

Hierarchical Routing Algorithm for Industrial Mobile Robots by Signal Temporal Logic Specifications

[Extended Abstract]

Balázs Csutak
Faculty of Information
Technology and Bionics,
Pázmány Péter Catholic
University
H-1083 Práter u. 50/a,
Budapest, Hungary
csutak.balazs@hallgato.
ppke.hu

Tamás Péni
Systems and Control
Laboratory, Institute for
Computer Science and
Control
H-1111 Kende u. 13-17.,
Budapest, Hungary
peni.tamas@sztaki.mta.hu

Gábor Szederkényi
Faculty of Information
Technology and Bionics,
Pázmány Péter Catholic
University
H-1083 Práter u. 50/a,
Budapest, Hungary
szederkenyi@itk.ppke.hu

ABSTRACT

A two-level route planning algorithm based on model predictive control (MPC) is proposed in this paper for industrial mobile robots, executing tasks in an environment specified using the methodology of signal temporal logic (STL). STL is applied to describe various conditions like collision-free and deadlock-free operation, followed by the transformation of the formulas into a mixed integer linear programming (MILP) problem, solved using dedicated software. To achieve real-time operation, the route planning is divided into two distinct phases using different underlying vehicle models. The correctness of the approach is guaranteed by the applied formal design method.

Categories and Subject Descriptors

[**Embedded and cyber-physical systems**]: Robotics—*Robotic control*; [**Applied computing**]: Physical sciences and engineering—*Engineering*

Keywords

mobile robots, route planning, signal temporal logic, optimization

1. INTRODUCTION

Optimal route planning based on transport demands is an intensively investigated topic in engineering fields. Depending on the applied model and assumptions, the computational complexity of such tasks and the effectiveness of the solution moves on a wide scale.

The problem itself generally consists of numerous autonomous guided vehicles (AGV) moving along given routes in a closed

space (e.g. in an industrial plant), assuming a microscopic routing environment (i.e., the size of the vehicles is not negligible compared to the available space). This environment can be modeled as a directed graph, with only one agent allowed at a time in a given node or edge [1], which is suitable for a physically large setting, but might prove to be ineffective in a more crowded location. As another possible approach, the plant can be modeled as a coordinates system, in which agents can move freely with the exception of a number of obstacles or restricted zones.

Some of the solutions concentrate on giving suboptimal but real-time solution for the problem, using greedy iterative algorithms or heuristics. In the simplest case, the route planning of the agents is carried out in a completely independent manner: knowing the location of the obstacles, each agent computes a path locally, and avoids collision with other vehicles in real-time. This approach however is neither optimal regarding the completion time of the movements, nor has any guarantee to prevent situations like a deadlock formation. More complex solutions feature distributed calculation, but with a centrally accessible resource (like already planned paths of the other vehicles) [1, 2].

It is also possible to model route planning tasks as optimization problems [3]. These algorithms are indeed capable of giving the truly optimal solution regarding completion time, guarantee the collision-free and deadlock-free operation on the price of high computational complexity which might hamper real-time application.

Linear Temporal Logic (LTL) is a formalism originally developed for the formal design and verification of computer programs. In essence, LTL extends the set of operators familiar from Boolean logic with tools to describe time dependencies between the statements (such as 'always', 'never', 'eventually', 'next step', etc.). Signal Temporal Logic (STL) further extends the possibilities offered by LTL by introducing quantitative operators regarding time in LTL formulas. STL has been successfully used for describing traffic-related systems, due to its ability to express complex rule sets or complicated time dependencies [4, 7, 8].

One of the first results on the successful application of LTL in vehicle routing is [9], where the computation task is written as a network flow problem with constraints. In [10], an incremental algorithm is given for the complex path planning of a single robot, where the specifications are given using LTL. A motion planning method for multiple agents is presented in [11], where STL is used for describing specifications, taking into consideration imperfect communication channels, too.

