2025/2026

Bureau d'Etudes Industrielles
INP ENSEEIHT / EasyMile

Table des Matières

Présentation du Sujet

Contexte

L'EZDolly est un chariot de manutention autonome électrique développé par EasyMile et TLD, conçu pour le transport de bagages et de fret dans les aéroports. L'EZDolly est capable de transporter jusqu'à 7 tonnes et est compatible avec les principaux types de conteneurs ULD (Unit Load Devices). Son système de transfert de charge automatisé optimise les flux logistiques, en réduisant les temps de transfert et en maximisant l'efficacité sans intervention manuelle.

Pour plus d'informations, voici une vidéo de présentation du système:
TractEasy reveals the definitive EZDolly

L'EZDolly est doté d'une configuration de traction multi-moteurs avec un moteur électrique par roue. Cela implique plusieurs défis: la synchronisation précise du couple est essentielle pour un mouvement fluide, éviter l'usure inégale du système et optimiser l'efficacité. La répartition du couple doit également assurer la stabilité du véhicule lors des changements de direction ou de vitesse, et garantir sa sûreté en cas de défaillance moteur. Enfin, une gestion de l'énergie optimisée via la répartition du couple est cruciale pour assurer une autonomie suffisante pour les opérations.

Objectif

L'objectif de ce bureau d'études est d'analyser les stratégies d'allocation de couple pour le système multi-moteurs de l'EZDolly. La méthode consistera à répartir entre quatre moteurs un couple d'entrée global:

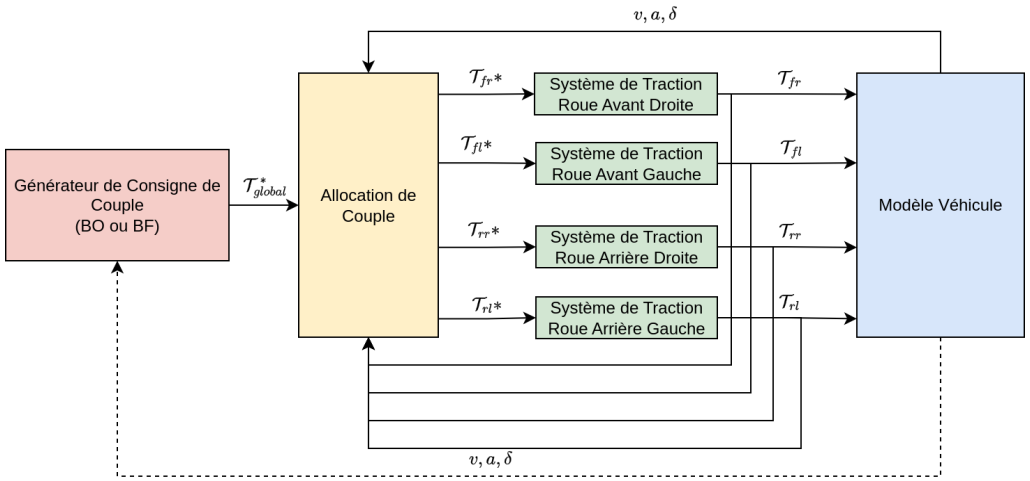


Schéma simplifié de la répartition du couple.
Note: la gestion de l'angle de braquage des roues δ n'est pas modélisée ici.

Les moteurs asynchrones de l'EZDolly présentent un rendement réduit à faible couple. Le véhicule fonctionnant principalement à basse vitesse et rarement à pleine charge, une répartition égale du couple entre les moteurs n'est pas la solution la plus efficace. Il est donc essentiel de mettre en œuvre une stratégie d'allocation du couple intelligente pour optimiser la consommation énergétique globale du véhicule.

Cette étude débutera par une revue exhaustive de la littérature scientifique existante, afin d'identifier les différentes méthodes et approches développées pour l'allocation de couple dans des contextes similaires.

