MITSUBISHI ELECTRIC RESEARCH LABORATORIES
https://www.merl.com

Truck Fleet Coordination for Warehouse Trailer Management by Temporal Logic with Energy Constraints

Cardona, Gustavo; Vasile, Cristian-Ioan; Di Cairano, Stefano

TR2025-103 July 09, 2025

Abstract

We consider the coordination of a fleet of tractor trucks to manage trailers in a large warehouse complex and propose an approach that leverages Metric Temporal Logic (MTL) to describe missions to be executed. Each mission includes multiple tasks, such as reaching a trailer, connecting to it, moving it to a sequence of specific warehouse regions, such as loading docks, internal holding areas, and departure parking lots, and eventually disconnecting from it. The electric- powered tractor trucks must also be recharged by visiting charging stations. The MTL formulation avoids an operator manually designing a mission specification, which can quickly become unfeasible with many requests and possible assignments of tractor trucks. MTL specifications and motion dynamics are formulated as a mixed integer linear programming (MILP) approach, where the cost function includes performance ob- jectives such as minimizing the trailer motions and energy- efficient usage. Since missions are added and removed during operation and to also reduce the computation time, we modify the method to allow for a receding horizon approach that allows for partial satisfaction of the MTL specification and uses the cost function to favor the progress towards completion of partially satisfied specifications. We compare different MILP formulations in simulations.

American Control Conference (ACC) 2025

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

Truck Fleet Coordination for Warehouse Trailer Management by Temporal Logic with Energy Constraints

Gustavo A. Cardona, Cristian-Ioan Vasile, and Stefano Di Cairano

***Abstract*— We consider the coordination of a fleet of tractor trucks to manage trailers in a large warehouse complex and propose an approach that leverages Metric Temporal Logic (MTL) to describe missions to be executed. Each mission includes multiple tasks, such as reaching a trailer, connecting to it, moving it to a sequence of specific warehouse regions, such as loading docks, internal holding areas, and departure parking lots, and eventually disconnecting from it. The electric-powered tractor trucks must also be recharged by visiting charging stations. The MTL formulation avoids an operator manually designing a mission specification, which can quickly become unfeasible with many requests and possible assignments of tractor trucks. MTL specifications and motion dynamics are formulated as a mixed integer linear programming (MILP) approach, where the cost function includes performance objectives such as minimizing the trailer motions and energy-efficient usage. Since missions are added and removed during operation and to also reduce the computation time, we modify the method to allow for a receding horizon approach that allows for partial satisfaction of the MTL specification and uses the cost function to favor the progress towards completion of partially satisfied specifications. We compare different MILP formulations in simulations.**

I. Introduction

Modern supply chains include several warehouses [1], that are critical junctions of the chains where materials from upstream locations, e.g., suppliers, arrive, are re-packaged, and shipped to several downstream locations, e.g., retail stores. The inbound and outbound goods are placed in trailers that are moved throughout the warehouse area for loading, unloading, holding, or simply transferring. Specially designed tractor trucks, from now on, trucks for short, are used to execute the requested trailer operations, called missions. The missions are decomposed in a sequence of standard tasks, including reaching to and departing from a trailer, connecting to and disconnecting from it, and moving it to different warehouse areas, such as unloading and loading bays, holding and departing areas, and positioning appropriately in the assigned locations. Since trucks are increasingly electric powered, their battery discharge must be considered in the mission execution as well as the need to periodically re-charge them at specific, and often limited, charging stations. The trailer management shares some features with the coordination of robots for managing storage inside the warehouse [2]–[4]. However, some of the constraints in the trailer operation are

Gustavo A. Cardona and Cristian-Ioan Vasile are with the Mechanical Engineering and Mechanics Department at Lehigh University, PA, USA: {gcardona, cvasile}@lehigh.edu

Stefano Di Cairano is with the Mitsubishi Electric Research Laboratories, Cambridge, MA, 02139, USA: dicairano@merl.com. This work was done during Gustavo's internship at Mitsubishi Electric Research Laboratories.

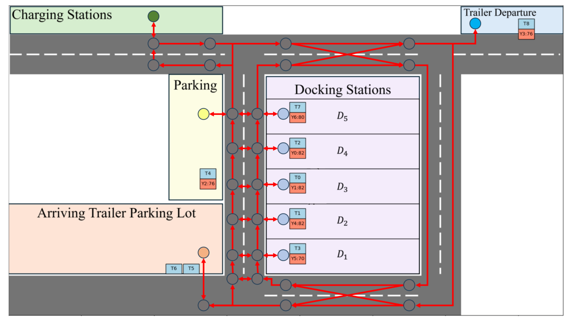


Fig. 1: Warehouse environment with its corresponding abstracted transition system.

specific to the problem considered here. It also shares some features with the traffic scheduling problems, see, e.g., the surveys in [5], [6] and [7], [8], which are also the basis of this work, but with the additional complexity of the management of the trailer, multiple operating modes, and the need to account for battery charging. The overall management of the warehouse area may be fairly complex since it may involve hundreds of trailers and tens of trucks at any time and is a dynamic problem in which new trailers and missions constantly arrive. The manual formulation of the missions, truck allocations, and charging scheduling as a numerical problem may be extremely challenging. Instead, temporal logic formalisms have emerged as powerful tools for modeling and addressing these challenges. Given their rich expressiveness and structure to capturing dynamic behaviors, task constraints, and time-sensitive goals [9]. This paper focuses on task allocation, route planning, and charging scheduling for trucks to manage trailer missions in a warehouse. The trucks may be autonomous and follow the obtained plan autonomously, but they can also be human drivers when the operator receives and follows the obtained plan manually. Linear temporal logic (LTL) has been considered for specification of similar allocation and scheduling tasks [10], [11]. After translating the specification into an automaton and constructing the Cartesian product with a transition system abstracting the warehouse, they compute a solution using graph search methods. However, the approach has scalability issues when considering a larger number of trailers and trucks and large warehouses. Here, we use specifications in Metric Temporal Logic (MTL), which, unlike LTL, allows to express an explicit definition of time and convert them into numerical problems that can be solved by Mixed Integer Linear Programming (MILP), including in the cost function performance criteria such as the minimization of the truck motion and energy consumption.

