Delft University of Technology

Multi-Machine Coordination for Logistics

ME44300 (2024/25)

Project Assignment

Authors:

5173388	Roeland van Hagen - MT
5096375	Evert Ingwersen - TIL
5076870	Sophie Timmerman - TIL
5097304	Berend de Vries - MT

June 15, 2025

Contents

1 Part 1

In this chapter, a definition of a double stacked container train will described. Firstly, the main missions of this system will be mentioned. After, some suggestions for making the system more intelligent will be set out. The control objectives for these more intelligent options are in paragraph 1.3. Then, the state-space model is formulated and described. Lastly, the formal definitions to evaluate the performance will be formulated in paragraph 1.5.

1.1 Main Missions

A double stack container train is a means of transport for containers typically originating from a port or other large transport hub as shown in figure 1. A double stack container train usually consists of one or two locomotive which provide the power for the train. The rest of the train consists of wagons which can accommodate two forty foot container units stacked on top of each other or a combination of four twenty foot container units. These trains have the main mission of transporting a large amount of containers on one train (Ruf et al., 2022; Upadhyay et al., 2017). To complete this mission it has the following operational tasks.

- Needs to transport cargo at desired speed and power
- Needs to be stable
- Wagons need to be coupled and decoupled easily

The physical structure of the double stack container train would consist of different systems onboard of the train locomotive. The train locomotive uses a diesel engine to generate power for the traction motors to power the wheels . For braking of the locomotive and the locomotive's and car's air brakes are used. The air for this is compressed by a compressor on the locomotive. This air is then provided to all the cars in the train (PRC Rail Consulting Ltd, 2019).

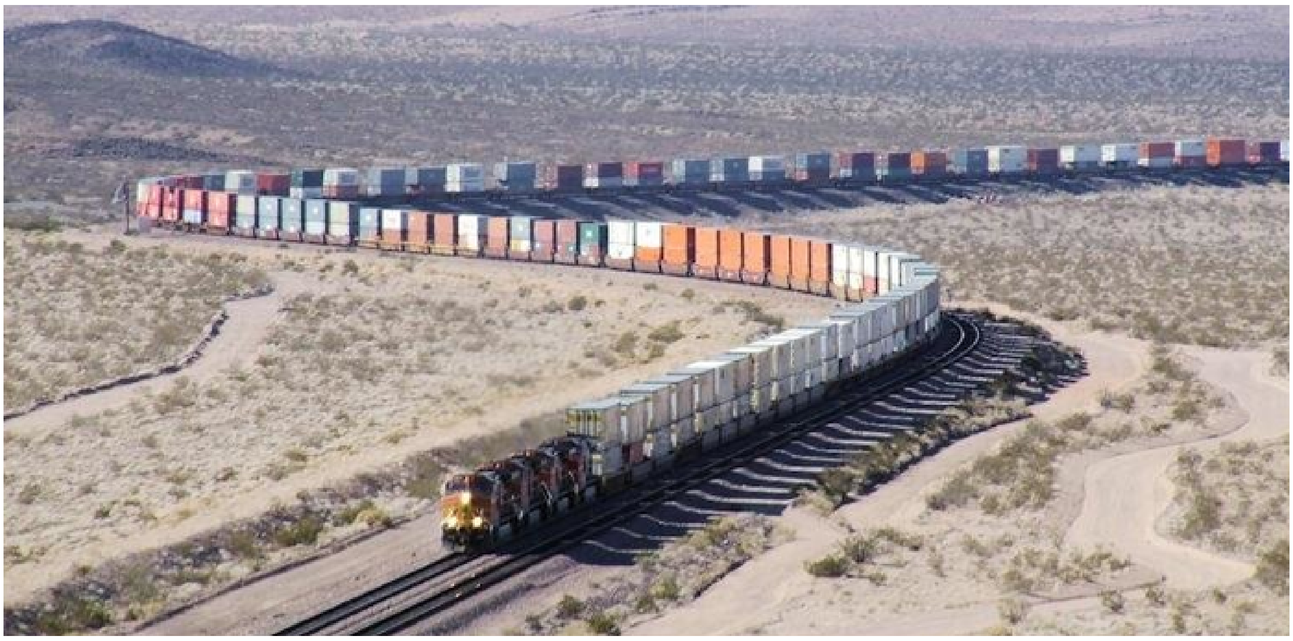

Figure 1: A double-stacked container train en route (Joshua, 2015)

