Decentralized and Dynamic Routing for a Cognitive Conveyor

T. Krühn* and L. Overmeyer

Institute of Transport and Automation Technology, Leibniz Universität Hannover, Germany
*E-Mail: tobias.kruehn@ita.uni-hannover.de

Summary: Today's logistic industry faces the challenge of short product life cycles and small batch sizes. Technical solutions that provide enormous flexibility are therefore needed to meet the customer's needs. The Cognitive Conveyor provides this flexibility for tomorrow's logistics. Small-scaled, multidirectional conveyor modules are coupled to a network. This network solves transportation tasks without a central control by cooperation of the modules. To move a transport unit, multiple modules have to form a group which acts synchronously and target-oriented. To prevent deadlocks, routes have to be reserved before a transport unit is moved on the conveyor. This contribution addresses the group forming and presents the routing algorithm for transport units on the Cognitive Conveyor.

Keywords: Logistics, Control, Algorithm, Decision Making.

1. Introduction

As part of the logistics industry, in-plant material handling offers an enormous potential for rationalisation for all industries [1, 2]. The requirements of a material handling system are becoming more complex from the mechanical point of view concerning the material to be conveyed, its functionality as well as the system control. Today's systems are mechanically predefined and therefore functionalities such as sorting or transferring are locally bound to the system.

Conveyors are usually designed according to the largest material that can be handled, which usually results in reduced performance for smaller materials to be conveyed. Layout changes require considerable effort. A new trend-setting solution for material handling must meet individual complex tasks of transportation with implemented intelligence and must be flexible enough for changes and adjustments with low investment costs. Operating intralogistic plants, maintenance, repair time, corresponding availability and functional flexibility play a substantial role. [3]

Small-scaled, flexible material flow components with integrated control and data processing provide a solution. Connected to a module matrix (Figure 1) these single modules are able to solve complex material flow tasks and promise cost advantages by their uniform design. These small-scaled modules allow a simplified exchange of entire modules without complex assembly and adjustment work.

The first prototypes of small-scaled, multidirectional conveyor modules have been designed, constructed and tested successfully. Each module is equipped with a light sensor to detect the presence of a transport unit. The identification of transport units takes place as soon as the unit enters the conveyor system. A module covers an area of 70 mm x 70 mm and has a height of 200 mm.

Besides the mechanical design of the small-scaled conveyor modules, the control of these elements needs to be considered. Common materials handling systems are controlled by a central control system. The control system has to control each individual element. If the materials handling system is changed, for example a conveyor is replaced by a sorter, the control system also needs to be reconfigured.

Due to the large number of modules in a system consisting of small-scaled modules, it is not possible to program a control for each element. Furthermore, it should be possible to modify the materials handling system without the need to reprogram a central control [4]. The solution – a decentralized control for small-scaled conveyor modules – will be the subject of this paper.

The key difference to other decentralized controls for materials handling as described in [5] is that the transport units are much larger than the conveyor modules. Thus, modules have to form groups to move the transport units placed on them. Moreover, the fine granularity of the modules infers problems with common routing protocols and deadlock protection mechanisms as shown in this paper.

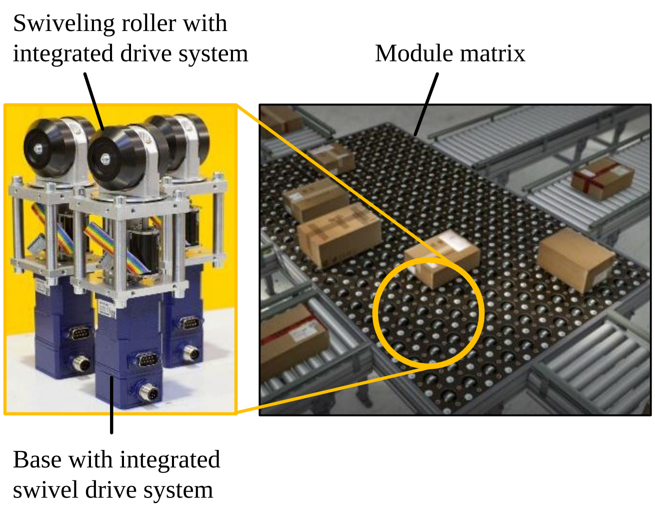

Figure 1. Small-scaled conveyor modules connected to a module matrix.

Besides the routing task, the forming of groups needs to be considered. Multiple small-scaled modules need to team up to handle a transport unit. While transport units move on the matrix, the team membership of modules changes dynamically and rapidly. Hence, the team-building algorithm must be efficient and fast enough, allowing it to be implemented in a real-time environment.