MILPs are well known to have combinatorial worst-case complexity. However, effective solvers exist, providing solutions to several classes of problems in a reasonable time, especially when the problem structure can be exploited. In this paper, we leverage the formulation as network flow problems, which are known to lead to efficient encodings. Network flow representations have been used to capture system dynamics and to ensure that scheduling and allocation constraints are met [12]. The resulting MILP encodings can be efficiently solved using off-the-shelf tools such as Gurobi. Here, we describe the motion of trailers and trucks by modeling flow networks on a transition system that abstracts the environment as shown in Fig. 1. We also couple the flow of trailers to the flow of trucks and constrain energy usage.

The coordination of trailers in the warehouse is a dynamic problem in which new tasks are constantly added, and some may take a long time to complete. Hence, it is challenging to design the temporal specification to capture all requests and ensure that it is possible within a given fixed horizon to complete all the missions, i.e., to ensure that the temporal logic formulae are feasible, for all the states in which the trucks and trailer may be in. To address this issue, we formulate a receding horizon approach [13], [14] of the problem that considers an automatic generation of the specification and iteratively satisfies the mission. To track the mission's satisfaction, we formulate a partial satisfaction encoding and design a cost function that encourages progress toward satisfaction at every iteration.

The main contributions of this work are: (i) the MTL formulation of the coordination of trucks to execute the trailer missions in the warehouse, while also accounting for truck battery status; (ii) the translation of the formulation into an effective MILP that captures the trailer and truck dynamics constraints, minimizes the trailers motion and encourages truck energy efficiency; (iii) a receding horizon approach to solve the problem, which avoids the requirement of a feasible specification and finds a solution faster, with a limited impact on extending the mission duration.

II. Notation and Preliminaries

1) Notation: Let $\mathbb{R}$ denote the set of all real numbers, $\mathbb{Z}$ the set of integers, $\mathbb{B}$ the binary set, and $\mathbb{Z}_{\geq 0}$ the set of non-negative integers. For a set $\mathcal{S}$, $2^{\mathcal{S}}$ and $|\mathcal{S}|$ represent its power set and cardinality, and $\alpha + S = \{\alpha + x \mid x \in S\}$. The integer interval (range) from a to b is $[a..b]$, and $\underline{I} = a$, $\bar{I} = b$. The j-th component of $x \in \mathbb{R}^d$ is given by x_j, $j \in [1..d]$.

2) Metric Temporal Logic: Metric Temporal Logic (MTL), first introduced in [15] is a formal specification language that expresses explicit real-time system properties. The syntax of MTL is

$$\phi ::= \top \mid \neg\phi \mid \pi \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2 \mid \Box_I \phi \mid \Diamond_I \phi,$$

where ϕ, ϕ_1, ϕ_2 are MTL formulae, $\top$ denotes the logical value *True*, and $\pi \in \Pi$ is an atomic proposition. The operators $\neg$, $\vee$, $\wedge$ are Boolean negation, disjunction, and conjunction, respectively. Additionally, $\Box_I$ and $\Diamond_I$ represent the timed operators *always* and *eventually*, with $I = [a..b]$, $0 \leq a \leq b$, denoting a discrete-time interval. The logical value *False* is expressed as $\bot = \neg\top$. The semantics of an MTL formula ϕ at time k is defined recursively over discrete-time signals $s : [0..\infty] \to 2^{\Pi}$, where $s = s(0), s(1), \ldots$ represents a sequence of sets of atomic propositions, i.e., $s(k) \in 2^{\Pi}$,

$$\begin{aligned}
(s,k) &\vDash \pi \equiv \pi \in s(k),\\
(s,k) &\vDash \neg\phi \equiv (s,k) \nvDash \phi,\\
(s,k) &\vDash \phi_1 \wedge \phi_2 \equiv (s,k) \vDash \phi_1 \wedge (s,k) \vDash \phi_2,\\
(s,k) &\vDash \phi_1 \vee \phi_2 \equiv (s,k) \vDash \phi_1 \vee (s,k) \vDash \phi_2,\\
(s,k) &\vDash \Diamond_I \phi \equiv \exists k' \in k + I, (s,k') \vDash \phi,\\
(s,k) &\vDash \Box_I \phi \equiv \forall k' \in k + I, (s,k') \vDash \phi.
\end{aligned} \tag{1}$$

The symbols $\vDash$, $\nvDash$, $\equiv$ represent satisfaction, violation, and equivalence, respectively. A discrete-time signal s satisfies the formula ϕ, denoted by $s \vDash \phi$, if and only if $(s,0) \vDash \phi$

3) Time Horizon of MTL formula [16]:

$$\|\phi\| = \begin{cases} 0, & \text{if } \phi = \pi,\\ \|\phi_1\|, & \text{if } \phi = \neg\phi_1,\\ \max\{\|\phi_1\|, \|\phi_2\|\}, & \text{if } \phi \in \{\phi_1 \wedge \phi_2, \phi_1 \vee \phi_2\},\\ \|\phi\| + \bar{I}, & \text{if } \phi \in \{\Box_I \phi, \Diamond_I \phi\}. \end{cases} \tag{2}$$

III. Problem Formulation

In this section, we present the problem of coordinating a fleet of trucks to manage trailer operations in a warehouse. The trucks pick up, transport, position, and release trailers at various locations, such as arrival, loading, unloading, holding, and departure areas. Requests specifying where trailers must be placed are defined using MTL. The truck batteries deplete while operating; hence, truck operations must also account for recharging and maintaining sufficient battery levels for the entire fleet. Next, we describe the models for the environment, trucks, trailers, and task requests.

Definition 1 (Environment)**.** *The environment is abstracted as a weighted transition system, represented by the tuple* $\mathcal{M} = (\mathcal{Q}, \mathcal{E}, \mathcal{W}, \Pi, \mathcal{L})$, *where* $\mathcal{Q}$ *denotes a finite set of locations of interest (states), and* $\mathcal{E} \subseteq \mathcal{Q} \times \mathcal{Q}$ *captures the possible transitions between locations. The function* $\mathcal{W} : \mathcal{E} \to \mathbb{Z}_{\geq 1}$ *assigns a travel duration to each transition, while* Π *is a set of atomic propositions that label the states in* $\mathcal{Q}$. *The state-labeling function* $\mathcal{L} : \mathcal{Q} \to 2^{\Pi}$ *associates each state with a subset of atomic propositions. A stationary truck at* $q \in \mathcal{Q}$ *is modeled by a unit-weight self-transition, i.e.,* $(q,q) \in \mathcal{E}$ *for all* $q \in \mathcal{Q}$, *with* $\mathcal{W}((q,q)) = 1$.