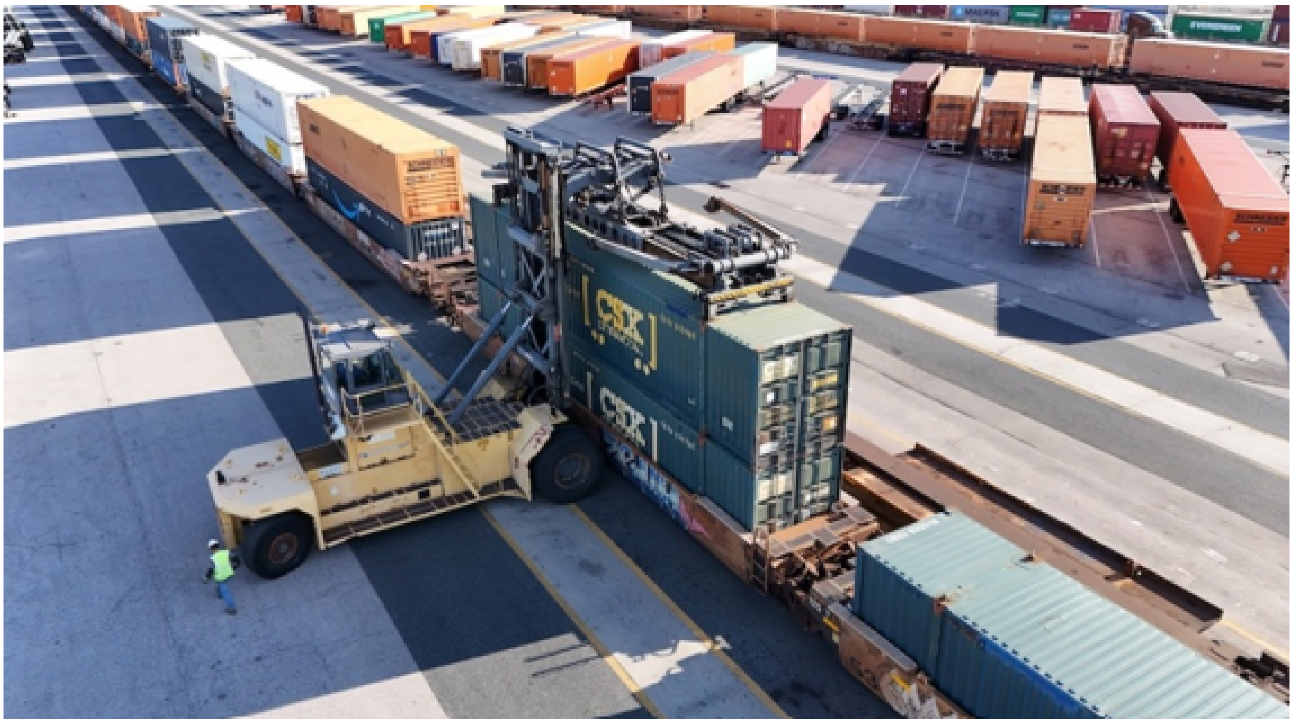

Figure 2: A double-stacked container train in a port terminal (Stephens, 2024)

1.2 Motivation for Intelligent Control

The motivation to make the double stack container train more intelligent is driven by the need to improve efficiency, safety, and autonomy in freight transport operations. Currently, many of the actions such as driving, coupling/decoupling, and positioning for loading and unloading currently require human input, which results in delays, increased labor costs, and the possibility of human error. Intelligent operation would allow the system to:

- Drive autonomously to reduce dependence on human drivers who are constrained by labor laws and fatigue, and improve overall operational efficiency.
- Synchronize its position with cranes or other infrastructure for autonomous loading/unloading, minimizing idle time and increasing terminal throughput.
- Continuously monitor and adjust for stability, for example, when moving double-stacked containers over uneven rails or at higher speeds.

1.3 Control Objectives

Since a train drives on tracks and switches are often operated remotely, the main control objective of driving a train autonomously is reaching and maintaining a desired speed, i.e. the maximum speed on that part of track or the maximum safe speed to operate the train.

When loading and unloading autonomously, speed is not the control objective, but a constraint. The control objective is the position the train needs to have to line up with a crane, a maximum speed on that part of track would be a constraint in how fast the train can move to obtain the control objective.

The control objective for stabilization would be to try to minimize angle accelerations to make sure the containers stay in their place when experiencing disturbances in the track or to compensate for forces in corners. This could for instance be done with a gyrostabilizer.

- For autonomous driving: to reach a desired speed v_{ref}

- For autonomous loading unloading: to reach a desired position x_{ref}
- For autonomous stabilisator of the containers during transport: to minimize angle accelerations at the wagon

1.4 State Space Model

A state space model of the double stacked container train is shown in figure 3. At the top, the desired speed, x-position and y-position $(x_{ref}, y_{ref}, v_{ref})$ are given into the control panel. The control panel than controls the motor power as action $(u(k) = P(k))$. Due to a higher motor power, the train will move forwards.

This system has 3 types of measurements. The first one is the measurement of the engine power. This is measured by a sensor at the motor (P_{sensor}). The speed of the train is measured by measuring the rotations of the wheels ($V = \omega \cdot R$, with R = radius of the wheel). The position of the train is also measured by a GPS which gives the x_{pos} and y_{pos} of the train. All these measurements are given back to the controller.

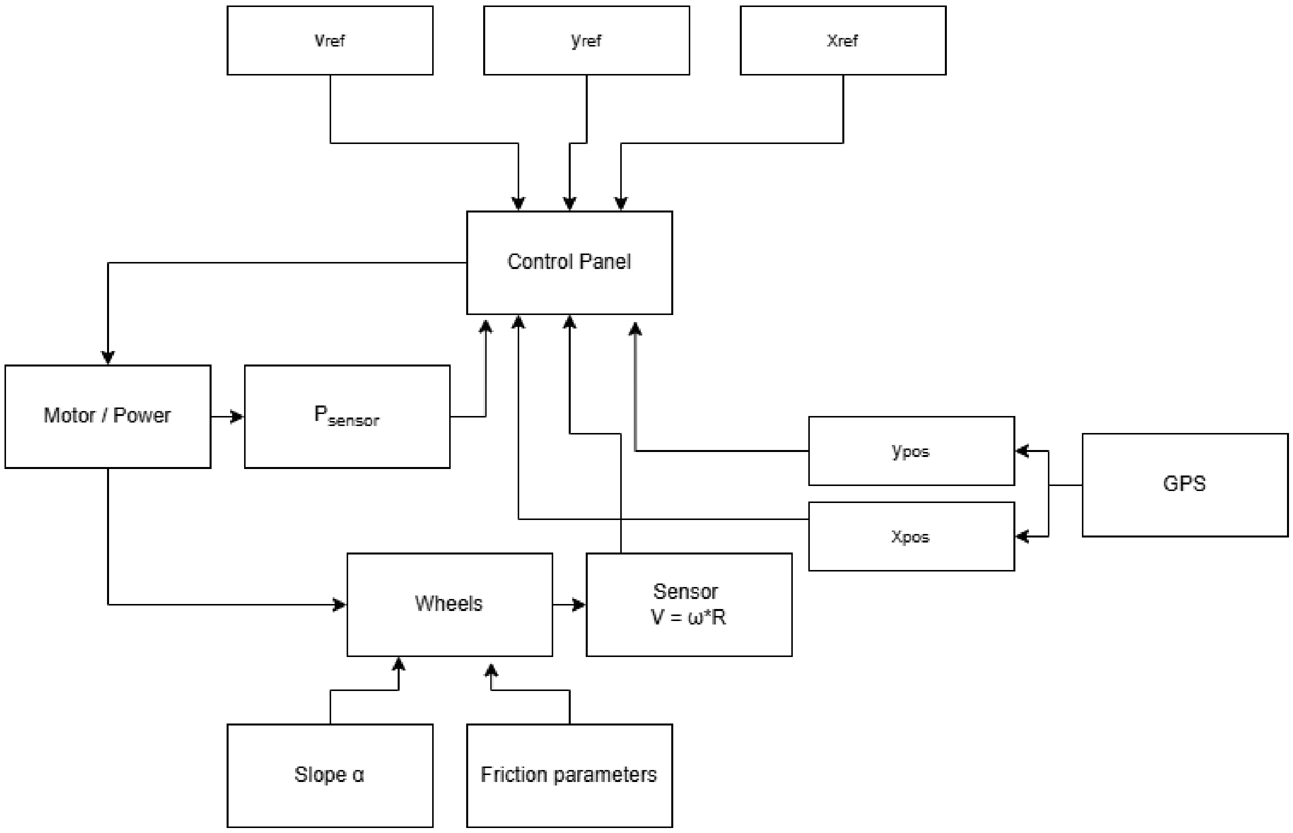

Figure 3: The state space model of the double stacked container train

1.5 Performance Functions

The system's performance is represented as a quadratic cost function according to standard MPC formulations (Qin & Badgwell, 2003):

$$J = \sum_{j=1}^{\infty} \|x_{k+j}\|_Q^2 + \|u_{k+j}\|_R^2 \tag{1}$$

where $\|x\|_Q^2 = x^T Q x$ and $\|u\|_R^2 = u^T R u$. This structure evaluates both the magnitude of the state and the control effort (Lee & Morari, 1999).