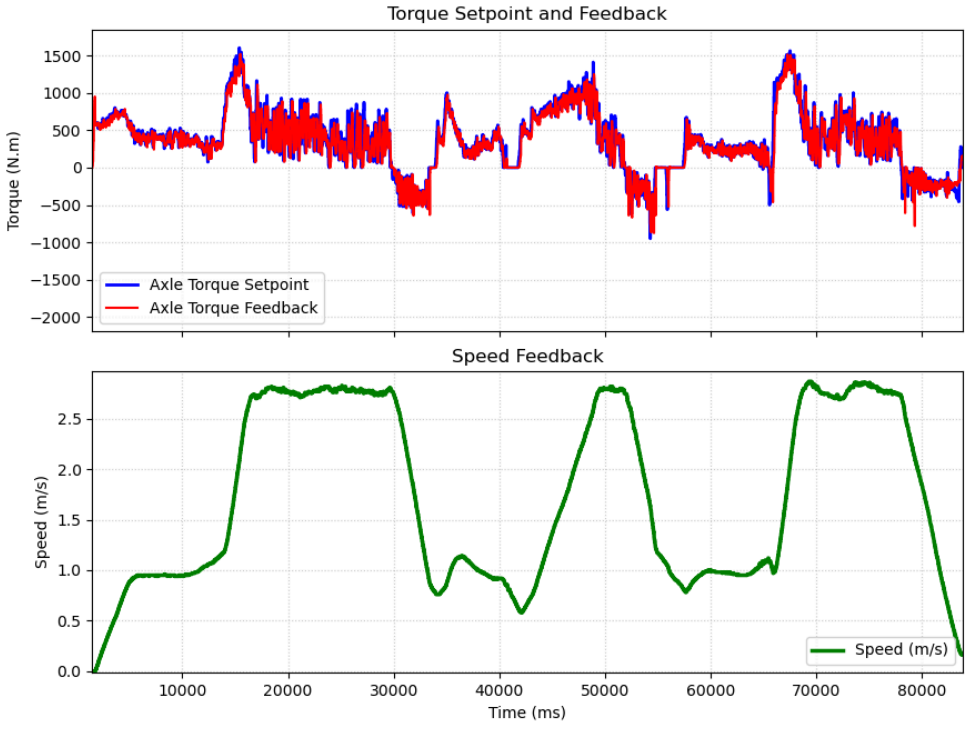
La phase de recherche bibliographique permettra la sélection d'une ou plusieurs méthodes jugées les plus pertinentes pour l'EZDolly selon un ensemble de critères, incluant notamment:

- **Efficacité:** mesurer les performances en termes de précision de suivi du couple et de consommation de l'énergie.
- **Complexité:** évaluer la facilité de mise en œuvre et le coût de calcul.
- **Adaptabilité:** évaluer la capacité de la méthode à s'adapter à différents scénarios de conduite ou d'utilisation de l'EZDolly.
- **Stabilité:** garantir la stabilité du système, même en présence de perturbations ou de variations des conditions de fonctionnement.
- **Robustesse:** évaluer la performance de la méthode face aux incertitudes et aux erreurs de mesure.

La ou les méthodes sélectionnées seront ensuite prototypées en simulation. La simulation devra être dans un environnement Python, afin de simplifier le développement notamment par le biais des librairies scientifiques disponibles. L'étude en simulation permettra de valider et affiner la stratégie d'allocation de couple. Elle mettra en évidence les performances de la stratégie étudiée sur une variété de scénarios de tests réalistes et représentatifs des conditions d'utilisation de l'EZDolly.

L'étude se concentrera tout d'abord sur la dynamique longitudinale, c'est-à-dire que les performances seront étudiées sur un suivi en ligne droite, sans considération de la dynamique latérale. La caractérisation des performances sera à évaluer sur différents scénarios pouvant inclure:

- **Boucle ouverte simple:** suivi d'une séquence de consignes de couple simples (e.g. rampes)
- **Boucle ouverte sur consignes réelles:** suivi d'une séquence de consignes de couple issues de tests véhicules (fourni en csv)
- **Variation de charge:** simuler l'effet de différents charges transportées sur la performance de l'allocation du couple
- **Situations de défaillance:** évaluer la capacité de la stratégie à maintenir un niveau de performance acceptable ou à réagir de manière sécuritaire en cas de défaillance.



Exemple de dataset pour roulage à 10 km/h à vide (csv fourni)

Ensuite, selon le temps restant, différents axes d'exploration seront possibles, comme l'ajout d'un contrôleur de vitesse pour générer les consignes de couple en boucle fermée, puis la considération de la dynamique latérale en considérant le suivi de virages, ou enfin inclure un modèle de batterie pour évaluer l'autonomie du véhicule.

Hypothèses

Dans un premier temps, il est conseillé de simplifier l'étude par la formulation d'hypothèses. Ces hypothèses peuvent par exemple porter sur différents éléments comme:

- **Transmission couple/roue:** roulement sans glissement entre le pneu et la surface de la route.
- **Dynamique du véhicule:** rigidité du châssis, répartition du poids uniforme, aérodynamisme et résistance au roulement connus, suspension simplifiée, etc.
- **Système de traction:** fonctions de transfert des moteurs simples, limitations du couple et rendements connus, transmission à rapport fixe, etc.
- **Contrôleur:** disponibilité et précision des mesures capteur (e.g. vitesse, braquage, couple, …) , fréquence d'échantillonnage fixée, etc.
- **Environnement:** pente, conditions d'adhérence et vitesse du vent connues

Il est recommandé d'adopter une approche progressive : débuter avec une modélisation simplifiée puis complexifier au fil de l'étude.

