Using Imitation Learning to Implement Control Orchestration for Smart Chassis

Sarah Imene Khelil[1], Bruno Monsuez[1], Maud Geoffriault[2] and Anh Lam Do[2]
[1]*Department of Computer and System Engineering (U2IS), ENSTA Paris, Institut Polytechnique de Paris, Palaiseau, France*
[2]*Chassis Systems Department, Groupe Renault, Guyancourt, France*

Keywords: Behavioral Cloning, Control Allocation, Imitation Learning, Deep Reinforcement Learning, Supervised Learning.

Abstract: Despite the advances in control allocation for over-actuated systems, the need for a comprehensive, optimized, and safe solution remains ongoing. Traditional methods, though mature, struggle with the complexities of coupled non-linear allocation and the need for extensive computational resources. Machine learning may provide significant advantages through its generalization and adaptation capabilities, especially in scenarios where linear approximations are employed to reduce computational burdens or when the effectiveness of actuators is uncertain. Recent advances in imitation learning, particularly behavioral cloning, and deep reinforcement learning have demonstrated promising results in addressing these challenges. This paper aims to determine the potential of using machine learning in control orchestration for smart chassis to go beyond allocation issues to include interaction management across systems, resource balance, and safety and performance limits. We present a set of techniques that we believe are relevant to experiment to address potential challenges like prediction and complexity for control allocation in smart chassis systems, which will be tested in the upcoming articles.

1 INTRODUCTION

Chassis system design prioritizes safety, stability, and performance. Initially, these systems aimed to enhance individual vehicle parameters by utilizing longitudinal tire forces for longitudinal control and lateral tire forces for lateral control. Over time, more sophisticated methods emerged, such as differential steering between the right and left tires for longitudinal control and differential acceleration between the front and rear tires for lateral control, electronic stability programs and torque vectoring are examples of such techniques. The integration of multiple subsystems that are critical for reinforcing safety, and that drive the same physical quantity in over-actuated vehicles raises the problem of how to orchestrate those subsystems without degrading neither the dynamic safety nor the vehicle performance.

Traditionally, rule-based strategies based on expert knowledge of vehicle dynamics are employed to address these issues, under what we call the downstream approach. Unlike traditional approaches, optimization-based control allocation methods ensure stability while leveraging the unique strengths of each system, promising better performance and reliability across diverse driving conditions (Kissai, 2019). We call this the upstream approach.

Integrated Chassis Control (ICC) (Skrickij, Kojis, Šabanovič, Shyrokau, & Ivanov, 2024) depends on the upstream approach playing an important role in harmonizing these active sub-systems to optimize vehicle dynamics, safety, comfort, and energy based on current conditions and actuator limitations. Orchestration goes beyond that by managing the coordination, and planning of subsystems over longer timeframes, considering strategic goals and future states.

The two terms will be used interchangeably as the current state-of-the-art swings between the two terms.

Going back to ICC, it follows a modular strategy for orchestration as it separates the tracking aspect of control from the distribution perspective. As illustrated in Figure 1, the high-level control specifies the desired control. A control allocator distributes this effort among various actuators to produce the desired

Khelil, S., Moonsuez, B., Geoffriault, M. and Do, A.
Using Imitation Learning to Implement Control Orchestration for Smart Chassis.
DOI: 10.5220/0013049300003822

In *Proceedings of the 21st International Conference on Informatics in Control, Automation and Robotics (ICINCO 2024) - Volume 1*, pages 462-471
ISBN: 978-989-758-717-7; ISSN: 2184-2809

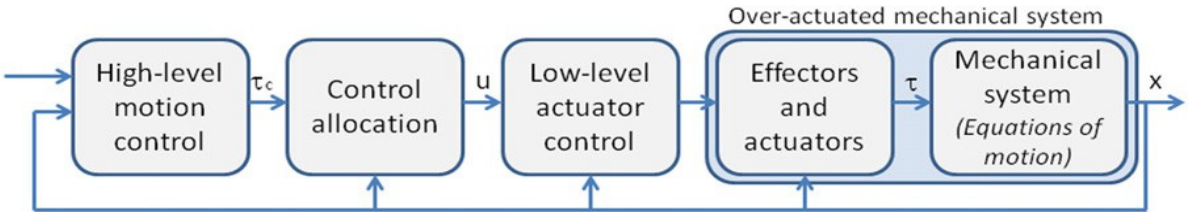

Figure 1: Control system structure including control allocation (Johansen & Fossen, 2013).

effect while optimizing for additional objectives such as minimizing energy consumption (multi-objective optimization).