Since tracking the reference x_{ref} is our objective, the cost function is reformulated to minimize deviations from this reference:

$$J = \sum_{k=0}^{N} (x(k) - x_{\text{ref}})^T Q (x(k) - x_{\text{ref}}) + u(k)^T R u(k) \tag{2}$$

The degree to which the controller prioritizes minimizing the position error is determined by the matrix Q. A greater value in Q causes the controller to concentrate more on precise placement. The trade-off is set by the matrix R, which weights the motor's effort. At the expense of possibly allowing greater position errors, a higher R produces more fluid movement with less energy or actuator usage.

The matrix formulation can be simplified to scalar weights in a single-state, single-input system:

$$J = \sum_{k=0}^{N} q(x(k) - x_{\text{ref}})^2 + r u(k)^2 \tag{3}$$

With only position and motor force as controlled variables, this scalar formulation is appropriate for a one-dimensional system.

2 Part 2

This chapter explores how the double stack container train can be controlled intelligently as a single agent. First, the foundational components of intelligent control are introduced, including feedback structures, state-space modelling, and performance evaluation. Then, a future model-based controller is proposed, supported by a mathematical model of the train's kinematics. This part also includes a body-brain diagram that links the physical and computational subsystems, and outlines the steps of the controller algorithm.

2.1 Single-Agent Intelligent Control Mechanisms

A single-agent intelligent control mechanism is a control structure where one machine, such as a vehicle or robot, uses sensor data and feedback to adjust its actions autonomously. The goal is to reach specific objectives while dealing with disturbances, constraints, or changes in the environment. In contrast to open-loop or rule-based systems, intelligent controllers adjust their actions continuously based on feedback and updated predictions.

2.1.1 Closed-Loop Control Structure

Intelligent control is built on a closed-loop system:

- The agent collects measurements through sensors.
- A controller calculates control inputs based on those measurements and a system model.
- The system's response is measured and used as feedback to adjust future decisions.

2.1.2 State-Space Modelling

A common way to describe the system mathematically is with a discrete-time state-space model:

$$\begin{aligned} x(k+1) &= f(x(k), u(k), d(k)) \\ y(k) &= g(x(k)) \end{aligned}$$

where:

- $x(k)$ is the internal state (e.g., position or velocity),
- $u(k)$ is the input applied by the controller (e.g., force),
- $d(k)$ is a disturbance (e.g., wind or friction),
- $y(k)$ is the measured output.

2.1.3 Control Objectives and Performance

The controller tries to achieve certain goals, such as:

- Reaching or maintaining a target position or speed,
- Using as little power as possible,
- Compensating for external disturbances.

These objectives are usually expressed in a cost function, such as:

$$J = \sum_{k=0}^{N} q(x(k) - x_{\text{ref}})^2 + ru(k)^2$$

The parameters q and r determine the trade-off beteween the importance of accuracy and control effort.

2.1.4 Model Predictive Control (MPC)

An example of intelligent control is Model Predictive Control (MPC). In this method:

- The controller predicts how the system will behave over a finite horizon.
- It computes a control sequence that minimizes the cost function,
- Only the first control input is applied, and the process is repeated at each time step.

MPC is useful when the system has constraints or complex dynamics, since it can plan ahead and adjust as new information becomes available.

In short, a single-agent intelligent controller makes decisions autonomously combining feedback, prediction, and optimization, allowing the system to react effectively in dynamic conditions.

2.2 Future Intelligent Model-Based Controller

Before the double-stacked container train can be modeled by a Model Predicative Control, first the kinematics model must be defeined. As discussed in section 1.4 we will focus on the position of the train, the speed and acceleration of the locomotive, the power of the engine and the slope and direction of the track.

2.2.1 State, control and distubrance vectors

A schematic of the train on the rails is given in Figure 4.