Livrables Attendus

- Dossier d'étude et de validation de la stratégie d'allocation de couple
- Prototype de la / des stratégie(s) choisie(s) dans une simulation Python
- Documentation associée au simulateur

Données d'Entrée

Bibliographie

Il existe plusieurs stratégies d'allocation, certains papiers regroupent et analysent diverses méthodes, offrant ainsi une approche exhaustive du sujet. Une première analyse du sujet est proposée, par exemple, dans *Control Allocation - A Survey* de Johansen & Fossen.

Modélisation

Il est conseillé de commencer la modélisation en parallèle de l'étude bibliographique.

De manière générale, la modélisation d'un véhicule comprend deux grands axes:

- **Modèle longitudinal:** décrit le mouvement du véhicule selon son axe principal (vitesse, accélération, freinage). Il intègre les forces de propulsion (moteur), de résistance (aérodynamique, roulement, gravité), l'inertie, les systèmes de freinage et l'interaction pneu-route.
- **Modèle latéral:** décrit le mouvement perpendiculaire à l'axe longitudinal (e.g. virages). Il prend en compte la géométrie de la suspension/direction, les forces latérales des pneus, les transferts de charge, les moments de lacet et la dynamique de roulis/tangage.

L'intégration de ces modèles forme un modèle dynamique complet. Bien que ces modèles soient couplés, il est possible sous certaines conditions de les découpler afin de simplifier le modèle global.

Il est également important de noter la distinction entre les modèles dynamiques et cinématiques. Bien que les modèles dynamiques (qui prennent en compte les forces) soient plus réalistes, il est possible de simplifier le problème, notamment pour le modèle latéral, en se concentrant uniquement sur la cinématique (l'étude du mouvement).

Développement du Prototype

Environnement de Développement

IDE

Du support sera fourni pour la mise en place de l'environnement de développement. Il existe des IDE spécifiques pour Python (ex: Spyder, PyCharm…), il est possible d'utiliser des IDE plus génériques (ex: VSCode), ou plus minimaux (ex: Vim). Le choix de l'environnement de développement est libre.

Librairies

Un avantage de l'utilisation de Python sont les nombreuses librairies scientifiques à disposition pour faciliter le développement du prototype. Par exemple, *scipy* offre des algorithmes d'optimisation intéressants, et *control-toolbox* simplifie la modélisation de système tout en intégrant des stratégies de contrôle classiques.

Gestion des paquets

Il est **fortement recommandé** de mettre en place un environnement virtuel permettant une meilleure gestion des dépendances afin d'isoler les paquets utilisés pour ce projet, et de s'assurer que chacun code avec les mêmes dépendances.
Par exemple, il est possible d'utiliser *venv* qui permet rapidement de créer un environnement virtuel et de définir les différentes dépendances du projet.
Voir 12. Environnements virtuels et paquets — Documentation Python 3.12.0

Gestion des versions

Il est conseillé de mettre en place un dépôt *Git* pour faciliter la gestion des versions et l'intégration des différents sous systèmes de la simulation.

Exemple de tutoriel git: Learn Git Branching

Implémentation

> ⚠ *Avant de commencer à coder, il est **fortement conseillé** de commencer par définir les interfaces, l'architecture et les différentes fonctions afin de faciliter l'implémentation et l'intégration des différents sous systèmes.*

Une première étape est d'établir les différents éléments que devraient comporter le simulateur, par exemple: génération des consignes, contrôleur, modèle, boucle de simulation, affichage des résultats, etc.

Architecture typique en Python

Il existe plusieurs types d'architecture en Python, les principales étant le "*flat layout*" et le "*src layout*" (src layout vs flat layout — Python Packaging User Guide). Le choix de l'architecture est laissé libre.

Test du code

La mise en place de tests unitaires et d'intégration permet de vérifier chaque fonction du simulateur et de détecter les régressions. Il existe plusieurs méthodes pour mettre en place des tests unitaires, une approche classique est d'utiliser Pytest.

Il est aussi possible d'automatiser les tests via une CI (Continuous Integration), ou même de lancer les tests automatiquement à chaque commit git (git hooks).