Fig. 1 shows a warehouse environment with multiple regions of interest, such as arriving trailer parking lot, trailer departure location, charging stations, temporary parking, roads, and multiple docking (i.e., loading/unloading) stations and its corresponding abstract transition system.

Definition 2 (Truck)**.** *A truck* $y \in \mathcal{Y}$ *is defined by its position* $s_y(k) \in \mathcal{Q} \cup \mathcal{E}$ *and battery state* $b_y(k) \in \mathbb{R}$ *at time* $k \in [0..H]$, *where* H *is the time horizon and* $\mathcal{Y}$ *is the set of all trucks. The trajectory* $s_y : \mathbb{Z}_{\geq 0} \to \mathcal{Q} \cup \mathcal{E}$ *captures the motion of truck* y, *i.e., occupied locations* $q \in \mathcal{Q}$ *and traversed transitions* $e \in \mathcal{E}$, *in the environment* $\mathcal{M}$ *at each time* $k \in [0..H]$.

The synchronous trajectory of the truck fleet is denoted by $s_{\mathcal{Y}} : \mathcal{Z}_{\geq 0} \to (\mathcal{Q} \cup \mathcal{E})^{|\mathcal{Y}|}$. The energy cost for the fleet of trucks $\mathcal{Y}$ is $\mathcal{J}_b = \sum_{y \in \mathcal{Y}} \sum_{k=0}^{H} b_y(k)$.

Definition 3 (Trailer)**.** *A trailer $t \in \mathcal{T}$ is defined by its position $s_t(k) \in (\mathcal{Q} \setminus \mathcal{Q}_c) \cup \mathcal{E}$ at time $k \in [0..H]$, where H is the time horizon, $\mathcal{T}$ is the set of all trailers, and $\mathcal{Q}_c := \{q \mid q \in \mathcal{L}^{-1}(\pi_{charching})\}$ is the set of charging stations. The trajectory $s_t : \mathcal{Z}_{\geq 0} \to (\mathcal{Q} \setminus \mathcal{Q}_c) \cup \mathcal{E}$ captures the motion of trailer t, i.e., occupied locations $q \in \mathcal{Q}$ and traversed transitions $e \in \mathcal{E}$, in the environment $\mathcal{M}$ at each time $k \in [0..H]$.*

The trailers cannot enter charging stations, and cannot be in states labeled as "road" without a truck assigned to it, i.e., cannot be left in the middle of the road. The synchronous trajectory of all trailers is denoted by $s_{\mathcal{T}} : \mathcal{Z}_{\geq 0} \to (\mathcal{Q} \cup \mathcal{E})^{|\mathcal{T}|}$.

Let $s_t^{\mathcal{Q}} = q_0 q_1 \ldots$ be the sequence of states in $\mathcal{Q}$ visited by trajectory s_t of trailer $t \in \mathcal{T}$, and $\mathcal{E}(s_t^{\mathcal{Q}}) = ((q_\ell, q_{\ell+1}) \mid \ell \in [0..|s_t^{\mathcal{Q}}| - 1], q_\ell \neq q_{\ell+1})$ be the sequence of transitions in $\mathcal{E}$ that excludes self-transitions (q, q) that capture stationary trailers. The motion cost of all trailers is $\mathcal{J}_m = \sum_{t \in \mathcal{T}} \sum_{e \in \mathcal{E}(s_t^{\mathcal{Q}})} \mathcal{W}(e)$.

The MTL primitive units to describe the mission specifications are *requests* that capture the location to which a specific trailer must be moved, and the minimum time to be spent at that location.

Definition 4 (Request)**.** *A request is a tuple $r = (t, d, \pi_r)$, where $t \in \mathcal{T}$ is the trailer tasked to move to a region with label $\pi_r \in \Pi_r \subseteq \Pi$, where Π_r correspond to the docking stations, arrival, and departure locations, and stay there for a duration $d \in \mathbb{Z}_{\geq 1}$. The set of all requests in the mission is defined as $\mathcal{R} = \{r_1, \ldots, r_{|R|}\}$.*

Docking stations, arrival, and departure locations q_r have exactly one label from Π_r, i.e., $|\mathcal{L}(q_r)| = 1$.

A request can be translated into MTL specifications via $\phi_r = \Box_{[0,d]} \varpi_{t,q}$, where $\varpi_{t,q}$ denotes that trailer t is at location $q \in \mathcal{Q}$ labeled with $\mathcal{L}(q) = \{\pi_r\}$ and therefore serve request $r \in \mathcal{R}$.

Finally, we define the plans that satisfy an MTL specification ϕ over requests $\mathcal{R}$ (i.e., over atomic propositions $\varpi_{t,q}$).

A transportation schedule $Asg : \mathcal{T} \times [0..H] \to \mathcal{T} \cup \{\epsilon\}$ is an assignment of trucks y to trailers t at each time k, where ϵ denotes no truck is assigned. A feasible transportation is feasible if (i) motion is synchronized, $s_y(k) = s_t(k)$ whenever $y = Asg(t, k)$, and (ii) every truck pulls at most one trailer, $|\{t \mid Asg(t, k) = y\}| \leq 1$ for all $y \in \mathcal{Y}$, $k \in [0..H]$.

Definition 5 (Plan)**.** *A plan Γ is the joint state trajectory $s_{\mathcal{Y}}, s_{\mathcal{T}}$ generated by the trucks pulling trailers to and from docking, arrival, and departure locations in the environment $\mathcal{M}$, and transportation schedule Asg. Formally, we require $(s_{\mathcal{Y}}, s_{\mathcal{T}}) \vDash \phi$ such that $b_y(k) \geq 0$ for all $y \in \mathcal{Y}$ and $k \in [0..H]$, and Asg is feasible.*

Next we formally describe our problem

Problem 1. *Given a fleet of trucks $\mathcal{Y}$, an abstracted environment $\mathcal{M}$, a group of trailers $\mathcal{T}$, and an MTL specification ϕ over requests $\mathcal{R}$, find a plan Γ such that ϕ is satisfied and cost $\mathcal{J} = \mathcal{J}_b - \mathcal{J}_m$ is maximized.*

The cost function $\mathcal{J}$ tries to keep the level of energy of all trucks as high as possible and minimize the unnecessary trailer motions. While plan Γ requires only the trailer trajectories to satisfy the specification, the trailers can only be transported in the environment by a truck that pulls them.