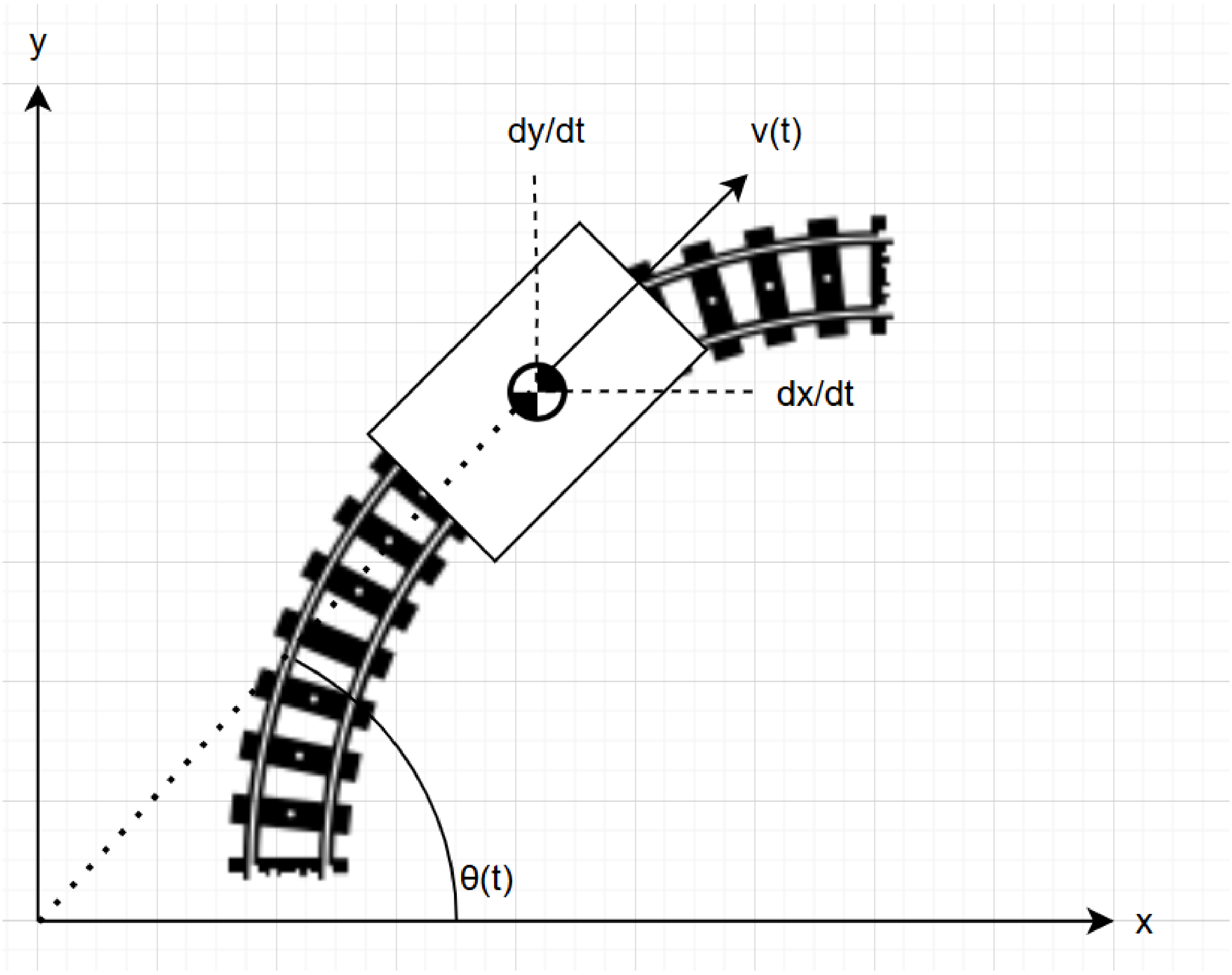

Figure 4: Top-down view of the train

We will define the state of train as a vector consisting of its location $x(t)$ and $y(t)$, and also its speed $V(t)$. For the kinematics, the state space are in continuous time t. The control input will be given as the engine power. The disturbances are made of the slope of the track α and the friction parameters. These friction parameters model the wheel, engine and air resistances in the model. It is beyond the scope of this model to come up with an accurate value for these parameters, but they will be modeled.

$$\begin{aligned} \text{State vector:} \quad & \mathbf{x}(t) = \begin{bmatrix} x(t) \\ y(t) \\ V(t) \end{bmatrix} \quad \text{(position and speed)} \\ \text{Control input:} \quad & \mathbf{u}(t) = P_{\text{engine}}(t) \quad \text{(engine power)} \\ \text{Disturbances:} \quad & \mathbf{d}(t) = \begin{bmatrix} \alpha(t) \\ \text{friction parameters} \end{bmatrix} \quad \text{(slope and resistance)} \end{aligned}$$

Kinematic Equations

We assume the train moves in a 2D plane, with speed $V(t)$ and direction $\theta(t)$. The position updates are given by:

$$\dot{x}(t) = V(t)\cos(\theta(t)) \tag{4}$$
$$\dot{y}(t) = V(t)\sin(\theta(t)) \tag{5}$$

These describe how the position changes based on current speed and heading.

Dynamic Equation for Speed

A free-body diagram is given in figure 5 and for the locomotive on a slope in figure 6.

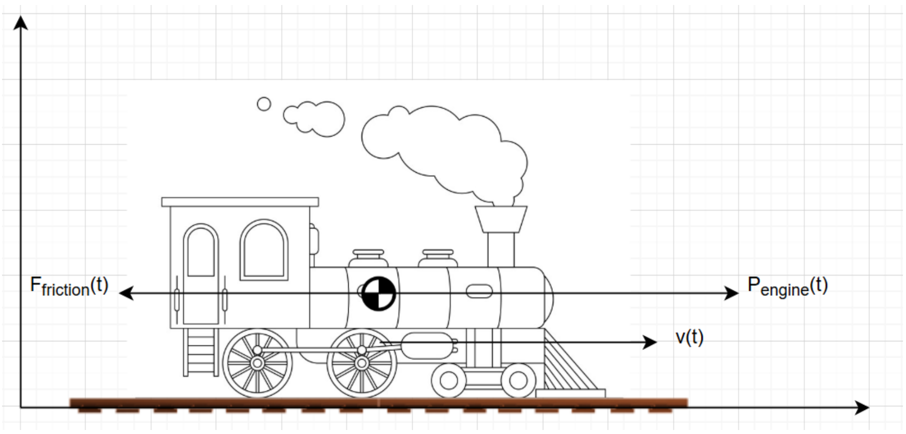

Figure 5: FBD of the locomotive

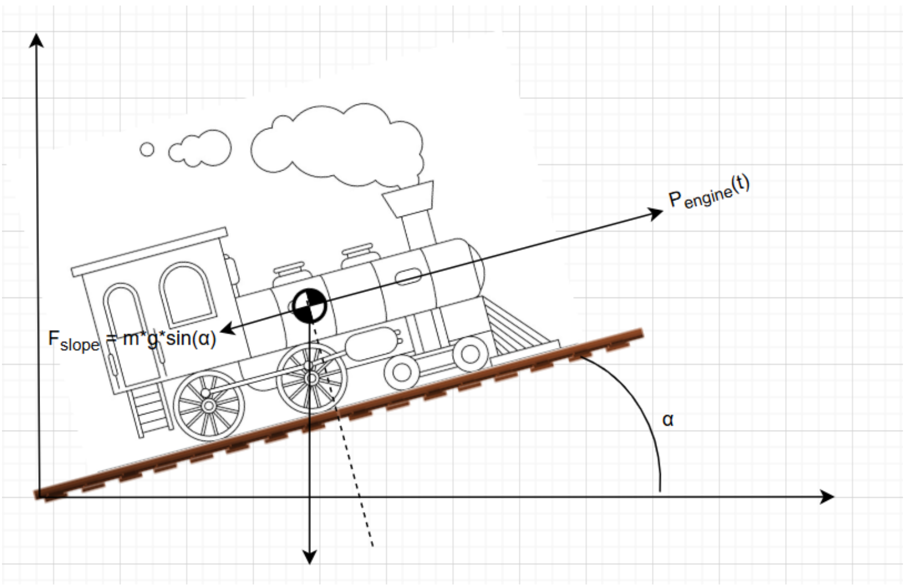

Figure 6: FBD of the locomotive on a slope

The speed changes due to net forces acting on the train. We model the train as a point mass affected by three forces:

- Engine force F_{engine}
- Friction force F_{friction}
- Gravitational force due to slope F_{slope}

The engine force is derived from power as:

$$F_{\text{engine}} = \frac{P_{\text{engine}}(t)}{V(t) + \varepsilon} \tag{6}$$

where $\varepsilon > 0$ is a small constant to prevent division by zero at low speeds.