Afin de simplifier le test de chaque composant de la simulation, il est conseillé de bien séparer les composants dans des modules distincts.

Bonnes pratiques

Style de code

Il est important de garder une uniformité dans le code pour une meilleure intégration des différentes fonctions.
Le *PEP 8* est une convention classique de style de code (PEP 8 – Style Guide for Python Code). Afin de faciliter l'application de cette convention, il est conseillé de paramétrer son IDE avec des outils de vérification de style.

Importations

Il est recommandé de définir des paquets de façon à faciliter l'importation du code des différents sous systèmes, ex:

```
from simulation.vehicle_model import EZDolly
```

Au lieu de faire des importations avec des chemins codés en dur (peu robuste) comme

```
from ..vehicle_model import EZDolly
```

Voir 5. The import system — Python 3.12.0 documentation

Simulation et Validation

Afin de valider la méthode, il est nécessaire de définir différents scénarios de tests unitaires et de simulation. Une première étape pour simplifier le problème est de considérer une séquence de consigne de couple simple (e.g. valeurs constantes ou forme trapézoïdale) ainsi que l'absence de perturbations. Les scénarios pourront être complexifiés au fur et à mesure de l'étude.

Note: quelques informations sur les données de roulage fournie en csv:
- ces mesures sont faites à vide (pas d'ULD chargé sur le véhicule)
- lorsque le véhicule est à l'arrêt, une commande de couple de freinage est envoyée par défaut mais il n'est pas appliqué (le véhicule est maintenu à l'arrêt par un frein de stationnement)

Pour valider la performance de la stratégie, il est intéressant de pouvoir afficher des résultats sous forme de courbes (e.g. courbe de réponse de la vitesse, accélération et couple) et/ou de métriques (e.g. précision du suivi de couple, consommation énergétique totale, etc.).

Fiche Véhicule

Plateforme

- Poids à vide : 4900 kg
- Dimensions :
  - Longueur : 6,4 m
  - Largeur : 3,0 m
  - Hauteur : 3,0 m
- Roues :
  - Rayon : 0,24 m
- Châssis :
  - Empattement : 4,8 m
  - Voie : 2,2 m

Conditions d'Opération

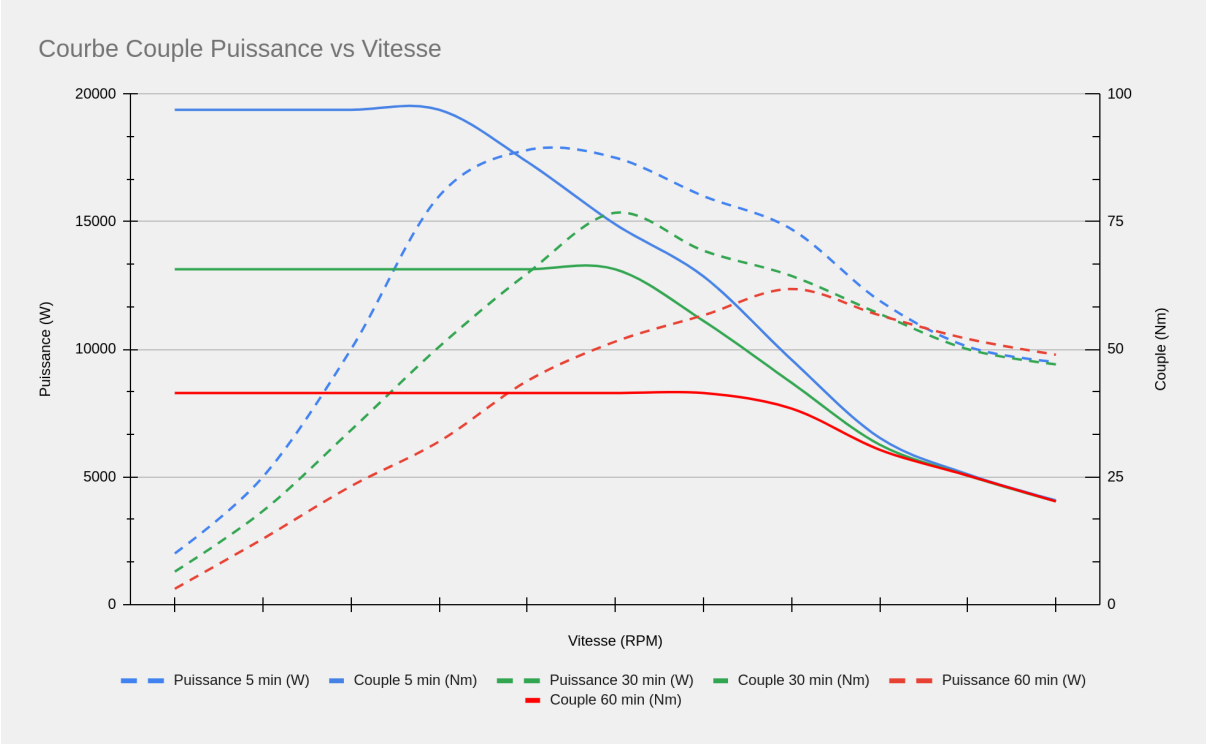
- Charge maximale: 7T
- Vitesse maximale: 15 km/h