Currently, researchers assume a linear relationship between actuator actions and resulting control outcomes to use computationally efficient algorithms such as daisy chaining, redistributed pseudo-inverse or cascaded generalized inverse, and methods based on linear programming and quadratic programming for control allocation (CA) (Johansen & Fossen, 2013). However, the CA problem in cars is inherently coupled and nonlinear suffering from discontinuities as the system behavior changes according to the changes in the environment and changes in the system itself. The dependencies between the vehicle states make the study even more complex.

As the objective is to have a control orchestration that predicts the saturation of the actuators by taking into account their dynamics while keeping track of the change in the environment and behaving accordingly, preview capabilities are of a big interest to such a problem.

Predictive nonlinear control allocation is challenging necessitating significant onboard computational power and accurate models of the effectiveness function to find the optimal actuator combination. Here comes machine learning (ML) offering solutions to optimization and generalization needs. Depending on the neural nets used in the generation of the cloner, this latter can be enhanced by predicting future states and behaviors based on past data if such neural nets have a memory (e.g. long short-term memories). It captures complex, nonlinear, and time-varying vehicle dynamics and accounts for uncertainty and variability in real-world driving conditions, improving reliability. Also, ML personalizes control strategies to individual driver behavior and preferences.

The use of a cloner causes big challenges as the same level of performance and the same quality of results should be ensured while addressing the big problem of the computational burden of the already existing solutions. A comparative study should be carried out during the development phase.

Recent advances in imitation learning and deep reinforcement learning show promising capabilities that may be used in CA formulation. Recent work by Khan, Mobeen, Rajput, and Riaz (2024) explored the use of a neural network (NN) in CA to account for the nonlinear CA problem formulation and compared it to quadratic programming (QP) allocation. The two approaches have similar performance with reduced run time for the proposed scheme of ANN-based control allocator. From another part, the optimization problem has been studied by Skulstad, Li, Fossen, and Zhang (2023) who proposed a deep learning technique for the problem of dynamic positioning in marine vessels. The study used behavioral cloning to capture the intricate relationships between control inputs and system responses. The solution was as performant as a sequential QP-based CA however the constraints were violated leading to another research by the same authors comprising the addition of a costume NN layer to enforce hard constraints (Raghunathan, Skulstad, Li, & Zhang, 2023). Wu and Litt (2023) explored the use of deep reinforcement learning (RL) for controlling an aircraft equipped with distributed electric propulsion (DEP). The research demonstrates that RL can efficiently manage the allocation of thrust among multiple electric engines, achieving stable flight even under conditions of actuator failure without a knowledge of the efficiency matrix of the CA.

Through this article, we aim to highlight the potential of utilizing machine learning in control allocation, especially for coupled non-linear problems.

This paper is organized as follows. Section 2 introduces the CA problem formulation for a simple chassis problem. Section 3 presents the existing learning techniques of CA. The study conducted by the state of the art is demonstrated through the methodologies and results in section 4. In section 5, a simple automotive-related example is presented. It is followed by concluding remarks in section 6.

2 CONTROL ALLOCATION THEORY

Nothing is better than introducing a problem through an example like controlling yaw rate for better steering and stability. Active Front Steering (AFS) and Active Rear Steering (ARS) have been proposed as lateral methods that can directly affect the parameter in question. However, physics laws proved the possibility of using braking-based systems to generate yaw moments, which gives rise to the problem of managing both longitudinal, through the Vehicle Dynamics Control (VDC) system, and lateral, through ARS, controls to achieve the desired behavior. Such a system is called an over-actuated system, and the CA aims here to find the optimal combination of the actuator distribution to achieve the desired behavior.

As has been presented by Kissai (2019) the upstream CA approach generates more realistic commands as it takes into account the tire dynamic couplings and constraints.

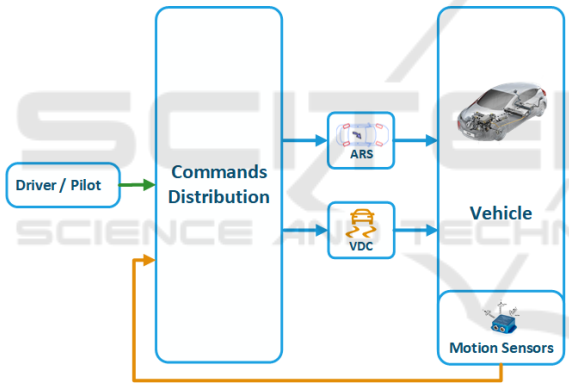

Figure 2: Structure of the upstream coordination approach (Kissai, 2019).

