

Rapport de projet



Régulateur de Vitesse Adaptatif (ACC)

Master : Systèmes Embarqués et Robotique.
Module : Système Embarqué : Applications et mise en oeuvre

Réalisé par :
MOKHTARI Anass

Encadré par :
Mme. DIB Faiza

Introduction

L'**ACC (Adaptive Cruise Control)**, ou régulateur de vitesse adaptatif, est un système intelligent d'aide à la conduite qui ajuste automatiquement la vitesse du véhicule pour maintenir une distance de sécurité constante avec celui qui le précède. Contrairement à un régulateur classique qui maintient une allure fixe, l'ACC utilise des capteurs (radar, caméras ou laser) pour détecter les variations de trafic : il freine si la voiture de devant ralentit et réaccélère jusqu'à la vitesse de consigne dès que la voie est libre.

Résumé

Ce projet porte sur la conception et le développement d'une application de simulation du **Régulateur de Vitesse Adaptatif (Adaptive Cruise Control – ACC)** basée sur le réseau de communication **CAN (Controller Area Network)**. L'objectif principal est de reproduire le fonctionnement logique et dynamique d'un système ACC réel, en modélisant les interactions entre les différents calculateurs électroniques (ECU), les capteurs virtuels et les actionneurs du véhicule.

La solution proposée repose sur une architecture logicielle modulaire développée en **Python**, intégrant une interface graphique interactive permettant de visualiser le comportement du véhicule, les états du régulateur et les échanges CAN en temps réel. Plusieurs scénarios de conduite ont été simulés afin d'évaluer les performances du système, notamment la conduite sur route libre, le suivi de véhicule, les situations de trafic dense et les freinages brusques.

Les résultats obtenus montrent que le système développé respecte les exigences fonctionnelles et de sécurité définies dans le cahier des charges. Cette simulation constitue ainsi un outil pédagogique efficace pour la compréhension des systèmes ADAS et des communications embarquées, tout en offrant une base extensible pour des développements futurs.

Table des Matières

Résumé.....	I
Table des Matières	II
Liste des Figures	IV
Liste des Abréviations.....	V
Introduction générale	1
Chapitre 1 : Cahier des charges	2
1.1 Introduction	2
1.2 Définitions	2
1.3 Disposition physique	4
1.4 Aperçu opérationnel	6
1.4.1 Initialisation	6
1.4.2 Activation du régulateur de vitesse.....	7
1.4.3 Fonctionnement en mode de contrôle de vitesse (ACC speed control).....	7
1.4.4 Fonctionnement en mode de suivi (contrôle de l'intervalle de temps ACC)..	7
1.4.5 Transition entre les modes de contrôle de vitesse et suivi	8
1.4.6 Annulation du fonctionnement du régulateur de vitesse.....	9
1.5 Conclusion.....	9
Chapitre 2 : Conception et réalisation de l'application de simulation ACC.....	10
2.1 Introduction	10
2.2 Bibliothèques Python utilisées	10
2.2.1 PySide6 (Qt for Python).....	10
2.2.2 Bibliothèques standard Python	11
2.3 Architecture générale du programme	11
2.3.1 Principe ECU + Bus CAN	12

2.3.2	Bus CAN virtuel	12
2.4	Programmation de la logique ACC	13
2.4.1	Machine à états	13
2.4.2	Algorithme de régulation	14
2.5	Simulation du véhicule et de l'environnement	14
2.5.1	Modèle du véhicule ego	14
2.5.2	Véhicule cible et radar	15
2.6	Interface Homme-Machine (IHM)	15
2.6.1	Simulation Dashboard.....	15
2.6.2	CAN Monitor	15
2.7	Synchronisation et temps réel.....	16
2.8	Conclusion.....	16
Chapitre 3 : Tests de simulation et analyse des résultats		17
3.1	Introduction	17
3.2	Méthodologie de test et stratégie de validation	17
3.3	Scénario 1 : Conduite en route libre (Free Drive)	18
3.4	Scénario 2 : Suivi d'un véhicule cible (Follow Vehicle).....	19
3.5	Scénario 3 : Stop & Go.....	20
3.6	Scénario 4 : Freinage brusque du véhicule cible	21
3.7	Scénario 5 : Disparition du véhicule cible.....	21
3.8	Analyse globale des résultats	21
3.9	Limites observées et perspectives d'amélioration.....	22
3.10	Conclusion.....	22
Conclusion générale.....		23