IV. MILP Formulation for Truck Fleet Coordination

In this section, we formulate Problem 1 as a Mixed Integer Linear Programming (MILP) problem.

Motion of trucks and trailers: We model the motion of trucks and trailers as a flow network problem. Then, we constrain the flow of trailers on the flow of trucks, so a trailer can only move together with a truck, i.e., by being carried by it. Let the binary variables $Y_{q,i,k} \in \mathbb{B}$ and $U_{e,i,k} \in \mathbb{B}$ represent whether the truck $i \in \mathcal{Y}$ is at state $q \in \mathcal{Q}$ or traversing edge $e \in \mathcal{E}$ at time $k \leq \|\phi\|$, with $\|\phi\|$ time horizon of the specification computed as in (2). Similarly, $T_{q,j,k} \in \mathbb{B}$ and $V_{e,j,k} \in \mathbb{B}$ are binary variables that represent whether trailer $j \in \mathcal{T}$ is at state $q \in \mathcal{Q}$ or traversing edge $e \in \mathcal{E}$ at time $k \leq \|\phi\|$.

1) Motion of trucks: The constraints on truck flow conservation are

$$Y_{q,i,0} = |\{i \in \mathcal{Y} \mid q_{0,i} = q\}|, \tag{3a}$$

$$Y_{q,i,k} = \sum_{(q',q) \in \mathcal{E}} U_{(q',q),i,k-\mathcal{W}(q',q)} \leq N_q^y, \tag{3b}$$

$$\sum_{(q,q') \in \mathcal{E}} U_{(q,q'),i,k} = \sum_{(q',q) \in \mathcal{E}} U_{(q',q),i,k-\mathcal{W}(q',q)}, \tag{3c}$$

for all $q \in \mathcal{Q}$, $i \in \mathcal{Y}$, $k \in [0..\|\phi\|]$, where (3a) is the initial distribution of trucks in the environment, and (3b) are the trucks in state $q \in \mathcal{Q}$ at time $k \leq \|\phi\|$ considering the durations of the incoming edge transitions. Additionally, N_q^y imposes a capacity constraint that may be used to guarantee no collisions of trucks at road nodes ($N_q^y = 1$, for all $q \in \mathcal{Q}$ with label $\pi = \pi_{road}$) or the maximum number of trucks to enter same charging station node ($N_q^y \in \mathbb{Z}_{\geq 1}$, for all $q \in \mathcal{Q}$ with label $\pi = \pi_{charging}$). Finally, (3c) imposes the conservation of truck flow by requiring that the number of trucks in incoming $(q', q) \in \mathcal{E}$ and outgoing $(q, q') \in \mathcal{E}$ edges is equal.

2) Motion of trailers: The constraints on trailer flow conservation are

$$T_{q,j,0} = |\{j \in \mathcal{T} \mid q_{0,j} = q\}|, \tag{4a}$$

$$T_{q,j,k} = \sum_{(q',q) \in \mathcal{E}} V_{(q',q),j,k-\mathcal{W}(q',q)}, \tag{4b}$$

$$\sum_{(q,q') \in \mathcal{E}} V_{(q,q'),j,k} = \sum_{(q',q) \in \mathcal{E}} V_{(q',q),j,k-\mathcal{W}(q',q)}. \tag{4c}$$

for all $q \in \mathcal{Q}$, $j \in \mathcal{T}$, and $k \in [0..\|\phi\|]$, where (4a) is the initial distribution of trailers, (3b) is the number of trailers in state $q \in \mathcal{Q}$ at time $k \leq \|\phi\|$ considering the durations of the incoming edge transitions, and (3c) imposes the conservation of the trailer flow by requiring that the number of trailers in incoming $(q', q) \in \mathcal{E}$ and outgoing $(q, q') \in \mathcal{E}$ edges is equal.

3) Coupling motion of trucks and trailers: The trailers must be moved by trucks, hence we couple their motion over every edge $e = (q, q') \in \mathcal{E} \setminus \{q = q'\}$, except in self-loops since the trailer is stationary, resulting in the constraints

$$V_{e,j,k} \leq \sum_{i \in \mathcal{Y}} U_{e,i,k}, \quad \sum_{j \in \mathcal{T}} V_{e,j,k} + \sum_{i \in \mathcal{Y}} U_{e,i,k} \leq 2, \tag{5}$$

for all $k \in [0..\|\phi\|]$, where we enforce that the motion of trucks and trailers are coupled, i.e., no trailer can transition to another state if a truck is not transitioning on the same edge, and that only a truck is assigned to a trailer.

4) Avoiding undesirable trailer behaviors: In order to prevent trailers from being "temporarily abandoned" at the road nodes, we impose the constraint

$$\sum_{i \in \mathcal{Y}} Y_{q^r,i,k} \geq \sum_{j \in \mathcal{T}} T_{q^r,j,k}. \tag{6}$$

for all $q^r := \{q \mid q \in \mathcal{L}^{-1}(\pi_{road})\}$ and at all times $k \in [0..\|\phi\|]$.

Similarly, we avoid that two trucks "switch trailers" during their operations by imposing

$$\sum_{i \in \mathcal{Y}} U_{e=(q,q'),i,k} + \sum_{i \in \mathcal{Y}} U_{e=(q',q),i,k} \leq 1, \tag{7}$$

for all $k \in [0..\|\phi\|]$ and $e = (q, q') \in \mathcal{E} \setminus \{q = q'\}$ where q and q' belong to road or docking labeled nodes.

Finally, we prevent trucks from bringing trailers to charging stations by imposing

$$T_{q,j,k} = 0, \forall q \in \mathcal{L}^{-1}(\pi = \pi_{charging}). \tag{8}$$

The combination of (3)–(8) ensures that a truck picks up a trailer only if it has enough energy to move it to destination.