We want to control the yaw rate rate $\dot{\psi}$ using the forces generated by the existing actuators. To synthesize this control allocator we take the yaw moment M_z generated as a high-level command, the following equation should be considered:

$$M_z = F_{x_{f,l}}\left(l_f \sin\delta_{f_i} - \frac{d}{2}\cos\delta_f\right) + F_{x_{f,r}}\left(l_f \sin\delta_{f_i} + \frac{d}{2}\cos\delta_{f_i}\right) + \left(F_{x_{r,r}} - F_{x_{r,l}}\right)\frac{d}{2} - F_{y_r} l_r \quad (1)$$

where:

- d: vehicle's track,
- l_k : distance between the axle k and the vehicle's center of gravity, where k designates the front or the rear,
- F_{yk}: overall lateral force at the axle k,
- $F_{x_{k,j}}$: longitudinal force of each tire, where j designates left or right,
- δ_{k_i} : initial steering angle before the application of the tire force.

Illustrated in Figure 1, the control system structure prominently features control allocation. CA of over-actuated systems functions through three distinct stages:

2.1 High-Level Control

This initial stage involves generating virtual control commands τ_c, without necessitating detailed actuator information, relying on a simplified linear model.

$$\dot{x} = f(x,t) + g(x,t)\tau \quad (2)$$

where:

- f, g are functions,
- $x \in R^n$ is the state vector,
- t is time,
- $\tau \in R^m$ is the virtual input vector.

Let us take $\dot{\psi}$ as the high-level controlled state x and M_z as the virtual command, from the global vehicle model, we get:

$$\dot{\psi}(s) = \frac{M_z}{I_z s} \quad (3)$$

where:

- I_z: inertia moment,
- s: Laplace operator.

2.2 Control Allocation

Following high-level control, this stage coordinates various effectors, accounting for input constraints and fault tolerance, and maps the vector of virtual input forces and moments τ_c into individual effector forces or moments τ.

$$\tau = h(u,x,t) \quad (4)$$

where:

- h is a function,
- $u \in U \subset R^p$ is the control input, with U representing control constraints due to saturation and other physical constraints.

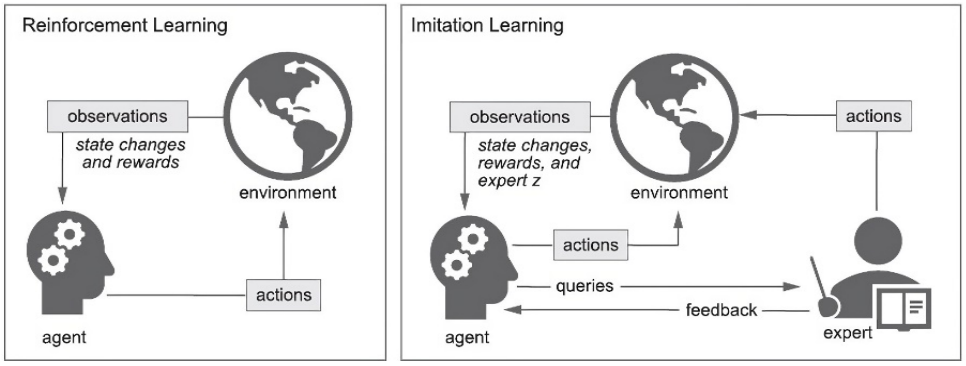

Figure 3: The reinforcement and imitation learning paradigms (Davila Delgado & Oyedele, 2022).

Commonly, effector models are linear in u:

$$\tau = h(u,x,t) = B(x,t)u \quad (5)$$

By taking the forces generated by the existing actuators as $u$and $M_z$as the virtual command, we get the following effectiveness matrix:

$$B = [l_f \sin\delta_f - \frac{d}{2}\cos\delta_f,\ l_f \sin\delta_f + \frac{d}{2}\cos\delta_f,\ -\frac{d}{2},\ \frac{d}{2},\ -l_r] \quad (6)$$

Clearly, by looking at the effectiveness matrix, there is a nonlinear relationship between the CA in-out pair as δ_f depends on the forces and vice-versa through the tire model.

The model can be complexified more by including the actuators' dynamics and delays.

2.3 Low-Level Control

This stage focuses on controlling each effector via its actuators.

Control allocation computes a control input u that ensures $\tau_c = h(u,x,t)$ at all times t.

If a feasible u cannot be found, the control allocation searches for a control input that minimizes the allocation error $\tau_c - \tau$.

$$\min_{u \in R^p} Q(\tau_c - \tau) \quad (7)$$

Incorporated within this process are actuator rate and position limits to ensure stability and performance.