Liste des Figures

Figure 1.1 - Relation entre les véhicules de l'ACC.....	2
Figure 1.2 - Les états du système ACC et ses Transitions	4
Figure 1.3 - Aperçu opérationnel	6
Figure 2.1 - Importation des bibliothèques PySide6.....	11
Figure 2.2 - Importation des bibliothèques standard	11
Figure 2.3 - Envoi d'une trame CAN.....	13
Figure 2.4 - Structure d'une trame CAN	13
Figure 2.5 - Définition des états ACC.....	13
Figure 2.6 - Calcul de la distance cible (Time Gap)	14
Figure 2.7 - Mise à jour du véhicule ego	14
Figure 2.8 - Génération de la mesure radar.....	15
Figure 2.9 - Construction du Dashboard.....	15
Figure 2.10 - Table de visualisation CAN.....	16
Figure 2.11 - Boucle principale de simulation	16
Figure 3.1 - Simulation du régulateur ACC en mode Speed Control (route libre)	18
Figure 3.2 - Simulation du régulateur ACC en mode Time Gap Control (suivi de véhicule) .	19
Figure 3.3 - Trames CAN lors du fonctionnement de l'ACC en mode Speed Control	19
Figure 3.4 - Trames CAN lors du suivi d'un véhicule cible en mode Time Gap Control	20

Liste des Abréviations

Abréviation	Signification
ACC	Adaptive Cruise Control (Régulateur de Vitesse Adaptatif)
ADAS	Advanced Driver Assistance Systems (Systèmes Avancés d'Aide à la Conduite)
CAN	Controller Area Network
ECU	Electronic Control Unit (Calculateur électronique)
HMI	Human–Machine Interface (Interface Homme–Machine)
UI	User Interface (Interface utilisateur)
GUI	Graphical User Interface (Interface graphique utilisateur)
FPS	Frames Per Second (Images par seconde)
PID	Proportional–Integral–Derivative (Correcteur proportionnel–intégral–dérivé)
BS1 / BS2	Brake Switch 1 / Brake Switch 2 (Contacteurs de frein)
Radar ECU	Calculateur radar
Engine ECU	Calculateur moteur
Brake ECU	Calculateur de freinage
Cluster ECU	Calculateur du combiné d'instruments
Lead Vehicle	Véhicule cible précédant le véhicule ego
Ego Vehicle	Véhicule principal contrôlé par le système
Δv	Vitesse relative entre le véhicule ego et le véhicule cible
km/h	Kilomètre par heure
m/s	Mètre par seconde
g	Accélération gravitationnelle
API	Application Programming Interface
QSS	Qt Style Sheets
Qt	Framework graphique multiplateforme
PySide6	Binding Python officiel du framework Qt
dt	Pas de temps de simulation

Introduction générale

L'évolution rapide des systèmes embarqués automobiles a conduit à une intégration croissante de fonctions d'assistance à la conduite, connues sous le nom de systèmes **ADAS (Advanced Driver Assistance Systems)**. Parmi ces fonctions, le **Régulateur de Vitesse Adaptatif (Adaptive Cruise Control – ACC)** occupe une place centrale, car il contribue à la fois au confort de conduite et à l'amélioration de la sécurité routière. Le fonctionnement de ces systèmes repose sur l'interaction de plusieurs calculateurs électroniques (ECU), des capteurs embarqués et des réseaux de communication temps réel, en particulier le **bus CAN (Controller Area Network)**.

Dans ce contexte, l'objectif de ce projet est de concevoir et de développer une **application de simulation du système ACC**, permettant de reproduire son comportement fonctionnel, ses mécanismes de décision et ses échanges de données sur le bus CAN. L'approche retenue repose sur une **simulation logicielle interactive**, offrant à la fois une visualisation graphique du système et un accès détaillé aux communications internes entre les différents calculateurs.

Ce rapport présente ainsi la réalisation complète de cette application, depuis la traduction du cahier des charges jusqu'à la validation du comportement du système à travers des scénarios de conduite représentatifs. L'accent est mis sur la compréhension des mécanismes internes du régulateur adaptatif et sur l'analyse des résultats obtenus dans un cadre pédagogique et expérimental.