5) Docking maneuver: A docking maneuver occurs when a truck transporting a trailer is positioned at the loading station dock. During this maneuver, the truck must perform specific motions to align the trailer with the dock. We prohibit other trucks from crossing the road while the docking maneuver is ongoing,

$$U_{e^+,i,k} + U_{\tilde{e},i,\tilde{k}} \leq 1, \tag{9}$$

where e^+ is the set of incoming edges of all road states $q^r \in \mathcal{Q}$ neighbors of the respective loading dock except for the self-loops and docking node, $\tilde{e}$ is the edge linking the road and docking station where the maneuver is performed, and the $\tilde{k}$ is the duration of the maneuver.

Energy constraints: Here we model the energy spent by each truck as proportional to the traveled distance. Trucks can recharge at designated charging stations. This allows them to restore their energy levels and resume contributing to completing the mission. Let $E_{i,k}$ capture the remaining amount of energy a truck $i \in \mathcal{Y}$ has at time $k \in [0..\|\phi\|]$, the constraints capturing the truck energy are

$$E_{i,k+\mathcal{W}(e)} >= E_{i,k} - D_e + C_e - (1 - U_{e,i,k}) \cdot M, \tag{10a}$$

$$E_{i,k+\mathcal{W}(e)} <= E_{i,k} - D_e + C_e + (1 - U_{e,i,k}) \cdot M, \tag{10b}$$

where $C_e \in \mathbb{R}_{\geq 0}$ is the charging rate at charging stations and $D_e \in \mathbb{R}_{\geq 0}$ is the discharging rate at a specific transition $e \in \mathcal{E}$, and M is a sufficiently large number, i.e., greater than the battery capacity, the so-called Big-M [17].

Mission satisfaction: For encoding request satisfaction, we use the binary variable $z_k^r \in \mathbb{B}$ which is 1 if request $r \in \mathcal{R}$ is satisfied at time $k \in [0..\|\phi\|]$ and 0 if violated. Thus,

$$z_k^r \leq T_{q,j,k}, \tag{11}$$

for all $r = (t, d, \pi_r)$ with $\pi_r \in \Pi_r \subseteq \Pi$, $t \in \mathcal{T}$, $d \in \mathbb{Z}_{\geq 0}$ ensures that trailer $j \in \mathcal{T}$ is at location $q \in \mathcal{Q}$ at the requested time $k \in [0..\|\phi\|]$. We enforce the entire MTL specification by a recursive encoding that assigns a binary variable $z_k^\phi \in \mathbb{B}$ to each subformula ϕ at time k, such that $z_k^\phi = 1$ if and only if ϕ holds at time k. The complete encoding follows [13], and we omit it for brevity.

Cost function terms: The cost function includes multiple terms modeling different performance objectives. A trailer should move only if necessary to satisfy the mission specification. Thus, we consider the cost function term

$$\tau_T = \sum_{k=0}^{\|\phi\|} \sum_{j \in \mathcal{T}} \sum_{(q,q') \in \mathcal{E} \setminus (q,q))} V_{e,j,k}, \tag{12}$$

where τ_T captures all of the trailer motion during the mission. We normalize the cost term by the weight $\sigma_T = \lambda_T / (\|\phi\| \cdot |\mathcal{T}|)$, where $\lambda_T \in [0, 1]$ is a priority weight to define the priorities in the cost function.

For trucks, it is desirable to minimize energy usage and incentivize recharging at a charging station when the energy is low. Therefore, we consider the cost function term

$$\tau_B = \sum_{k=0}^{\|\phi\|} \sum_{i \in \mathcal{Y}} E_{i,k}, \tag{13}$$

where τ_B is the total amount of energy of the fleet of trucks available, which optimizes the overall energy usage and increases the available energy in trucks so that they are ready for the next task. We normalize the cost term by the weight $\sigma_B = \lambda_B / (\|\phi\| \cdot |\mathcal{Y}|)$, where $\lambda_T \in [0, 1]$ is another priority weight. Combining the terms, we obtain the cost function that captures the desired performance criteria as $\mathcal{J} = \sigma_B \cdot \tau_B - \sigma_T \cdot \tau_T$.

Optimization Problem: We formulate a MILP optimization problem whose solution provides a solution to Problem 1 as

$$\begin{aligned} \max_{Y,u,T,v,B,Z} \; & \mathcal{J} \\ s.t. \quad & T_{q,j,k} \models \phi, \\ & (3), (4), (5), (6), \\ & (7), (8), (9), (10). \end{aligned} \tag{14}$$

where the constraints include the satisfaction of the specification, built from (11), the motion of trucks, trailers and their coupling, the avoidance of undesirable trailer behaviors, and docking maneuver and charging. For the problem in (14), the specification must be satisfiable for the initial truck and trailer conditions and the given requests within the horizon along which (14) is formulated. Determining a horizon that ensures the satisfiability of the specifications without being excessively large to avoid negative impact on the computation time is challenging, especially for scenarios with several trailers, trucks, and requests. Thus, next we modify the problem to be solved in receding horizon.

V. MILP PROBLEM REFORMULATION: A RECEDING HORIZON APPROACH

In this section, we reformulate the MILP problem in (14) to operate in a receding horizon, which makes it easier to design specifications that are feasible since the horizon can be extended without negatively impacting the computational load. We consider that a mission specification has the fixed structure

$$\tilde{\phi} = \bigwedge_{r \in \mathcal{R}} \lozenge_{[lb,\, ub]} r(t, d, \pi_r), \tag{15}$$

for all $r \in \mathcal{R}$, with lb and ub being predefined lower and upper bound of the time interval for the *eventually* operator. Instead of considering only satisfaction or violation of a specification, we use an encoding that accounts for fractions of satisfaction, allowing the mission to be partially satisfiable. This is enabled by a recursive encoding that uses a variable $z_k^{\tilde{\phi}} \in [0,1]$ for Boolean and temporal operators capturing the percentage of satisfaction and $z_k^{\pi_r} \in \mathbb{B}$ for atomic propositions. The complete encoding follows the one in [18], and we omit it for brevity. Then, to obtain a trajectory that fully satisfies the formula, we solve (14), iteratively over a shifting time horizon until the specification is fully satisfied, which indicates that all requests have been served. However, to encourage the solution of (14) to progress in partially satisfying a specification, which moves the solution towards the full satisfaction of the specification and avoids deadlocks, we need to modify the cost function.