3 EXISTING APPROACHES OF MACHINE LEARNING IN CONTROL ALLOCATION

As presented through the example, the CA problem in cars is inherently nonlinear and coupled. This behavior can be made more complex as the system behavior changes according to the changes in the environment (preferred actuators depending on the grip) and changes in the system itself (change in the number of the actuators) making it suffer from discontinuities. To achieve effective control orchestration that predicts actuator saturation while accounting for their dynamics and adapting to environmental changes, preview capabilities are essential. This task demands substantial onboard computational power and precise models of actuator effectiveness to find the optimal combination. Machine learning (ML) provides powerful solutions to these optimization and generalization challenges.

By predicting future states and behaviors from past data, ML can handle complex, nonlinear, and time-varying vehicle dynamics. It improves reliability by addressing uncertainty and variability in real-world driving conditions. Furthermore, ML can tailor control strategies to match individual driver behaviors and preferences.

Depending on the expert used for the learning, we can distinguish two existing approaches of learning in control allocation, as illustrated in Figure 3.

3.1 Learning of a Non-Existing Control Allocator Using Deep Reinforcement Learning

Deep reinforcement learning (DRL) combines reinforcement learning (RL) and deep learning (DL) to enable learning expert behavior.

3.1.1 Reinforcement Learning

It is a mapping between states and actions with the goal of maximizing a reward function.

In short the learning is achieved through environment exploitation.

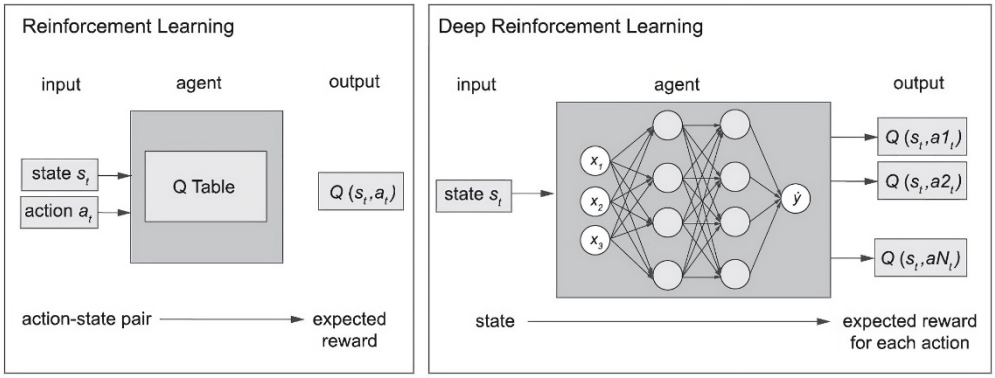

Figure 4: Main differences between RL and DRL (Davila Delgado & Oyedele, 2022).

It is constructed from (Manuel Davila Delgado & Oyedele, 2022):

Agents, Environments, Actions, and States: An agent observes the state $\boldsymbol{s_t}$ of the environment then takes actions $\boldsymbol{a_t}$ and receives feedback on the outcome of the choices made.

Rewards: A scalar that indicates the level of success to reach the goal every time the agent takes an action.

Value Function: The $\boldsymbol{Q}$ value of a state is the total reward that an agent can expect in the future starting from that given state. Rewards determine immediate success, while value $\boldsymbol{Q}$ indicates long-term accumulated success.

Policy: The policy $\boldsymbol{\pi}$ is a state-action mapping that maximizes the reward:

$$\pi^*(a) = \max_a Q(s,a) \quad (8)$$

3.1.2 Deep Learning

It defines the use of a neural network with a large number of hidden layers.

In DRL, deep neural networks are used to discover state-action mappings instead of a relational table known as a Q-table to map states to actions (Manuel Davila Delgado & Oyedele, 2022).

It can be categorized into model-based and model-free approaches:

Model-Based Methods: These methods involve learning a model of the environment and using it to plan actions.

Model-Free Methods: These methods learn policies directly from interactions with the environment without explicitly modeling them.

DRL presents one of the most needed features as it can generalize from specific training examples to unseen situations, given sufficient data diversity during training.

However, as opposed to imitation learning, as there is no expert the DRL algorithms typically require a large number of interactions with the environment to learn effective policies, leading to high computational costs. Designing appropriate reward functions can be complex and task-specific, and the exploration-exploitation trade-off presents a big limitation often leading to suboptimal exploration strategies (Zare, Kebria, Khosravi, & Nahavandi, 2023).

3.2 Learning of an Existing Control Allocator Through Supervised Learning

We call the use of supervised learning, Table 1, to clone the behavior of an expert imitation learning (IL).

IL aims at mimicking an agent that is considered to perform well in a particular task. This is essentially learning to map observations to actions.

Imitation learning is useful in fields like autonomous vehicles, robotics, and other industries where large sensory data of expert demonstrations are available (Dey, Marzullo, Zhang, & Henze, 2023).