The above results motivated us to use STL in our work for route planning. The main novelty of our approach is the division of the computation into two levels which allows us to handle a relatively long prediction horizon with a significantly reduced computation time.

2. PROBLEM FORMULATION

The problem is to plan and track routes for multiple AGVs, each of which is assumed to move in a two dimensional closed space, independently in the x and y directions. All agents have upper bounds for speed and velocity, respectively. Each agent has a target point assigned before running the planning algorithm. The objective for each agent is reaching its target point in minimal possible time, without colliding with obstacles or with each other.

Let N be the number of agents. On the low level, we model the agents as simple mass-points in a two dimensional cartesian coordinates system. Therefore, the motion of each agent can be described in continuous time as two double integrators with acceleration command inputs, resulting in $4N$ state variables (coordinates and velocities), and $2N$ inputs. This model is discretized in time using a uniform sampling time t_s . Let $x_i(k)$ and $y_i(k)$ denote the x and y coordinates of the i th robot at time step k , respectively. The inputs for agent i at time step k are denoted by $u_{xi}(k)$ and $u_{yi}(k)$ along the x and y coordinates, respectively.

The borders of the environment, as well as the rectangle-shaped obstacles can be introduced as simple constraints for the x_i and y_i state variables. Let the number of obstacles be M , each obstacle being defined by its lower-left and upper-right corners $(a_1^{(l)}, b_1^{(l)})$ and $(a_2^{(l)}, b_2^{(l)})$. This means that avoiding obstacles can be written as a set of linear constraints:

$$\{x_i < a_1^{(l)} \text{ or } x_i > a_2^{(l)} \text{ or } y_i < b_1^{(l)} \text{ or } y_i > b_2^{(l)}, \quad \forall i, l\}$$

Collision-avoidance between agents is realized using a threshold value δ and the constraints $|x_i(k) - x_j(k)| > \delta \quad \forall i \neq j$. Note that due to the discrete-time model, this threshold must be carefully chosen considering maximum vehicle speed.

The planning itself is run on a finite time horizon, consisting of T steps. It is assumed, that all vehicles can reach their goal state within this time (given a reasonably large T). However, the computation should work even if some of them is not able to fulfil this constraint.

The optimization problem is minimizing the following objective function:

$$J(x, u) = \sum_{k=1}^T \sum_{i=1}^N (|x_i(k) - x_{ti}| + |y_i(k) - y_{ti}|),$$

subject to the collision, obstacle-avoidance and possible additional constraints, where x_{ti} and y_{ti} denote the prescribed target coordinates corresponding to agent i .

3. ONLINE ROUTE PLANNING BY TEMPORAL LOGIC CONSTRAINED OPTIMIZATION

Based on the problem formulation above, it can be clearly seen, that solving the routing problem requires at least $4NT$ variables, with $\mathcal{O}(2N^2) + \mathcal{O}(4NM)$ constraints, resulting from vehicle-vehicle collisions and obstacle avoidance respectively. Feasibility of obtaining solution in real-time is highly dependent on the applied solver, and on the constraints themselves.

Our experiments showed that the problem scales badly when the number of agents or the number of obstacles is increased. To overcome the issue, a two phase planning is proposed.

The double-integrator model is used only in the second phase, when a $\mathcal{O}(2 * N^2)$ part of the constraints can be omitted. Moreover, the second phase can be run individually for all agents, each having only $4T$ state variables and only $4M$ constraints resulting from obstacle avoidance (although some constraints for following the first-phase must be added, as described below).