1) Progress towards satisfaction cost function: Each request specifies that a particular trailer must move from its current location to a designated destination. Thus, inspired by [19], we compute a set of minimal distance paths from the trailers starting positions to their targets using Dijkstra's algorithm. Let $\mathcal{P}_r$ represent the set of shortest paths for trailer t in request $r = (t, d, \pi_r) \in \mathcal{R}$, from its current location $q \in \mathcal{Q}$ to the destination labeled by the proposition $q_d = \mathcal{L}^{-1}(\pi_r)$. For each request, we compute a monotonically decreasing cost function $\Theta(r)$ along the paths in $\mathcal{P}_r$, designed to encourage the trailer to advance towards its destination progressively,

$$\Theta(q,r) = \begin{cases} c \cdot d(q, q_d), & q_d = \mathcal{L}^{-1}(\pi_r) \wedge q \in \mathcal{P}_r \\ C, & otherwise \end{cases}, \tag{16}$$

where c and $C \in \mathbb{Z}_{\geq 0}$ such that $c \leq C$ are constant bounds for the cost function, and $d(q, q_d)$ is the Dijkstra computed distance. Therefore, the cost term representing the progress towards the destination is

$$\tau_p = \sum_{k=0}^{\|\phi\|} \sum_{j \in \mathcal{T}} \sum_{q \in \mathcal{Q}} \Theta(q,r) \cdot T_{q,j,k}, \tag{17}$$

for all $r \in \mathcal{R}$. The term in (17) becomes smaller if the state gets closer to the requested goal along the set of paths. Again, we construct a normalization weight $\sigma_p = \lambda_p / (\|\tilde{\phi}\| \cdot |\mathcal{T}|)$, where $\lambda_p \in [0,1]$ is a priority weight.

2) Receding horizon with partial satisfaction MILP problem: For a receding horizon solution that progresses towards the satisfaction of the MTL by partial satisfaction and progress towards satisfaction, the cost function $\mathcal{J}$ is modified into $\mathcal{J}_R = z_0^{\tilde{\phi}} + \tau_B \cdot \sigma_B - \tau_T \cdot \sigma_T - \tau_p \cdot \sigma_p$, and as a consequence the MILP problem (14) is modified into

$$\begin{aligned} \max_{Y,u,T,v,B,Z} \ & \mathcal{J}_R \\ s.t. \quad & T_{q,j,k} \models \tilde{\phi}, \\ & (3), (4), (5), (6), \\ & (7), (8), (9), (10). \end{aligned} \tag{18}$$

In the receding horizon implementation, (18) is solved over a fixed horizon, a part of the solution is stored, and then it is solved again along a shifted horizon. Such an iterative solution may simply be a way to control the computational load, i.e., without any feedback from the actual system, which amounts to initializing the next problem from the previous problem solution. However, the receding horizon solution may also implement feedback by commanding the trucks with the initial part of the sequence computed at the current step, and then solving again at a future step over the shifted horizon, using the updated positions of trucks and trailers at that time, which may be different from the predicted one due to imperfections in execution, model errors, or unexpected external effects. Using feedback provides some degree of robustness and prevents errors from accumulating, which is especially important for long operations.

VI. CASE STUDIES

In this section, we present some case studies that show the behavior of our proposed method. The case studies were simulated on a desktop computer with 6 cores at 2.60 GHz and 16 GB of RAM, using Gurobi as MILP solver, PyTeLo [20] and ANTLRv4 for encoding the MTL specifications, and Networkx to design and log the states in the the environment.

We consider the environment and abstracted transition system shown in Fig. 1, with nodes $\pi = \{\pi_{road}, \pi_{charging}, \pi_{docks}, \pi_{parking}, \pi_{arrive}, \pi_{depart}\}$, $|\mathcal{Y}| = 7$ trucks, $|\mathcal{T}| = 9$ trailers. For $e = (q, q') \in \mathcal{E}$, $D_e = -2$ if $q \neq q'$, and $D_e = 0$ if $q = q'$, i.e., for self-loops, and $C_e = 10$ for the self-loops at the *charging* station nodes, $C_e = 0$ zero otherwise. Capacity constraints are $N_q^y = 1$ for road and dock nodes, $N_q^y = 4$ for charging station node and departure area nodes, $N_q^y = 5$ for arrival area nodes, and $N_q^y = 2$ for parking area nodes. In the cost function, we consider $\lambda_B = 0.5$, $\lambda_T = 0.1$, and $\lambda_p = 1$, which gives priority first to progress towards satisfaction, then to energy conservation, and last to minimization of unnecessary motions.

The MTL mission specification to satisfy is $\phi_{cs} = \lozenge_{[9,10]} r(t_0, 1, \pi_{D_3}) \wedge \lozenge_{[9,10]} r(t_1, 1, \pi_{D_2}) \wedge \lozenge_{[14,15]} r(t_2, 1, \pi_{D_4}) \wedge \lozenge_{[14,15]} r(t_3, 1, \pi_{D_1}) \wedge \lozenge_{[9,10]} r(t_7, 1, \pi_{D_5}) \wedge \lozenge_{[14,15]} r(t_8, 1, \pi_{depart}) \wedge \lozenge_{[16,17]} r(t_4, 1, \pi_{parking})$, where for simplicity, we set the duration of all the requests to a one-time unit.