Friction is modeled as quadratic in speed:

$$F_{\text{friction}} = c_f V(t)^2 \tag{7}$$

Slope force is due to gravity pulling against the direction of motion:

$$F_{\text{slope}} = mg\sin(\alpha(t)) \tag{8}$$

The net force determines the acceleration:

$$\begin{aligned}\dot{V}(t) &= \frac{1}{m}\left(F_{\text{engine}} - F_{\text{friction}} - F_{\text{slope}}\right) \\ &= \frac{1}{m}\left(\frac{P_{\text{engine}}(t)}{V(t) + \varepsilon} - c_f V(t)^2 - mg\sin(\alpha(t))\right)\end{aligned} \tag{9}$$

Direction of Motion

The angle $\theta(t)$ can be computed from the reference trajectory (if known) as:

$$\theta(t) = \arctan 2\left(\dot{y}_{\text{ref}}(t), \dot{x}_{\text{ref}}(t)\right) \tag{10}$$

This defines the desired travel direction along the path.

Full System Equation

Combining the above, we write the system as a nonlinear function of state, input, and disturbances:

$$\dot{\mathbf{x}}(t) = f\left(\mathbf{x}(t), \mathbf{u}(t), \mathbf{d}(t)\right) \tag{11}$$

Discretized Model (Euler Approximation)

For MPC implementation, the model is discretized in time using the Euler method:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t \cdot f\left(\mathbf{x}_k, \mathbf{u}_k, \mathbf{d}_k\right) \tag{12}$$

This discrete-time model can now be used for MPC prediction and optimization over a planning horizon.

2.3 Body-Brain Diagram

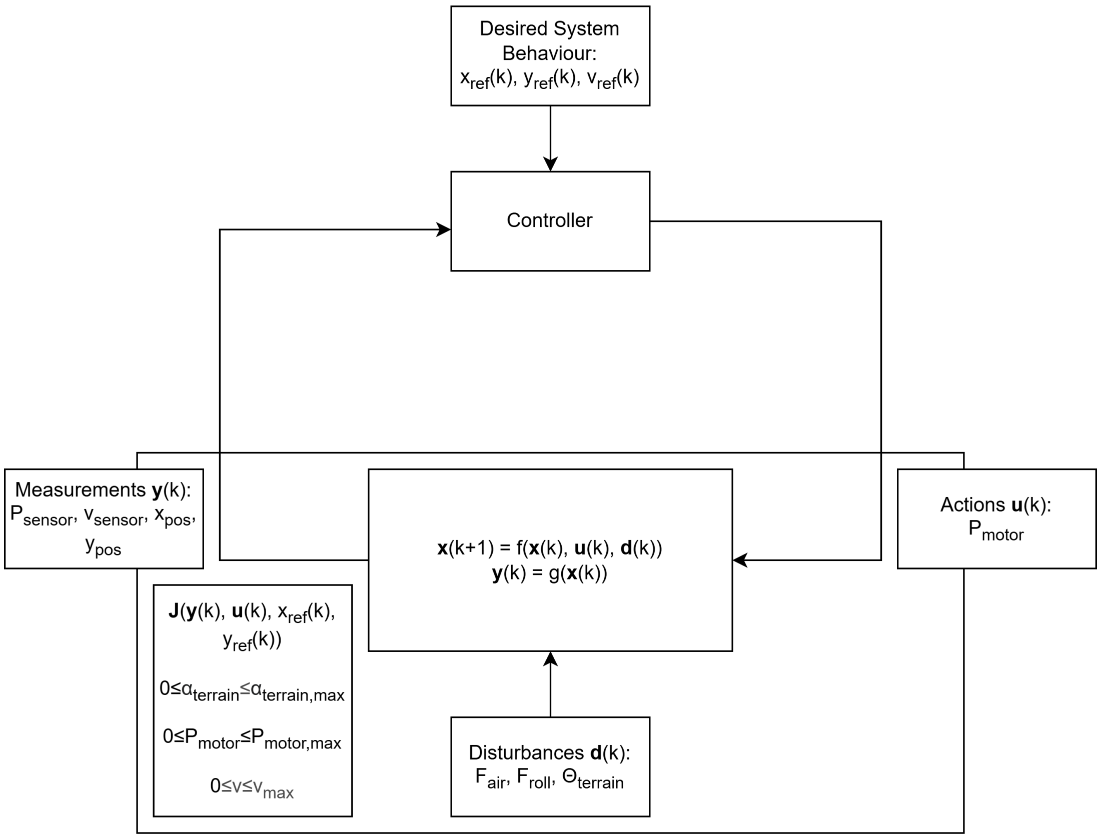
Desired System Behaviour: $x_{ref}(k)$, $y_{ref}(k)$, $v_{ref}(k)$
Controller
Measurements **y**(k): P_{sensor}, v_{sensor}, x_{pos}, y_{pos}
x(k+1) = f(**x**(k), **u**(k), **d**(k))
y(k) = g(**x**(k))
Actions **u**(k): P_{motor}
J(**y**(k), **u**(k), $x_{ref}(k)$, $y_{ref}(k)$)
$0 \leq \alpha_{terrain} \leq \alpha_{terrain,max}$
$0 \leq P_{motor} \leq P_{motor,max}$
$0 \leq v \leq v_{max}$
Disturbances **d**(k): F_{air}, F_{roll}, $\Theta_{terrain}$

Figure 7: Body-brain diagram of single agent controller

2.4 Computational Steps of Controller

For a MPC controller, the computational steps are as follows (Reppa, 2024):

At time step k=0:

1. Receive the measurements y(0)
2. Determine the initial state x(0) from the measurements
3. Determine the disturbances d(0), . . . , d(2) and the reference signal $x_{\mathrm{ref}}(1), \ldots, x_{\mathrm{ref}}(3)$
4. Solve the optimal control problem to get the optimal actions u(0), . . . , u(2)
5. Return the optimal actions u(k) as u(k) (for time step k) to the system

At time step k>0:

1. Receive new measurements $y(k)$
2. Estimate the current state $x(k)$ from the measurements
3. Determine or update:
   - Future disturbances $d(k), d(k+1), \ldots, d(k+N-1)$
   - Reference trajectory $x_{\mathrm{ref}}(k+1), x_{\mathrm{ref}}(k+2), \ldots, x_{\mathrm{ref}}(k+N)$
4. Solve the optimal control problem over the prediction horizon N to obtain the optimal control sequence $u(k), u(k+1), \ldots, u(k+N-1)$

5. Apply only the first control action $u(k)$ to the system

Using the kinematics we definied earlier, the operational steps are as follows. At first, we have the state vector defineid:

$$\bar{\mathbf{x}_\mathbf{k}} = \begin{bmatrix} x_k \\ y_k \\ V_k \end{bmatrix} \tag{13}$$

Then the disturbances for the track slope will be predicted over a time horizon N:

$$\alpha_k, \alpha_{k+1}, ..., \alpha_{k+N-1} \tag{14}$$

The reference will also be obtained:

$$\mathbf{x}_{\mathrm{ref},k+1}, \dots, \mathbf{x}_{\mathrm{ref},k+N}$$

where each $\mathbf{x}_{\mathrm{ref},k+i} = [x_{\mathrm{ref}}, y_{\mathrm{ref}}, V_{\mathrm{ref}}]^T$

Use the discretized dynamic model:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t \cdot f(\mathbf{x}_k, \mathbf{u}_k, \mathbf{d}_k) \tag{15}$$

to predict future states over the horizon.

The function $f(\mathbf{x}_k, \mathbf{u}_k, \mathbf{d}_k)$ consists of:

$$\dot{x}(t) = V(t)\cos(\theta(t)) \tag{16}$$
$$\dot{y}(t) = V(t)\sin(\theta(t)) \tag{17}$$

And

$$\begin{aligned} \dot{V}(t) &= \frac{1}{m}\left(F_{\mathrm{engine}} - F_{\mathrm{friction}} - F_{\mathrm{slope}}\right) \\ &= \frac{1}{m}\left(\frac{P_{\mathrm{engine}}(t)}{V(t)+\varepsilon} - c_f V(t)^2 - mg\sin(\alpha(t))\right) \end{aligned} \tag{18}$$

Where

$$\theta(t) = \arctan 2\left(\dot{y}_{\mathrm{ref}}(t), \dot{x}_{\mathrm{ref}}(t)\right) \tag{19}$$

Then, the optimization is

$$\min_{\mathbf{u}_k,\dots,\mathbf{u}_{k+N-1}} \sum_{i=1}^{N} \|\mathbf{x}_{k+i} - \mathbf{x}_{\mathrm{ref},k+i}\|^2 + \|\mathbf{u}_{k+i-1}\|^2$$

The constraints on the sytem are as follows:

- Power: $P_{min} \leq P(k) \leq P_{max}$
- Speed: $V_{min} \leq V(k) \leq V_{max}$
- Slope: $\alpha_{min} \leq \alpha(k) \leq \alpha_{max}$

3 Part 3

This chapter extends the single-agent control approach by introducing distributed coordination between multiple intelligent systems. The double stack container train, crane, and AGV are treated as separate agents, each with their own controller and local decision-making. Information exchange with a supervisory agent is used to synchronize tasks, resolve conflicts, and ensure efficient operation across the terminal. A proposed architecture and diagram illustrate how references and status updates are communicated between agents.

3.1 Multi-Agent and Multi-Level Control Architectures

A multi-agent system has, in contrast to a single agent system multiple controllers, as shown in figure 8.

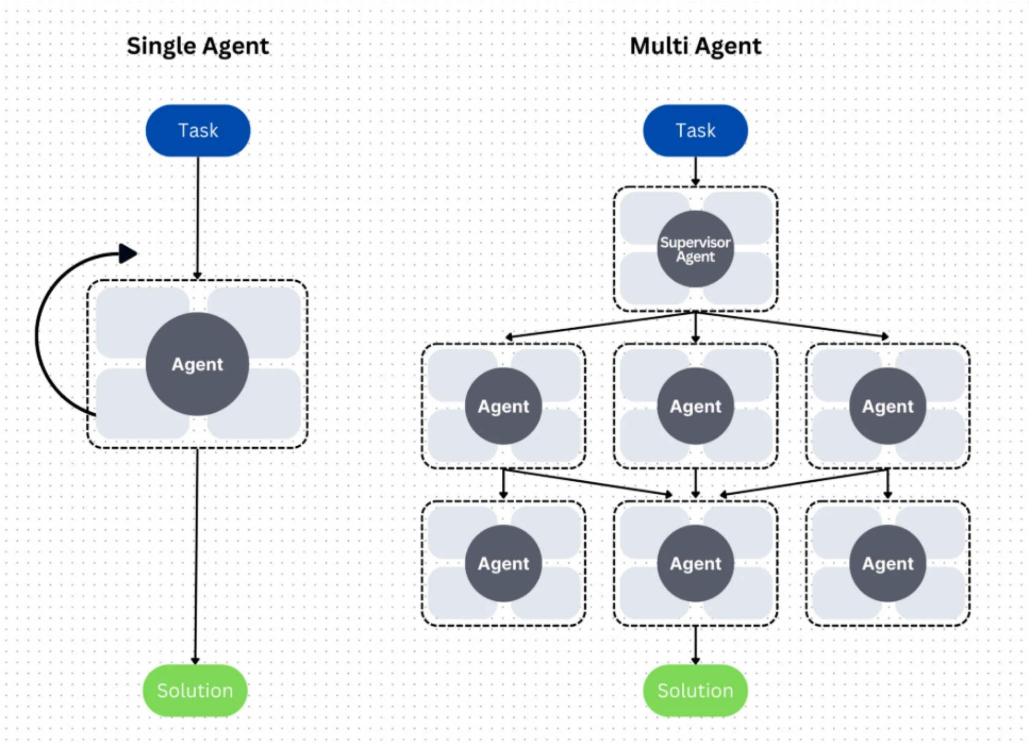

Figure 8: Distributed Control Architecture (Maierhöfer, 2025)

The control architecture of a multi-agent can be decentralized or distributed. A decentralized controller means that it does not consider cooperation and communication among local agents. A distributed controller on the other hand means that it does consider communication and cooperation among local agents (Negenborn and Hellendoorn, 2010). A decentralized system has a lower computing time, but also a lower optimality. A distributed system has in contrast better optimality, but longer computing time as stated in the lecture slides of ME44300.