3.1 First phase planning

In the first phase, a coarse route is designed with a relatively large sampling time $t_s^{(1)}$ to have as short prediction horizon as possible. Moreover, only the coordinates of the agents are taken into consideration, and the computed input $(\bar{u}_{xi}, \bar{u}_{yi})$ is the required coordinate change in each direction for agent i in each time step. This results in simpler dependency between the input and the desired output, and considerably reduces the required computation time.

$$\begin{cases} x_i(k+1) = x_i(k) + \bar{u}_{xi}(k) \\ y_i(k+1) = y_i(k) + \bar{u}_{yi}(k) \\ |\bar{u}_{xi}(k)| < K \\ |\bar{u}_{yi}(k)| < K \end{cases} \quad i = 1 \dots N$$

where K is an upper bound for coordinate changes which is determined using the maximum speed of the vehicle and the sampling time t_s .

The description of the rules for such a simple system is quite straightforward using temporal logic. Given a rectangle shaped obstacle as defined above, the STL formula for avoidance looks like:

$$\bigcirc_{[0, T]} (x_i < a_1^{(l)} \text{ or } x_i > a_2^{(l)} \text{ or } y_i < b_1^{(l)} \text{ or } y_i > b_2^{(l)})$$

. For avoiding collisions between the vehicles, we use:

$$\bigcirc_{[0, T]} (|y_i(t) - y_j(t)| > dist \text{ or } |x_i(t) - x_j(t)| > dist)$$

. (Here, the notation \bigcirc is the so-called *always* operator, which means, that the formula must be true in all time instants in the interval $[1, T]$ - which is the whole time of the planning in our case.)

3.2 Second phase planning

The double-integrator model for the agents is used in the second phase, where we assume, that agent i has a source and a target point, as well as a set of intermediary route points $P_i = \{(x_{pi}, y_{pi}) | p = 1 \dots T - 1\}$ known from the previous phase. The planning is done using a rolling horizon approach, using a smaller sampling time $t_s^{(2)} = t_s^{(1)}/k, k > 1$. In the simplest case this means that at route point p , the $p + 2$ -th point is set as a target. We use a temporal logic formula to ensure that the agent will be in the prescribed proximity of the point $p + 1$, and planning is done on a horizon of $T_2 = 2 \cdot k$. This can be easily extended to include more intermediary route points. This planning step is repeated when the agent reaches the route point $p + 1$. It can be shown, that by choosing this Δ proximity correctly, we can guarantee, that no collision between the agents can occur. The concept of the proof is that if the distance threshold from the first phase is equal to the half of the distance, an agent can move in the rough plan in one step ($\delta = K/2$), and we restrict the agents to remain in the $\Delta = \delta/2$ proximity of the given point for the respective time interval, then the agents are always located inside different non-overlapping $\delta \times \delta$ -sized squares. The concept is illustrated in Figure 1.

4. ILLUSTRATIVE EXAMPLE

In order to illustrate the concept, a case study containing ten agents was carried out. As it is visible in Fig. 2, we have a 16×10 units floorplan, containing 7 obstacles that must be avoided. The first-phase planning is running on a 40 step long horizon, which means at least $10 \times 40 \times 2 = 800$ continuous variables (x and y directions for all agents and all discrete time points). The number of temporal logic formulas is 10×7 (obstacles) + $10 \times 9/2$ (collisions) = 115, resulting in approximately 400×40 constraints for the optimization problem. The exact number of variables in our simulation (produced by the STL parser) was 10660 continuous and 33110 integer variables. The solution of the problem on an average consumer-grade laptop using two processor cores was 217.7 seconds.

To illustrate the detailed (second-phase) planning, we show the planned routes and the generated input signals for only one agent corresponding to obstacle avoidance. The situation is shown in Fig. 3. Here, we have the first-phase plan generated for the agents, and we use the second phase computation to calculate the detailed plan. The discretization time for the second phase was $t_s^{(2)} = t_s^{(1)}/10$, with a horizon of $T = 50$ steps. The rough plan is marked by the blue circles and consists of 5 intermediary points. The detailed plan is marked by red stars. As it is visible in the figure, the agent remains in the prescribed proximity of the first-phase plan. It must be noticed, that the agent correctly avoids the obstacle's corner, which was to be hit following directly the first phase rough route. The input signals (acceleration commands) generated for the agent are displayed in Fig. 4

