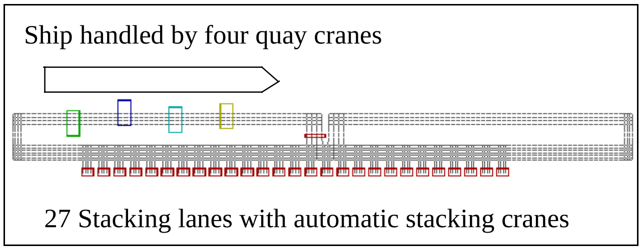

Figure 5 Layout of the DSL terminal at the Maasvlakte Rotterdam

Figure 5 shows the traffic lanes for the DSL-layout. To simplify this problem, only 4 of the 8 quay-cranes are modelled.

Simulation

To investigate the layout, an environment was built to simulate the relevant terminal operations. This simulation model consists of objects representing quay cranes, stacking cranes, AGVs and a loading plan. For the logistical control of the loading process, an MRP-like planning procedure was built, which is part of the concept SERVICES, described in (Evers 1999). In addition, an empty-car management procedure is provided. The effect of various factors on the capacity of the terminal is examined including :

- Number of AGVs
- Vehicle characteristics such as speed and acceleration
- Quay cranes cycle time distribution
- Stacking cranes response time distribution
- Sequence monitoring
- Manoeuvres to change orientation of AGV
- Prepositioning of containers in preferred stacking lanes
- Incidental breakdown of equipment and vehicles
- Interference by AGV-traffic with different origins and destinations

Results

The case-study presented above leads to a model with a total of more than 5000 entities (these mainly being nodes, corridors and semaphores). The experiments with this model have led to the following results. Despite the large number of entities, the model remains fast and stable. The measured performance of the simulated model appear to be of the same order of magnitude as the measurements on the actual system. Further study is necessary to verify this more accurately. (This verification will also be done by using ECTs existing model called DCT.)

In the actual daily terminal operations factors play a part that can not be modelled easily. A number of these factors are caused by human actions and their influence will be reduced by an increased degree of automation. It is expected that the performance of a further automated terminal will match the simulation results closer.

CONCLUSIONS

Current implementations of TRACES

TRACES has been shown to be suitable to study complicated traffic situations such as those encountered at terminals. The model can also be used to design new complex systems. The modular structure of TRACES ensures that the models remain clearly arranged. Moreover the designs can be quickly adapted to meet new demands relating to infrastructure and equipment. The simulation capabilities of TRACES makes it possible to determine the performance of new, complex designs.

Future developments and appications

TRACES is not only applicable within simulation models. It is also intended that it should be used in practice for to control operations. The control models can thus be defined, tested by using simulations and then used for the operational process.
At present the TRACES concept is or will be applied in the following projects:

- FAMAS : First All Modes All Sizes. The goal of this research programme is to develop a new generation of container terminals capable of handling all modalities (ship, train, truck etc.) of all sizes according to customer-specified service levels. The heart of the FAMAS programme is the automation of terminal transport and the central stack.
- OLS : Underground Logistic System: the planned underground link between the transhipment points in the flower auction market in Aalsmeer, the cargo area of Schiphol Airport and a rail terminal at Hoofddorp. TRACES is used in both the simulation and the AGV-control parts of the project.
- AGV-control: An AGV-producer is making AGVs that can navigate freely within a predefined area. Research is underway to investigate whether the new generation of AGV-control can be based on TRACES.
- Logistic laboratory: In this laboratory at the Delft University of Technology several AGVs will move about in a test environment. The TRACES concepts will be applied and tested here.

REFERENCES

Ben-Ari, M. 1990. *Principles of Concurrent and Distributed Programming.* Prentice Hall.
Dijkstra, E.W. 1968. *Co-operating sequential processes.* In F. Genuys (ed.) Programming Languages, Academic Press, New York.
Evers, J.J.M.. 1994. *DiTrans: an R&D programm for the development of an integrated system for robotized large scale container transport.* TRAIL Studies, Delft University of Technology.
Evers, J.J.M.; S. Koppers. 1996. *Automated guided vehicle traffic control at a container terminal.* Transportation Research (A), (Jan.).
Evers, J.J.M. 1996. *The Terminal of the Future.* Proceedings Terminal Operator Congress (TOC), Hamburg.
Evers, J.J.M.; G.J.M. van der Wielen, M.B. Duinkerken. 1997. *Logistic Terminal Lay-outs*. TRAIL Studies 97-2, Delft University of Technology.
Evers, J.J.M.; L. Loeve. 1997. *TRACES: an object based System for TRAnsport Control Engineering in Java.* TRAIL Studies, Delft University of Technology.
Evers, J.J.M.; L. Loeve; D.G. Lindeijer. 1999. *The service-oriented agile logistic control engineering system : SERVICES.* TRAIL Research School, Delft University of Technology (Apr.).
Hoare, C.A.R. 1985. *Communicating Sequential Processes.* Prentice Hall Int.
Ottjes, J.A.; M.B. Duinkerken; J.J.M. Evers; R. Dekker. 1996. *Robotised Inter Terminal Transport of Containers.* Proceedings 8th European Simulation Symposium. Genua [SCS] pp 621-625.
Sommerhalder, R. 1996. *A formalisation of structured traffic control networks using colored semaphores.* TRAIL Studies, Delft University of Technology.