2. Group forming

To move a transport unit and to solve tasks such as the deadlock prevention on crossings, modules have to form dynamically changing groups. As in the field of cellular automata, these groups are called neighbourhoods. Figure 2 shows three neighbourhoods. [6, 7]

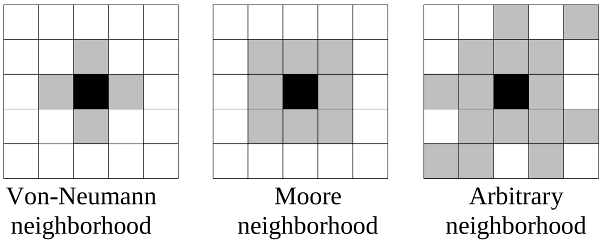

Figure 2. Examples for neighbourhoods in cellular automata [6].

To present an example, this contribution explains the rules to form the neighbourhood for a transport unit: The neighbourhood N_T is defined by the size of the transport unit located on the modules. Since the cells have no global information, they need a rule to determine their neighbourhood membership by their sensor information and the neighbours' states. The transport unit ID is used as a neighbourhood identifier. The variable's value at the next time step will be:

1. The current ID, if the cell has got an ID and its sensor detects a transport unit, or else
2. The ID of any cell C in the Von-Neumann-neighbourhood (Figure 2), if C's sensor detects a transport unit and C has got an ID, or else
3. The ID of any cell C in the Moore neighbourhood, if C's sensor detects a transport unit and C has got an ID or else
4. No ID

Using these rules, the transport unit's ID is transferred from cell to cell without a central control. The information follows the moving transport unit through the transporting modules, forming an information shadow. Together with the module's ID, any information regarding the transport unit can be transferred. This information can contain simple structured data or entire software agents. With similar simple rules neighbourhoods for other applications can be implemented, e.g. for crossing recognition.

3. Routing

Known routing algorithms with a huge variety of metrics for transport and communication systems are designed to transport a packet from one node to another; the packets only occupy one node at a time [8]. Algorithms for small-scaled modules have to work in a completely different way. Transport units have a fixed shape and always cover multiple modules at a time, which has wide-ranging consequences. For instance, it is possible that a route from one location to another exists but it is too narrow for the transport unit to pass or two transport units run into a deadlock on an intersection (Figure 3). Moreover, in narrow parts of the conveyor, the rotation of the transport unit is reasonable.

Furthermore, multiple modules have to act in a coordinated and synchronous way to move the transport unit located on them. The transport unit's movement is a result of multiple forces affecting it. The movement depends on physical constraints such as its mass centre, size, friction and system boundaries. The enumerated factors are outside the scope of common routing algorithms.

To tackle the problem, three approaches are proposed: either reduce the problem to a graph that can be used for common routing algorithms, modify known algorithms or find a new one to handle the new problem structure. All approaches are promising and will be examined. Furthermore, the acting of neighbouring modules has to be synchronised to handle transport units. This is a task beyond the capabilities of well-known routing algorithms.

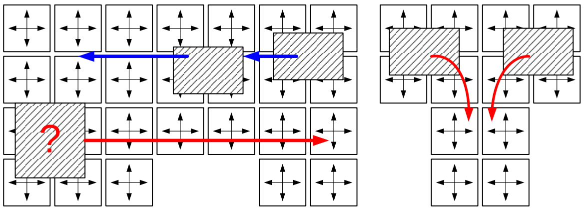

Figure 3. Routing with conventional algorithms fails due to physical width of transport units.

For the Cognitive Conveyor, a distance vector routing metric with split-horizon and temporarily locked dependencies is used, which is proven to be loop-free and quickly converging [9]. The cost function considers other routes in the system, the transport unit dimensions, necessary rotations and the physical system boundaries. The metric value M of a module is given by:

$$M = M_s + M_R + M_N + \begin{cases} M_0, if\ route\ exists \\ M_G, if\ opposing\ route\ exists \\ \infty, if\ system\ boundary\ too\ close \end{cases}$$

- M_s depends on the modules distance to the system boundary.
- M_R are the costs for rotating the transport unit.
- M_N is the metric value of the best neighbour.
- M_0 is the transport cost to move a transport unit by one module in the direction of an existing route.
- M_G is the transport cost to move a transport unit by one module against the direction of an existing route.

The quotient of M_G/M_0 defines the maximum detour a planned route will accept before the system tries to plan a route conflicting with an existing route. This may result in a situation, where the existing route will be re-planned. If this is not desired, the value of M_G/M_0 must be larger than the longest route in the system. The determination of the specific values is an optimisation problem and depends on the desired system behaviour.

Every time a module changes its metric value, the neighbours are informed and hence update their metric. Thus, each time a transport unit is inserted into the system and a new route is planned, the metric values of the modules may change.

The route reservation starts after a transport unit enters the Cognitive Conveyor. The system inserting the transport unit has also to provide the destination, because the small scaled modules are not able to identify the transport unit or to communicate with it.

An arbitrary module beneath the transport unit starts the reservation and informs the neighbour with the best metric, so that it must continue the route planning. That neighbour then subsequently informs its next neighbour with the best metric and so on. Hence, a route reservation runs from the transport unit to the destination. After the route reaches the destination, the destination module confirms the route. This confirmation propagates the route backwards and changes the state of each module from waiting to reserved. After the whole route is reserved, the transport unit starts moving through the system.