5. CONCLUSION

In this paper, we presented a possible way for describing route planning problems as optimisation problems, using the formalism offered by signal temporal logic. The planning phase is divided into two parts: in the first phase, we use a low-complexity model for creating a rough plan,

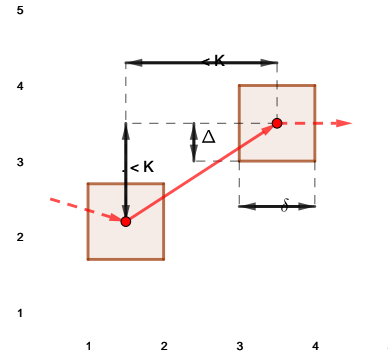


Figure 1: Distances kept by the agents during the first and second phase planning

taking into consideration the obstacles and vehicle-vehicle interactions as well. The algorithm calculates a set of consecutive intermediary route points for each agent, ensuring conflict-free behavior provided that agents are in the given proximity of the points for the respective time interval. In the second phase, each agent computes its own path, considering the points and intervals given in the first phase. Thus, vehicle-vehicle interactions need not to be checked on the detailed planning level, only smooth maneuvering and obstacle avoidance between the points are required.

Acknowledgements

This work has been supported by the research program titled "Exploring the Mathematical Foundations of Artificial Intelligence" (2018-1.2.1-NKP-00008). The partial support of the projects GINOP-2.3.2-15-2016-00002 and EFOP-3.6.3-VEKOP-16-2017-00002 is also acknowledged. T. Péni is the grantee of the Bolyai János Research Scholarship of the Hungarian Academy of Sciences. The work has been also supported by the UNKP-19-4-BME-29 New National Excellence Program of the Ministry of Human Capacities.

6. REFERENCES

- [1] Gawrilow, E., Köhler, E. and Möhring, R. and Stenzel, B. Dynamic Routing of Automated Guided Vehicles in Real-time (2008). In: *Krebs H.J., Jäger W. (eds) Mathematics – Key Technology for the Future. Springer, Berlin, Heidelberg* DOI: 10.1007/978-3-540-77203-3_12.
- [2] B. Csutak, T. Péni and G. Szederkényi. An optimization based algorithm for conflict-free navigation of autonomous guided vehicles In *Pannonian Conference on Advances in Information Technology* (2019), pp. 90-97.
- [3] T. Nishi, M. Ando and M. Konishi. Distributed route planning for multiple mobile robots using an augmented Lagrangian decomposition and coordination technique In *IEEE Transactions on Robotics*, vol. 21, no. 6, pp. 1191-1200, Dec. 2005.
- [4] E. S. Kim, S. Sadraddini, C. Belta, M. Arcaç and S. A. Seshia. Dynamic contracts for distributed temporal logic control of traffic networks In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, Melbourne, VIC, 2017, pp. 3640-3645.
- [5] Raman, V. and Donzé, A. and Maasoumy, M. and Murray, R. and Sangiovanni-Vincentelli, A. and A Seshia, S. (2014). Model Predictive Control with Signal Temporal Logic Specifications In *53rd IEEE Conference on Decision and Control* (2014): 81-87.
- [6] A. Donzé and V. Raman. BluSTL Controller Synthesis from