Such an approach is used whenever a well-developed expert, in our case a CA, exists and needs to be replaced because of its optimization burden, and complex relation formulation between the in-out pairs (coupling between in-out pairs, non-linearity for example).

Table 1: Types of machine learning approaches (Manuel Davila Delgado & Oyedele, 2022).

Type	Supervised Learning	Unsupervised Learning	Reinforcement Learning
Data	Labelled data: (x, y) x is data, y is the label.	Data: (x) x is unlabelled data.	Data: (s_t, a_t) s_t is state, a_t is action.
Goal	Learn function to map: $x \rightarrow y$.	Learn an underlying structure to find relationships.	Maximize future reward over many time steps through interaction with the environment or with an expert.

IL can be classified into:

3.2.1 Behavioral Cloning (BC)

It is a mapping of states $\boldsymbol{s_t}$ to actions $\boldsymbol{a_t}$ as shown bellow:

$$a_t = \pi(s_t) \quad (9)$$

The policy π can be learned by a supervised learning method from a dataset of pairs $D = \{s_t, a_t\}$.

The neural network mimics the expert behavior by learning state-action pairs.

3.2.2 Inverse Reinforcement Learning (IRL)

In inverse reinforcement learning the reward function is unknown and this needs to be recovered from the existing expert demonstration.

The neural network infers the underlying reward function (positive reward when the error between the setpoint and the output of the system gets smaller) from the expert demonstrations and then uses RL to learn the optimal policy.

In these approaches the objective is to learn the reward function –instead of the policy– directly from an expert's trajectory, and then find the optimal policy using an RL approach (Manuel Davila Delgado & Oyedele, 2022).

The Table 2 summarizes the main advantages and disadvantages of the two principal IL approaches.

3.2.3 Adversarial Imitation Learning (AIL)

This approach involves training a policy (generator) and a discriminator in an adversarial setup, where the discriminator learns to distinguish between the agent's actions and the expert's actions, while the policy learns to fool the discriminator by producing actions that are indistinguishable from those of the expert, as illustrated in Figure 5:

Table 2: Benefits and disadvantages of imitation learning approaches (Manuel Davila Delgado & Oyedele, 2022).

Type	Benefits	Disadvantages
Behavioral Cloning	-Simple implementation. -Effective for small state spaces.	Requires almost total coverage of the state space.
Inverse Reinforcement Learning	-No need to specify reward function. -Improved generalization. -Robust against changes in the environment.	Very high computational costs for relatively small state and action spaces.

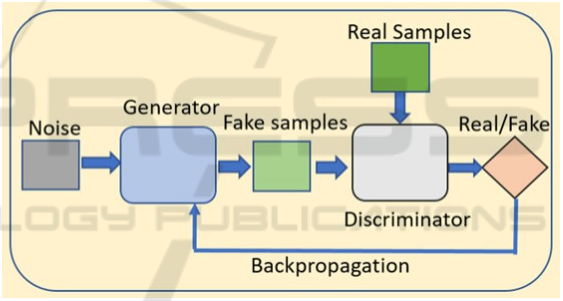

Figure 5: Graphical illustration of the working principle of adversarial imitation learning (Latif et al., 2021).

Illustrated in Figure 6, are the imitation learning techniques.

In short, using adversarial training, the generator (policy network) improves its ability to produce expert-like actions, thereby learning the mapping between states and actions effectively.

Clearly, imitation learning requires fewer interactions with the environment as it has pre-collected expert data. In IRL, the reward functions are learned from expert demonstrations.

However, such an approach suffers from the need for a collection of diverse and large-scale demonstrations, in our case the in-out pairs of CA in different scenarios. So, performance heavily depends on the quality and diversity of the provided demonstrations.

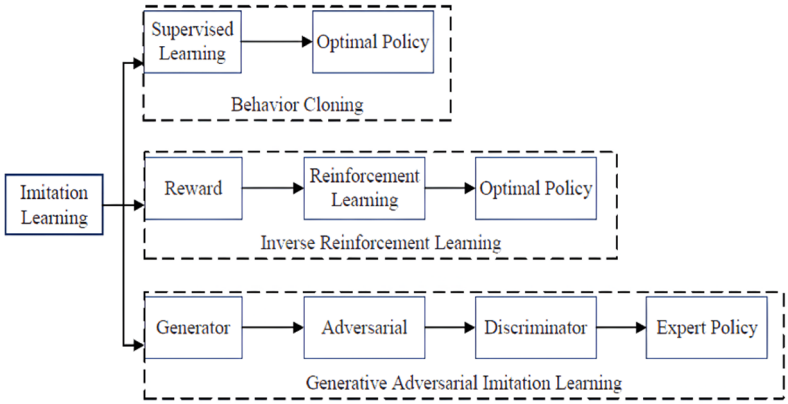

Figure 6: Classification of imitation learning (Hua, Zeng, Li, & Ju, 2021).