Case study 1: Fully satisfiable mission*:* First, we consider trailers and trucks to have an initial position at time $k = 0$, as shown in Fig. 2, where trucks are represented by the red rectangles and trailers the light blue rectangles. The battery energy levels range from 0 to 100 energy units, and initially, all trucks are at full charge capacity. For the given

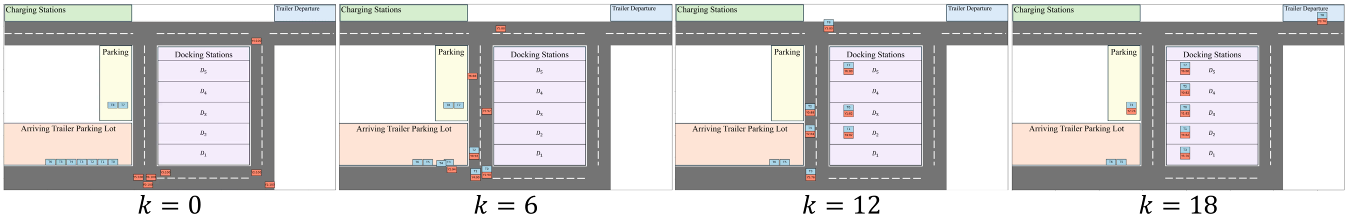


Fig. 2: Sequence of snapshots for the solution of case study 1. Trucks and trailers are red and blue rectangles, respectively.

abstracted environment and these initial conditions of trucks and trailers, ϕ_{cs} is a feasible specification. Snapshots at time instant $k = 0$, $k = 6$, $k = 12$, and $k = 18$ for the solution of the MILP (14) are shown in Fig. 2, where $\|\phi_{cs}\| = 18$. At $k = 6$, the trucks have already moved, three trailers are being transported by allocated trucks, and one truck is picking up a trailer at the arriving trailer parking area. At $k = 12$, three trailers have reached their destination, and all requested trailers are being transported by trucks. Finally, at $k = 18$, all the trailers have reached their destinations and fully satisfied the mission. The truck trajectories are collision-free since $N_q^y = 1$ for road and dock nodes.

In the same scenario, we have applied the encoding, enabling partial satisfaction of the MTL specification and obtained identical trajectories, suggesting that the partial satisfaction encoding may have a limited impact on the performance when the MTL specification is fully satisfiable. Then, we modify ϕ_{cs} according to (15) for receding horizon solution, where we arbitrarily set $lb = 0$ and $ub = 7$ for all requests, and run the MILP optimization problem in (18). The trajectories this way differ slightly from those obtained from (14). After three iterations, the specification is fully satisfied and at $\|\tilde{\phi}_{cs}\| = 24$. The receding horizon approach does not require the temporal specifications to be feasible, as, in fact, it will be impossible to satisfy all of them within $ub = 7$, but this comes at the expense of increased time to complete the overall mission. However, when specifications are more complex and change dynamically during execution, e.g., due to the introduction of new requests, the reduction in performance may be a worthwhile price to pay for not requiring that the specification be initially feasible.

Case study 2: Unfeasible mission: Next, we consider a scenario where the initial conditions are the same as in the previous case study, but three trucks are placed close to the charging stations. All trucks' initial battery energy levels are now 40 energy units, which is not enough to execute the entire mission. In this scenario, the MILP (14) is infeasible since there does not exist a trajectory that can satisfy the mission in the requested time, also reinforcing that guaranteeing feasibility for a specification for all initial conditions may be hard. For the MILP (14) with partial satisfaction encoding, the solution satisfies around 85% of the specification and, specifically, six out of seven requests. For the receding horizon approach solving MILP (18), we consider two different instances, one where the cost function of (18) includes the progress term (17), and another without it. The satisfaction rate with respect to the iteration instance is shown in Fig. 3 for both cases. The instance with the progress term in the cost function considered only requires 4 iterations to reach full satisfaction, while the instance without such terms requires 11 iterations to reach full satisfaction. Accordingly, there is a significant difference in the mission completion time, which is 32 time units for the case with the progress term in the case, and 88 time units without it. Thus, the progress term appears effective in promoting progress towards the goal, and reducing both the number of iterations and the time to complete the mission. A sequence of snapshots of the solution is shown in Fig. 4, where the initial conditions at time $k = 0$ are first shown. The trucks close to the charging station go charging until there is enough energy to serve all the requests. At time $k = 20$, some of the requests have been satisfied, and all available trucks are working to satisfy the remaining requests, except for one truck that is unnecessary for the mission and is commanded to recharge and stay out of the main road. At time $k = 32$, all requests have been effectively satisfied. It is important to notice again that 32 is not the optimal time to complete the mission, as this is not guaranteed by the receding horizon solution. In fact, for different parameters, a solution could be computed in only 3 iterations, with a lower completion time of 24 time units.

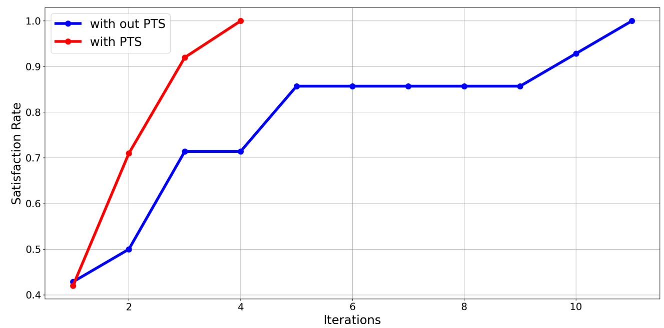
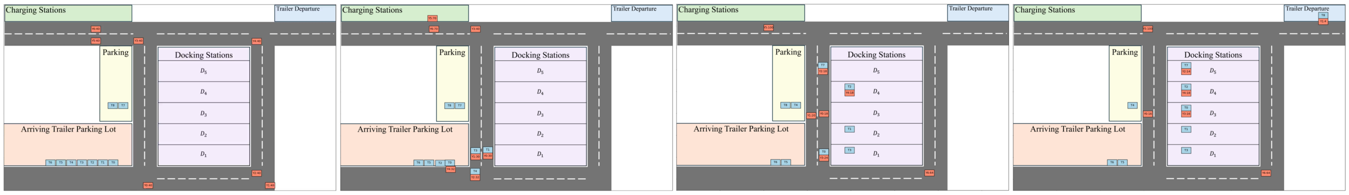


Fig. 3: Rate of satisfaction per iteration for case study 2.

Methods performance comparison: Here, we compare the computation time for the solutions of *case study 1* and *case study 2* for different combinations of previously described methods. The results are shown in Fig. 5, where "FS" stands for full satisfaction encoding, i.e., MILP problem in (14), "PS" stands for the partial satisfaction encoding, "PS+progress" is the partial satisfaction encoding with the addition of the progress term in the cost function, "RH+PS" is the receding horizon approach with partial satisfaction encoding, i.e., (18) without progress term, and "RH+PS+Progress" also include the progress term. For the



$k = 0$ $k = 5$ $k = 20$ $k = 32$

Fig. 4: Sequence of snapshots for the solution of case study 2.