A multi-agent system can have a single-layer structure where the controllers are on the same hierarchical level, or a multi-layer structure. A single-layer multi-agent system places all agents at the same decision-making level, where they may either act independently (decentralized) or cooperate (distributed), but without overarching control or guidance from higher layers. Such flat architectures are generally easier to implement and maintain but may not scale well with system size or complexity (Qin and Badgwell, 2003). In a multi-layer structure, the controllers are on a different hierarchical level. This structure allows for modularization, abstraction, and scalability, making it particularly suitable for large-scale systems such as industrial plants, transportation networks, or energy systems (Morari and Lee, 1999).

3.2 Proposed Distributed Control Architecture

The proposed distributed control architecture in Figure 9 treats the double stack container train, the crane, and the AGV as separate autonomous systems, each equipped with its own controller. These controllers make local decisions based on real-time sensor measurements and task-specific references. Instead of relying on centralized control, the agents coordinate through a shared communication structure. A supervisory controller oversees this coordination by assigning references, resolving conflicts, and sharing key status information (e.g., availability of infrastructure or time slots).

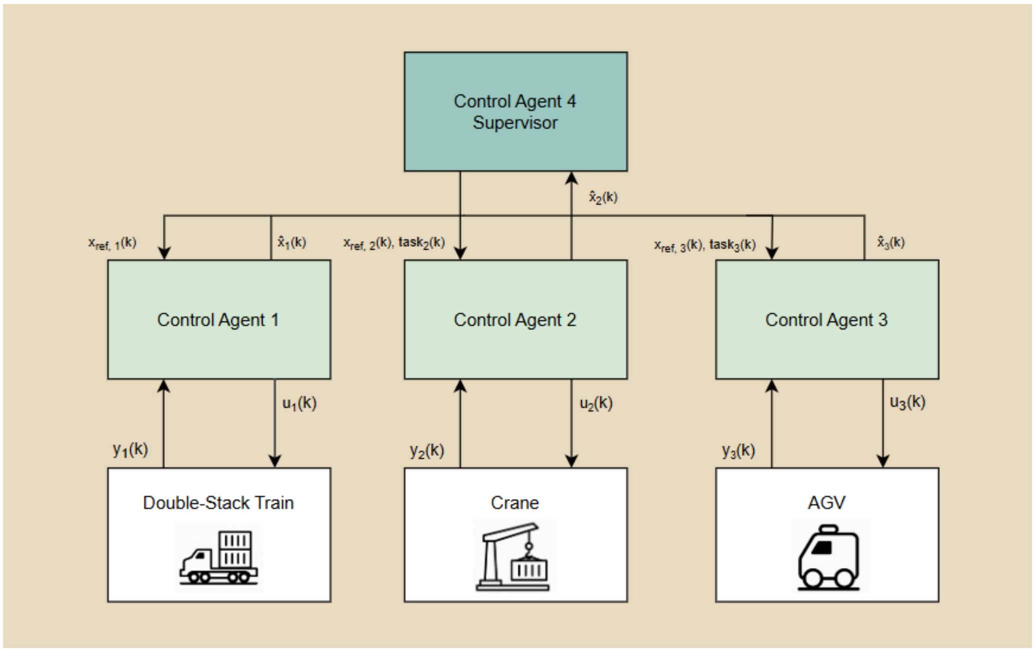

Figure 9: Distributed Control Architecture

3.2.1 Interconnected systems

The main interconnected systems are:

- The **double stack container train**, which moves along a fixed track to loading and unloading zones.
- The **crane**, which moves containers between the train and the AGV.
- The **AGV**, which transfers containers between the crane and other areas in the terminal.

These systems interact both physically (through container transfers) and operationally (through scheduling and shared resource planning).

3.2.2 Flow of information between each controller and its controlled system

Each control agent operates its own feedback loop. As shown in the diagram, each physical system sends measurement data $y_i(k)$ to its respective controller, which responds with a control input $u_i(k)$ to influence the system's behavior.

- The double stack container train controller (Control Agent 1) receives $y_1(k) = \begin{bmatrix} x(k) \\ y(k) \\ v(k) \end{bmatrix}$, representing the measured position and speed of the train. It computes $u_1(k)$, a control input representing tractive force or braking command, to reach a desired stop position and arrival time.

- The crane controller (Control Agent 2) receives $y_2(k) = \begin{bmatrix} x(k) \\ y(k) \\ z(k) \\ \text{status} \end{bmatrix}$, which includes the current 3D position of the gripper and its operational status (e.g., idle, busy, or container attached). Based on this, the controller computes a control input $u_2(k) = \begin{bmatrix} v_x(k) \\ v_y(k) \\ v_z(k) \end{bmatrix}$, representing the desired gripper velocity in x, y, and z directions. This allows the crane to move containers precisely between the train and the AGV.
- The AGV controller (Control Agent 3) receives $y_3(k) = \begin{bmatrix} x(k) \\ y(k) \\ \text{status} \end{bmatrix}$, including the AGV's 2D position and container load status. It computes $u_3(k) = \begin{bmatrix} v_x(k) \\ v_y(k) \end{bmatrix}$, a velocity vector used by the AGV's drive system to reach its assigned delivery location.

Each of these input-output loops is represented in the diagram as vertical arrows between each controller and its physical system: $y_i(k)$ flows upward from the system to the controller, and $u_i(k)$ flows downward from the controller to the system.