3.3 Learning of a Control Allocator Using Transfer Learning

Transfer Learning (TL) involves transferring knowledge gained from one task to improve learning in another related task, this can be achieved by transferring models or policies learned.

Such an approach is so efficient as it reduces the data and the training time by providing pre-trained models or knowledge from related tasks.

However, transferred models may perform poorly if the source and target domains differ significantly.

While each learning paradigm has its strengths, they also come with significant challenges. Combining these approaches (e.g., applying transfer learning to adapt DRL policies to new tasks) can leverage their advantages while mitigating individual limitations. As an example, BC can be used to mimic the existing CA, and DRL can be used to achieve a generalization of the behavior of the cloner through TL.

The choice of the technique will depend on the needs and the suitability of whether the expert exists or not and the quality of the data used in the learning, Table 3.

4 CURRENT RESEARCHES

Concerning the work that has been conducted in learning CA, researchers used BC and DRL till now.

Recent work by Khan, Mobeen, Rajput, and Riaz (2024), explored the use of an ANN in CA to account for the nonlinear CA problem formulation and compared it to quadratic programming (QP) allocation. They tested the two approaches on an aircraft equipped with redundant control effectors. The two approaches showed similar performance with reduced run time for the proposed scheme of ANN-based control allocator.

Table 3: Comparison of Learning Types: Imitation Learning, Deep Reinforcement Learning, and Transfer Learning.

Type	Pros	Cons
Imitation Learning	-Facilitates rapid skill acquisition. -Effective in structured environments.	-Limited to mimicking demonstrated behaviors. -Challenges in dealing with unstructured data.
Deep Reinforcement Learning	-Enables autonomous decision-making. -Suitable for complex, dynamic environments.	-Requires extensive computation and training time. -Prone to instability and convergence issues.
Transfer Learning	-Leverages pre-existing knowledge for new tasks. -Reduces data requirements for training.	-Performance heavily depends on source-target similarity.

The optimization problem has been studied by Skulstad, Li, Fossen, and Zhang (2023) in their article “Constrained control allocation for dynamic ship positioning using deep neural network” where they proposed a deep Learning Technique for the problem of dynamic positioning in marine vessels and trained to perform low-speed maneuvering and stationkeeping in a simulated environment. The study used behavioral cloning to capture the intricate relationships between control inputs and CA output.

Using custom loss functions, the network ensures motion objectives and thruster constraints are met. The solution was as performant as an SQP-based CA however the constraints were violated leading to another research by the same authors comprising the addition of a costume NN layer to enforce hard constraints (Raghunathan, Skulstad, Li, & Zhang, 2023). The layers use a specific activation function that inherently respects the constraints.

Wu and Litt (2023) explored the use of deep reinforcement learning (RL) for controlling an aircraft equipped with tributed electric propulsion (DEP) in their article "Reinforcement Learning Approach to Flight Control Allocation With Distributed Electric Propulsion". The research demonstrates that RL can efficiently manage the allocation of thrust among multiple electric engines, achieving stable flight even under conditions of actuator failure without a knowledge of the efficiency matrix of the CA.

As with any data-based approach, the learning-based CA goes through the following steps:

4.1 Data Collection and Preprocessing

All studies emphasize the importance of comprehensive data collection and preprocessing. The state of art precised clearly the relationship between the quality of the data processed and the generalization capability of the resulting CA.

For the BC, neither change in the number of actuators nor failure scenarios were considered in the data collection (Skulstad, Li, Fossen, & Zhang, 2023), (Khan, Mobeen, Rajput, & Riaz, 2024). For the DRL case, coordinated turns and one-wing fan failure were taken into the data collection scenarios and tested afterward.

4.2 Model Design and Training

From the previously presented learning methods, only two approaches have been used in the state of the art:

4.2.1 Behavioral Cloning

This approach has been used because a well-working CA already exists.

The neural network models for BC contained multiple layers to capture complex nonlinear relationships between in-out pairs. Recurrent neural nets have been used to facilitate constraining rates (Zare, Kebria, Khosravi, & Nahavandi, 2023). Specific attention was given to preventing overfitting through techniques such as dropout and regularization.

Constraints were taken into account in the development of the NN by introducing them as penalties in the loss function, or by clipping the outputs violating the constraints.

Multi-objective optimization was also taken into account to achieve secondary objectives such as minimizing power consumption and maximizing comfort. A deep autoencoder network was used to lay a hand on both the tracking and the allocation capabilities of the NN.

A small study on the change of the parameters of the neural network has been done to show the effect of the number of layers and their dimensions on the convergence of the output.