An existing route may be re-planned with one constraint: After a route was confirmed and the transport unit starts moving, it must be ensured that always a reserved and confirmed route exists. Thus, if an alternative route is planned, the existing route can only be dissolved if the alternative route is already confirmed.

4. Simulation results

The Cognitive Conveyor is simulated on basis of the multi-agent simulation toolkit MASON. Figure 4 shows an example situation where an existing route from transport unit TU_1 to its destination is re-planned after TU_3 is inserted into the system. Each square in figure 4 represents a small-scaled conveyor module, which can interact only with the four modules in the Von-Neumann neighbourhood.

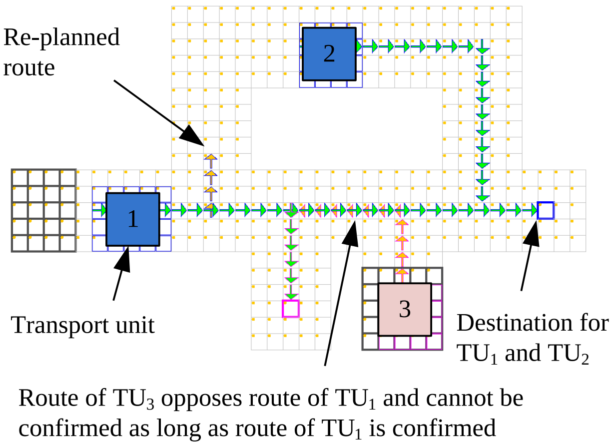

Figure 4. Example of re-planned route.

Parameters influencing the route planning and the used metric were systematically examined, including the ratios between M_s, M_R, M_0 and M_G. Moreover, a large number of topologies have been tested. The parameters were optimized by a simple hill climbing towards a higher average transport unit speed.

The executed simulations showed that the massively decentralized conveyor control on the basis of the small-scaled conveyor modules is possible and working properly. The metric for the routing is quickly converging, even during permanently changing condition on the conveyor. Routes are planned fast and reliably. The routing algorithm works efficient in complex topologies. Once a route is found, it is rarely re-planned. Due to the dynamic routing algorithm, the system is able to adapt changing load conditions.

In complex topologies, congestion and deadlocks can occur. To simulate the system, also a deadlock prevention algorithm was implemented, which is not addressed in this paper. The occurrence of congestion will be addressed in the ongoing research project.

5. Conclusion

The Cognitive Conveyor, consisting of small-scaled, multidirectional transport modules, is able to solve arbitrary transport tasks. The route planning and reservation presented in this contribution demonstrates the feasibility of a completely decentralized control of the conveyor. Transportation tasks are efficiently solved by the network of modules without a central control. Moreover, the system adjusts to changing load situation autonomously.

Current work concentrates on complex tasks such as the buffering and reordering of transport units. Additionally, the integration of the Cognitive Conveyor into larger conventional conveyors as well as in decentralised systems is considered.

Acknowledgements

This contribution was supported by the German federal state of Lower Saxony with funds of the European Regional Development Fund (ERDF) within the scope of the research project "Cognitive Logistic Networks (CogniLog)".

References

[1] Arnold, D., Furmans, K., 2007, Materialfluss in Logistiksystemen, Springer Verlag Heidelberg, 5. edn.

[2]Wolf-Kluthausen, H. (Ed.), 2007, Jahrbuch Logistik 2007, free beratung Gesellschaft für Kommunikation im Marketing GmbH, Unit Logistik.

[3] Bullinger, H.-J., ten Hompel, M., 2007, Internet der Dinge, Springer, Berlin.

[4] Overmeyer, L., Heiserich, G., Falkenberg, S., Jungk, A., 2009, Automatische Konfiguration und Optimierung von Materialflusssystemen durch kognitive Logistikmodule. 18. Deutscher Materialfluss-Kongress; VDI-Berichte 2066, Düsseldorf: VDI Verlag:197-208.

[5] Mayer, S. H., 2009, Development of a completely decentralized control system for modular continuous conveyors, Vol. 73, Universitätsverlag Karlsruhe.

[6] Dogaru, R., 2008, Systematic Design for Emergence in Cellular Nonlinear Networks,With Applications in Natural Computing and Signal Processing, Springer-Verlag Berlin Heidelberg (Studies in Computational Intelligence, 95).

[7] Wolfram, S., 1986, Theory and Application of Cellular Automata, (including selected papers 1983-1986). Singapore: World Scientific Publ.

[8] Baumann, R., Heimlicher, S., Strasser, M., Weibel, A., 2007, A Survey on Routing Metrics, Technical report, Computer Engineering and Networks Laboratory, ETH-Zentrum, Switzerland.

[9] Jaffe, J., Moss, F., 1982, A Responsive Distributed Routing Algorithm for Computer Networks: Communications, IEEE Transactions on, 7:1758-1762.