3.2.3 Information flow between controllers and local computations

Coordination across agents is achieved through information exchange with a supervisory agent, shown at the top of the diagram. Controllers receive references from the supervisor, such as desired positions and delivery tasks, and send back planning-relevant status updates. This information includes:

- **Train (Agent 1):** receives
$$x_{\text{ref},1}(k) = \begin{bmatrix} x_{\text{target},1} \\ y_{\text{target},1} \\ t_{\text{ref},1}(k) \end{bmatrix}$$
from the supervisor. It reports
$$\hat{x}_1(k) = t_{\text{eta},1}(k)$$
and uses this in a model predictive control (MPC) routine to compute its braking and motion profile.
- **Crane (Agent 2):** receives
$$x_{\text{ref},2}(k) = \begin{bmatrix} x_{\text{target},2} \\ y_{\text{target},2} \\ z_{\text{target},2} \end{bmatrix}$$
and task
$$\text{task}_2(k) = \begin{bmatrix} \text{container\_ID} \\ \text{source\_agent (e.g. train or AGV} \\ \text{target\_agent (e.g. train or AGV)} \end{bmatrix}$$
from the supervisor. It reports
$$\hat{x}_2(k) = \begin{bmatrix} x_c(k) \\ y_c(k) \\ z_c(k) \\ \text{task\_status}_2(k) \end{bmatrix}$$

- **AGV (Agent 3):** receives

$$x_{\text{ref},3}(k) = \begin{bmatrix} x_{\text{target}} \\ y_{\text{target}} \\ t_{\text{ref},3}(k) \end{bmatrix}, \quad \text{task}_3(k) = \text{container\_ID}$$

  and reports

$$\hat{x}_3(k) = \begin{bmatrix} t_{\text{eta},3}(k) \\ \text{task\_status}_3(k) \end{bmatrix}$$

Overall, this architecture enables autonomous yet coordinated behavior across agents, improves flexibility, and allows individual components to be upgraded or replaced independently, without requiring changes to the overall control structure.

3.3 Resolving Conflicts

Proposed Conflict Resolution Approach: Distributed Coordination with Shared Variable Synchronization

Core Components: Shared Variables:

1. TrainPositionStatus, Is the train aligned and stationary for loading?
2. CraneAvailability, Is the crane ready to lift?
3. AGVQueueStatus, Is the next container available for pickup?

Controllers:

1. Train Controller, Signals readiness and position.
2. Crane Controller, Coordinates lift schedule and checks if train and AGV are ready.
3. AGV Controller, Reports container delivery status and queues tasks.

Protocol:

1. Each controller updates shared variables and listens to others.
2. The supervising control agent is used to manage access to the shared loading resource (crane).
3. Priority logic determines which action proceeds next (e.g., crane waits if no container, AGV waits if crane is busy).

Example approach for a conflict where the train wants to move and the crane wants to load a container. In this case the the train has signaled it needs to move before being able to accept more containers. The crane has a container ready to be loaded onto the the train. In this case the supervising control agent sees that the train does not have any place for containers before it moves up to its new position to make new space available for the crane to load. The supervising agent tells the crane and potentially AGV's to wait until the train has moved to its new position before starting to load again. After the train has stopped at its new position the supervising agent tells the crane to resume the loading and the AGV's to continue to supply the crane with containers.

3.4 Potential Challenges

The implementation of this new technology presents several challenges that must be overcome. Some of these are common to many emerging technologies, such as investment costs and public perception. Others are more specific to the rail sector—for example, the reliability of sensors in adverse weather and lighting conditions, and the wide variety of track and signaling systems. Additional challenges will be elaborated below.

Currently, regulations for autonomous vehicles are not yet fully established, which creates legal obstacles for autonomous trains. Liability, for instance, remains a significant question: who is responsible for damages if a train derails or crashes? Will it be the company operating the train, or the one that developed the autonomous driving system? These uncertainties also make insurance companies reluctant to issue policies for cargo transported by autonomous trains.

Another challenge lies in the reliance on GPS and communication systems (especially in distributed control setups) when trains pass through tunnels or mountainous regions. In such environments, these systems may perform poorly or fail entirely. This could lead to dangerous situations where the train makes incorrect decisions due to incomplete or inaccurate information. While positioning errors could be mitigated using technologies like Kalman filters, maintaining communication with a central control system is more difficult to resolve.

Finally, removing the driver from the train significantly increases the system's complexity. As a result, maintenance becomes more challenging, requiring more advanced training for maintenance personnel. Additionally, remote operators monitoring the train must be highly skilled: they need to understand control interfaces, interpret sensor data, perform diagnostics, and make timely assessments of system decisions. Therefore, although fewer people may be needed to operate autonomous trains, those who are involved must be more highly educated and technically proficient.

List of Figures

References

Joshua, A. O. (2015). The need for multimodal transport development in nigeria. *Journal of Geography and Regional Planning*, *8*(9), 239–243. https://doi.org/10.5897/jgrp2015.0508

Lee, J. H., & Morari, M. (1999). Model predictive control: Past, present and future. *Computers Chemical Engineering*, *23*(4-5), 667–682.

Maierhöfer, J. (2025, March). AI Agent Observability with Langfuse. https://langfuse.com/blog/2024-07-ai-agent-observability-with-langfuse

Morari, M., & Lee, J. H. (1999). Model predictive control: past, present and future. *Computers Chemical Engineering*, *23*(4-5), 667–682. https://doi.org/10.1016/s0098-1354(98)00301-9

Negenborn, R. R., & Hellendoorn, H. (2010). Intelligence in transportation infrastructures via model-based predictive control. In R. Negenborn, Z. Lukszo, & H. Hellendoorn (Eds.), *Intelligent infrastructures* (pp. 3–24). Springer.

PRC Rail Consulting Ltd. (2019). Train equipment | The Railway Technical website |. http://www.railway-technical.com/trains/rolling-stock-index-l/train-equipment/#:~:text=A%20locomotive%20or%20multiple%20unit,to%20use%20the%20high%20voltage

Qin, S., & Badgwell, T. A. (2003). A survey of industrial model predictive control technology. *Control Engineering Practice*, *11*(7), 733–764. https://doi.org/10.1016/s0967-0661(02)00186-7

Reppa, V. (2024). Me44300: Multi-machine coordination for logistics – lecture 2 [PowerPoint slides].

Ruf, M., Cordeau, J.-F., & Frejinger, E. (2022). The load planning and sequencing problem for double-stack trains. *Journal of Rail Transport Planning & Management*, *23*, 100337. https://doi.org/10.1016/j.jrtpm.2022.100337

Stephens, B. (2024, October). CSX launches double-stack service at the Port of Baltimore. https://www.trains.com/trn/news-reviews/news-wire/csx-launches-double-stack-service-at-the-port-of-baltimore/

Upadhyay, A., Gu, W., & Bolia, N. (2017). Optimal loading of double-stack container trains. *Transportation Research Part E: Logistics and Transportation Review*, *107*, 1–22. https://doi.org/10.1016/j.tre.2017.08.010