4.2.2 Deep Reinforcement Learning

The DRL has been used because no CA was developed for such a system. The DRL agent was trained by interacting with the system model in a high-fidelity simulation.

Proximal Policy Optimization (PPO) was used with a reward function designed to encourage the minimization of the tracking error.

4.3 Results

The integration of deep learning and reinforcement learning into control allocation systems offers significant advantages as it leverages better handling of complexities and non-linearities plus an adaption to actuator failures if covered in the training phase.

The reviewed studies collectively demonstrate that learning-based control allocation methods reach similar performances to traditional approaches in terms of allocation and better performances in terms of multi-objective optimization.

Comparing the execution time of both approaches doesn't seem to give a collective answer about the real-time performance of the two approaches.

5 APPLICATION CASE TO SMART CHASSIS

As all the blocks are presented, there is nothing left but to assemble them.

Let's retake the same example used by Kissai, Monsuez, Mouton, Martinez, and Tapus (2019); control of the yaw rate $\dot{\psi}$ using the forces generated by the existing actuators (VDC and ARS).

The relationship between the yaw moment $M_z$and the different longitudinal and lateral forces were shown in (1). The non-linearities and coupling are clearly seen in the equation. What if forces in the Z-direction are considered, or a different set of actuators is used, or a change in the vehicle parameters (e.g.: mass) or the environment parameters happens, or prediction needs for a more comfortable ride are desired? Such needs clearly cause coupled, highly changing control allocation formulations.

Here comes machine learning (ML) in solving optimization and generalization concerns. Add the fact that if the neural nets have memory capabilities like long short-term memory (LSTM), then predicting future states and behaviors based on past data can be possible. Complicated nonlinear and time-varying vehicle dynamics can be considered, under significant uncertainty and variability in real-life driving conditions, providing better reliability. ML also has the ability to personalize control strategies to the driver.

As with any ML problem, the following steps should be followed:

- Data collection: As the driving environment is so dynamic and restrictive, these limitations can oppose with the big performance and comfort expectations. High safety and adaptability should be guaranteed. So, extensive driving data should be gathered for the training of the NN model under various conditions, including sensor readings, actuator inputs, and responses.
- Model training: The NN should be trained to produce the optimal output by taking into consideration all the constraints and objectives of the developer.
- Integration: The trained ML models should be embedded into the CA algorithm with the high-level controller and the actuator. Allocation and tracking performance should be checked. For the RL implementation, the ML model continuously learns by the interaction with the environment.
- Testing and validation: A detailed comparison with hopefully an already developed CA should be made.

5.1 Reinforcement Learning for Control Allocation

Reinforcement Learning (RL) is used when the efficiency matrix is uncertain. This can be the case with complex dynamics of the system studied. Here, the solution would be learned from the interaction with the environment.

- Define the environment: The hardest part is the choice of a suitable reward function. Depending on the objectives, reward can include factors like maintaining vehicle stability, minimizing tire slip, and ensuring passenger comfort.
- Model design: An RL algorithm should be selected to learn the optimal control policy.
- Model training: The RL agent interacts with the simulated environment, learning to optimize control actions based on rewards.
- Model deployment: The trained RL model is deployed in the vehicle's control system, continuously refining its strategy based on real-time data.

5.2 Behavioral Cloning for Control Allocation

This Behavioral Cloning (BC) can effectively learn and replicate the control strategies of an already-developed expert CA.

The main steps that should be followed are:

- Define the environment: As opposed to the plants used in the current state-of-the-art, the driving environment is so dynamic pushing the car to its functional limits more often. So, we should ensure that the data capture a wide range of driving scenarios to train a robust model.
- Data Collection and Preparation: Driving data should be collected from the advanced CA systems: Inputs: CA input (yaw rate in our case), system and environment specification (vehicle speed, vehicle mass, steering angle...); Outputs: CA output (longitudinal and lateral forces in our case).
- Model design and training: A neural network model with appropriate layers and neurons should be chosen. A great importance should be given to the type of the NNs used as it highly depends on the nature of the expert. Then, the model should be trained using the dataset, to minimize the error between the predicted and actual control actions. Multi-objective optimization can be achieved by integrating other objectives like energy optimization in the neural network loss function.

- Model deployment: The model is then implemented in the vehicle's onboard computer to ensure real-time execution for predicting control actions based on current sensor inputs.

For transfer learning (TL), the idea of combining BC and RL seems relevant. As an example, BC can be used to mimic the existing CA, and DRL can be used to achieve a generalization of the behavior of the cloner through TL to introduce adaptability. However, a study of the limitations of BC should be conducted to justify the use of TL as the automotive field is so restrictive in terms of computational power.