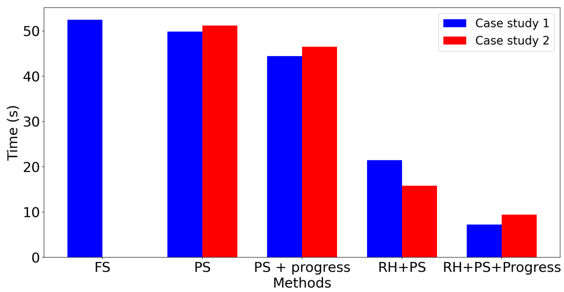


Fig. 5: Time performance comparison for both case studies, with different combinations of MILP problems.

second case, there is no data in the full satisfaction encoding since the problem is unfeasible. For both case studies, the MILP (18) is significantly faster to solve than (14). This difference in the methods' performance is expected to be even larger when the number of requests, the size of the environment, and the number of trailers and trucks grow. The computation time is particularly sensitive to growing the environment, since most of the MILP variables scale with the number of states and edges in the transition system.

VII. Conclusions

This paper proposes a method to coordinate multiple trucks, autonomous or manually driven by operators following centralized directions, in managing the operations of warehouse trailers. Trucks transport trailers in the environment to satisfy requests to carry a specific trailer to a specific location in the warehouse site. We consider the truck's energy consumption by modeling the battery discharge during motion and allowing for recharging at charging stations. We have formulated the requests as the primitive unit of a mission specification expressed using MTL, combined with network flows to capture and impose constraints on the motion of trucks and trailers. We have proposed different encodings based on full and partial satisfaction and allowing receding-horizon solutions where an additional term of the cost function encourages making decisions toward satisfying requests.

References

[1] I. H. Mohamud, M. A. Kafi, S. A. Shahron, N. Zainuddin, and S. Musa, "The role of warehouse layout and operations in warehouse efficiency: A literature review.," *Journal Européen des Systèmes Automatisés*, vol. 56, no. 1, 2023.

[2] N. Seenu, K. C. RM, M. Ramya, and M. N. Janardhanan, "Review on state-of-the-art dynamic task allocation strategies for multiple-robot systems," *Industrial Robot: the international journal of robotics research and application*, vol. 47, no. 6, pp. 929–942, 2020.

[3] A. Bolu and Ö. Korçak, "Adaptive task planning for multi-robot smart warehouse," *IEEE Access*, vol. 9, pp. 27346–27358, 2021.

[4] K. Sharma and R. Doriya, "Coordination of multi-robot path planning for warehouse application using smart approach for identifying destinations," *Intelligent Service Robotics*, vol. 14, no. 2, pp. 313–325, 2021.

[5] M. Eom and B.-I. Kim, "The traffic signal control problem for intersections: a review," *European Transport Research Review*, vol. 12, p. 50, Sep 2020.

[6] J. Rios-Torres and A. A. Malikopoulos, "A survey on the coordination of connected and automated vehicles at intersections and merging at highway on-ramps," *IEEE Trans. Intell. Transp. Systems*, vol. 18, no. 5, pp. 1066–1077, 2017.

[7] R. Firoozi, R. Quirynen, and S. Di Cairano, "Coordination of autonomous vehicles and dynamic traffic rules in mixed automated/manual traffic," in *2022 American Control Conference (ACC)*, pp. 1030–1035, 2022.

[8] N. Suriyarachchi, R. Quirynen, J. S. Baras, and S. Di Cairano, "Optimization-based coordination and control of traffic lights and mixed traffic in multi-intersection environments," in *2023 American Control Conference (ACC)*, pp. 3162–3168, 2023.

[9] Y. Yang, X. Yin, and S. Li, "A distributed framework for multi-robot task planning with temporal logic specifications," in *2020 IEEE 16th International Conference on Control & Automation (ICCA)*, pp. 570–575, IEEE, 2020.

[10] S. Hao, Z. Huang, L. Wang, R. Zhang, X. Zhang, J. Peng, and W. Yu, "An optimal task decision method for a warehouse robot with multiple tasks based on linear temporal logic," in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 1453–1458, IEEE, 2017.

[11] Y. Yang, J. Zhao, X. Yin, and S. Li, "Formal synthesis of warehouse robotic systems with temporal logic specifications," in *2019 IEEE International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM)*, pp. 469–474, IEEE, 2019.

[12] G. A. Cardona and C.-I. Vasile, "Planning for heterogeneous teams of robots with temporal logic, capability, and resource constraints," *The International Journal of Robotics Research*, p. 02783649241247285, 2024.

[13] S. Sadraddini and C. Belta, "Robust temporal logic model predictive control," in *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 772–779, IEEE, 2015.

[14] T. Wongpiromsarn, U. Topcu, and R. M. Murray, "Receding horizon control for temporal logic specifications," in *Proceedings of the 13th ACM international conference on Hybrid systems: computation and control*, pp. 101–110, 2010.

[15] C. Brzoska, "Programming in metric temporal logic," *Theoretical computer science*, vol. 202, no. 1-2, pp. 55–125, 1998.

[16] A. Dokhanchi, B. Hoxha, and G. Fainekos, "On-line monitoring for temporal logic robustness," in *International Conference on Runtime Verification*, pp. 231–246, Springer, 2014.

[17] G. L. Nemhauser and L. A. Wolsey, *Integer and Combinatorial Optimization*. New York, NY, USA: Wiley-Interscience, 1988.

[18] G. A. Cardona and C.-I. Vasile, "Preferences on partial satisfaction using weighted signal temporal logic specifications," in *2023 European Control Conference (ECC)*, pp. 1–6, IEEE, 2023.

[19] S. Di Cairano, W. M. H. Heemels, M. Lazar, and A. Bemporad, "Stabilizing dynamic controllers for hybrid systems: A hybrid control lyapunov function approach," *IEEE Transactions on Automatic Control*, vol. 59, no. 10, pp. 2629–2643, 2014.

[20] G. A. Cardona, K. Leahy, M. Mann, and C.-I. Vasile, "A flexible and efficient temporal logic tool for python: Pytelo," *arXiv preprint arXiv:2310.08714*, 2023.