Traction

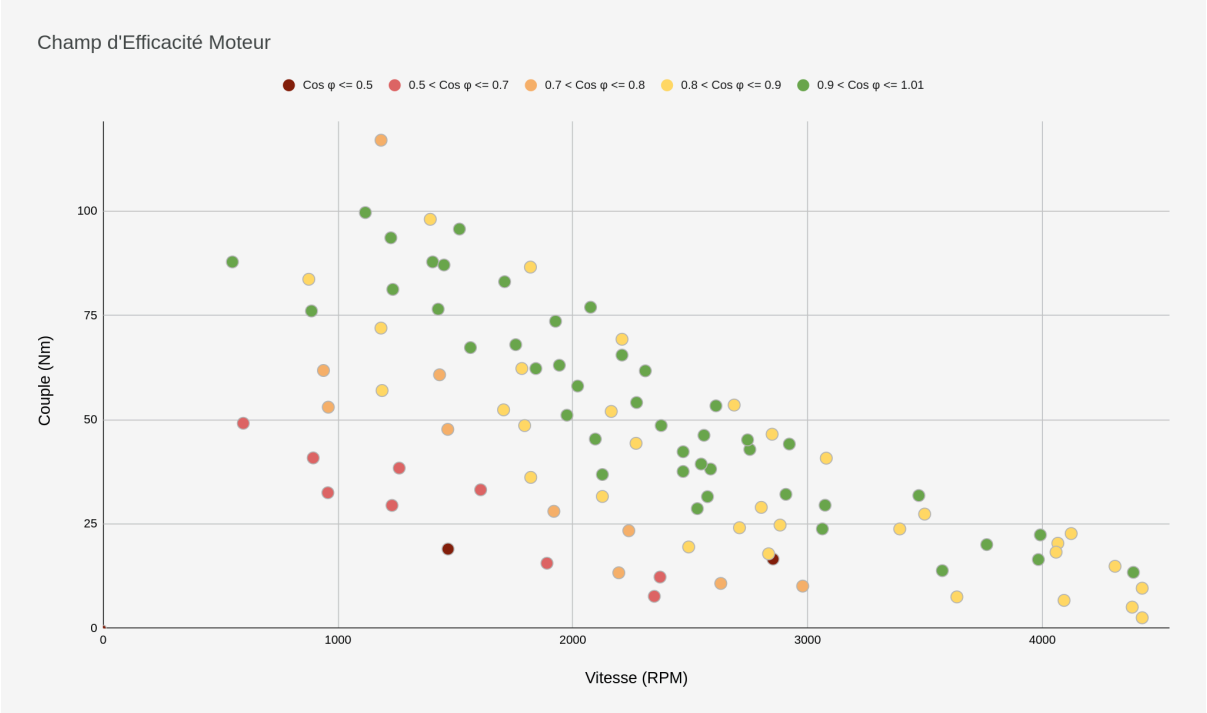
- Motorisation: 4 moteurs électrique asynchrones (un par roue)
- Rapport de réduction moteur/essieu: 26

Plaque signalétique:

Puissance Nominale (W)	8000
Vitesse Nominale (RPM)	2200
Fréquence (Hz)	55
Couple Nominal (Nm)	34.3
Courant Nominal (A)	109
Facteur de Puissance ($\cos\Phi$)	0.91
Nombre de pôles	4



Courbes caractéristiques du moteur



Couple en fonction de la vitesse, par facteur de puissance

Les données affichées ci-dessus sont fournies sous format csv.

Contacts

- Lucas Veit (Équipe Navigation): lucas.veit@easymile.com
- Larbi Omari (Équipe Contrôle): larbi.omari@easymile.com
- Djahid Rabehi (Équipe Contrôle): djahid.rabehi@easymile.com
- Damien Verdier (Équipe Système): damien.verdier@easymile.com
- Maxime Hohl (Équipe Détection): maxime.hohl@easymile.com