6 CONCLUSIONS

In this paper, three learning approaches to implement control allocation have been introduced. Control orchestration needs for chassis systems have been presented and the limitations of optimization-based coordination have been discussed. The main results obtained from the state of the art present a big motivation to satisfy our needs in terms of generalization, prediction, and fidelity.

The coming work will concern an in-depth discussion of the main missing points and doubts that have to be revisited in our future studies relating to imitation learning in general, and learning of CA more specifically. The example taken in this article will be used to see the reliability of imitation learning in CA problems.

REFERENCES

Vries, P., & Van Kampen, E.-J. (2019). Reinforcement learning-based control allocation for the innovative control effectors aircraft. In *Proceedings of the AIAA Scitech 2019 Forum*. https://doi.org/10.2514/6.2019-0144.

Raghunathan, R. N., Skulstad, R., Li, G., & Zhang, H. (2023). Design of constraints for a neural network based thrust allocator for dynamic ship positioning. In *IECON 2023 - 49th Annual Conference of the IEEE Industrial Electronics Society* (pp. 1–6). https://doi.org/10.1109/IECON51785.2023.10312100.

Khan, H. Z. I., Mobeen, S., Rajput, J., & Riaz, J. (2024). Nonlinear control allocation: A learning-based approach. *arXiv*. https://arxiv.org/abs/2201.06180.

Skulstad, R., Li, G., Fossen, T. I., & Zhang, H. (2023). Constrained control allocation for dynamic ship positioning using deep neural network. *Ocean Engineering, 279*, 114434. https://doi.org/10.1016/j.oceaneng.2023.114434.

Wu, K. C., & Litt, J. S. (2023). *Reinforcement learning approach to flight control allocation with distributed electric propulsion* (NASA Technical Memorandum No. 20230014863). National Aeronautics and Space Administration, Glenn Research Center. https://ntrs.nasa.gov/.

Skrickij, V., Kojis, P., Šabanovič, E., Shyrokau, B., & Ivanov, V. (2024). Review of integrated chassis control techniques for automated ground vehicles. *Sensors, 24*(2), 600. https://doi.org/10.3390/s24020600.

Hua, J., Zeng, L., Li, G., & Ju, Z. (2021). Learning for a robot: Deep reinforcement learning, imitation learning, transfer learning. *Sensors, 21*(4), 1278. https://doi.org/10.3390/s21041278.

Zare, M., Kebria, P. M., Khosravi, A., & Nahavandi, S. (2023). A survey of imitation learning: Algorithms, recent developments, and challenges. *arXiv*. https://arxiv.org/abs/2309.02473.

Johansen, T. A., & Fossen, T. I. (2013). Control allocation—A survey. *Automatica, 49*(5), 1087–1103. https://doi.org/10.1016/j.automatica.2013.01.035.

Kissai, M. Optimal Coordination of Chassis Systems for Vehicle Motion Control. Automatic Control Engineering. Universite´ Paris Saclay (COmUE), 2019. English. ⟨NNT : 2019SACLY004⟩.

Norouzi, A., Heidarifar, H., Borhan, H., Shahbakhti, M., & Koch, C. R. (2023). Integrating machine learning and model predictive control for automotive applications: A review and future directions. *Engineering Applications of Artificial Intelligence, 120*, 105878. https://doi.org/10.1016/j.engappai.2023.105878.

Kissai, M., Monsuez, B., Mouton, X., Martinez, D., & Tapus, A. (2019). Model predictive control allocation of systems with different dynamics. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)* (pp.4170–4177). https://doi.org/10.1109/ITSC.2019.8917438.

Kissai, M., Monsuez, B., & Tapus, A. (2017, June). Current and future architectures for integrated vehicle dynamics control.

Yuan, Y., Chen, L., Wu, H., & Li, L. (2022). Advanced agricultural disease image recognition technologies: A review. *Information Processing in Agriculture, 9*(1), 48–59. https://doi.org/10.1016/j.inpa.2021.01.003.

Latif, S., Cuayáhuitl, H., Pervez, F., Shamshad, F., Ali, H. S., & Cambria, E. (2021). A survey on deep reinforcement learning for audio-based applications. *arXiv*. https://arxiv.org/abs/2101.00240.

Dey, S., Marzullo, T., Zhang, X., & Henze, G. (2023). Reinforcement learning building control approach harnessing imitation learning. Energy and AI, 14, 100255. https://doi.org/10.1016/j.egyai.2023.100255.

Manuel Davila Delgado, J., & Oyedele, L. (2022). Robotics in construction: A critical review of the reinforcement learning and imitation learning paradigms. *Advanced Engineering Informatics, 54*, 101787. https://doi.org/10.1016/j.aei.2022.101787.