- Signal Temporal Logic Specifications In *1st and 2nd International Workshop on Applied verification for Continuous and Hybrid Systems*, 2015
- [7] S. Coogan and M. Arcak. Freeway traffic control from linear temporal logic specifications In *2014 ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS)*, Berlin, 2014, pp. 36-47.
- [8] N. Mehr, D. Sadigh, R. Horowitz, S. S. Sastry and S. A. Seshia. Stochastic predictive freeway ramp metering from Signal Temporal Logic specifications In *2017 American Control Conference (ACC)*, Seattle, WA, 2017, pp. 4884-4889.
- [9] S. Karaman, and E. Frazzoli. Linear temporal logic vehicle routing with applications to multi-UAV mission planning. *International Journal of Robust and Nonlinear Control*, 21:1372–1395, 2011.
- [10] C. I. Vasile, and C. Belta. Sampling-Based Temporal Logic Path Planning. In *Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 4817-4822, 2013.
- [11] Z. Liu, J. Dai, B. Wu, and H. Lin. Communication-aware Motion Planning for Multi-agent Systems from Signal Temporal Logic Specifications. In *Proceedings of the 2017 American Control Conference*, 2516-2521, 2017.

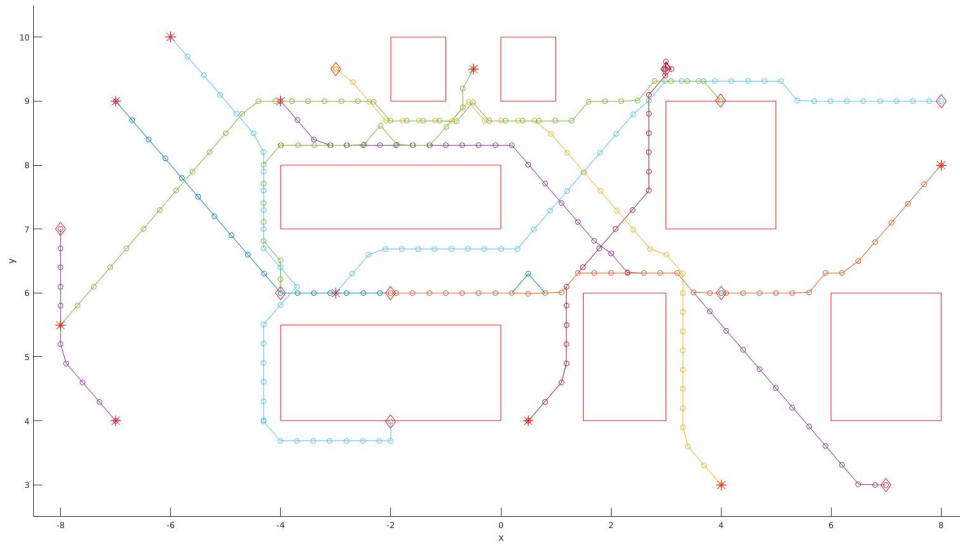


Figure 2: Paths created by the coarse first-phase planning for 10 agents
 The start and target positions of the agents are marked by stars and diamonds, respectively.

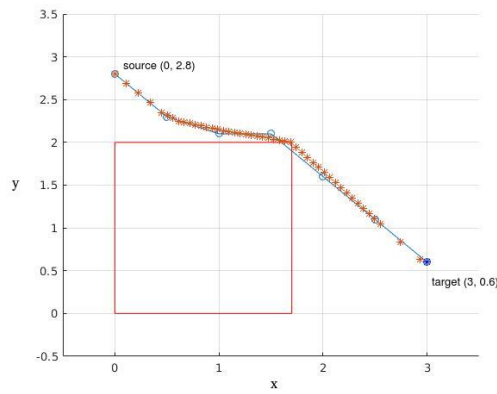


Figure 3: The first- and second phase plans for one agent

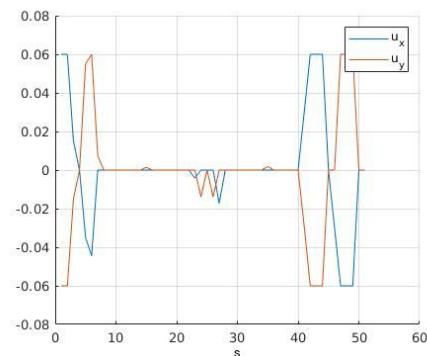


Figure 4: Input signals computed for the agent