Chapitre 1 : Cahier des charges

1.1 Introduction

Le régulateur de vitesse adaptatif (ACC) est une fonction automobile qui permet au système de régulateur de vitesse d'un véhicule d'adapter la vitesse du véhicule à l'environnement de circulation. Un système radar fixé à l'avant du véhicule est utilisé pour détecter si des véhicules se déplaçant plus lentement se trouvent sur la trajectoire du véhicule ACC. Si un véhicule se déplaçant plus lentement est détecté, le système ACC ralentira le véhicule et contrôlera le dégagement, ou l'intervalle de temps, entre le véhicule ACC et le véhicule avant. Si le système détecte que le véhicule à l'avant n'est plus sur la trajectoire du véhicule ACC, le système ACC accélérera le véhicule jusqu'à la vitesse réglée pour le régulateur de vitesse. Cette opération permet au véhicule ACC de ralentir et d'accélérer de manière autonome avec le trafic sans intervention du conducteur. La méthode par laquelle la vitesse du véhicule ACC est contrôlée se fait via la commande d'accélération du moteur et le fonctionnement limité des freins.

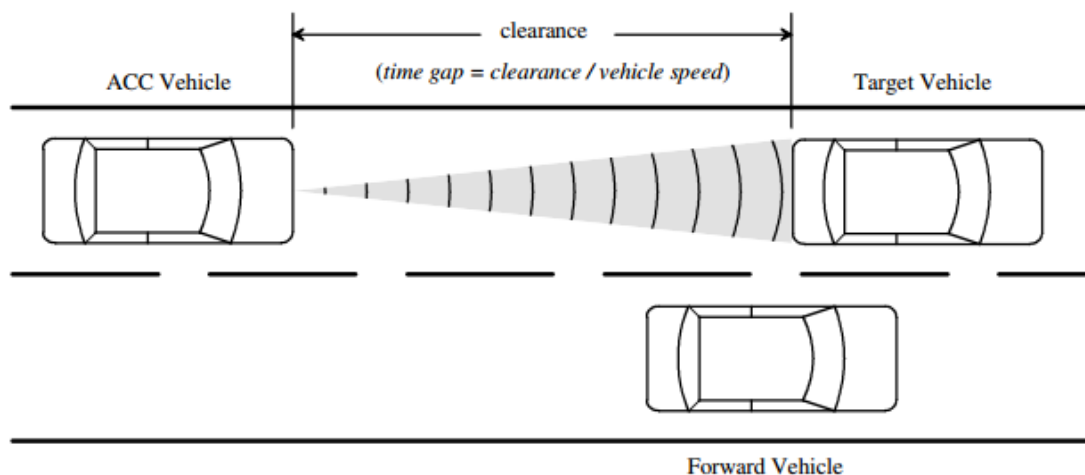


Figure 1.1 - Relation entre les véhicules de l'ACC

1.2 Définitions

Régulateur de vitesse adaptatif (ACC) - Une amélioration d'un système de régulateur de vitesse conventionnel qui permet au véhicule ACC de suivre un véhicule avant à une distance appropriée.

Commande de freinage active - une fonction qui provoque l'application des freins sans que le conducteur n'appuie sur la pédale de frein.

Dégagement (Clearance) – distance entre la surface arrière du véhicule avant et la surface avant du véhicule ACC.

Véhicule à l'avant (Forward Vehicle) – l'un quelconque des véhicules devant et se déplaçant dans la même direction et circulant sur la même route que le véhicule de l'ACC.

Vitesse réglée – la vitesse de déplacement souhaitée du régulateur de vitesse définie par le conducteur et correspond à la vitesse maximale souhaitée du véhicule sous le contrôle de l'ACC.

États du système

- **ACC OFF** – l'accès direct à l'état « ACC actif » est désactivé.
- **ACC standby (état de veille)** – le système est prêt à être activé par le conducteur.
- **ACC active** – le système ACC est en contrôle actif de la vitesse du véhicule.
 - ✓ **État de contrôle de la vitesse ACC** – un sous-état de l'état « ACC actif » dans lequel aucun véhicule avant n'est présent, de sorte que le système ACC contrôle la vitesse du véhicule à la vitesse définie, comme c'est le cas avec les systèmes de régulateur de vitesse conventionnels.
 - ✓ **État de contrôle de l'intervalle de temps ACC** – un sous-état de l'état « ACC actif » dans lequel l'intervalle de temps, ou l'écart, entre le véhicule ACC et le véhicule cible est contrôlé.

Véhicule cible (Target Vehicle) – l'un des véhicules à l'avant sur la trajectoire du véhicule ACC.

Intervalle de temps (Time gap) – l'intervalle de temps entre le véhicule ACC et le véhicule cible. L'« intervalle de temps » est lié à la « dégagement » et à la vitesse du véhicule par :

$$\text{intervalle de temps} = \text{dégagement} / \text{vitesse du véhicule ACC}$$

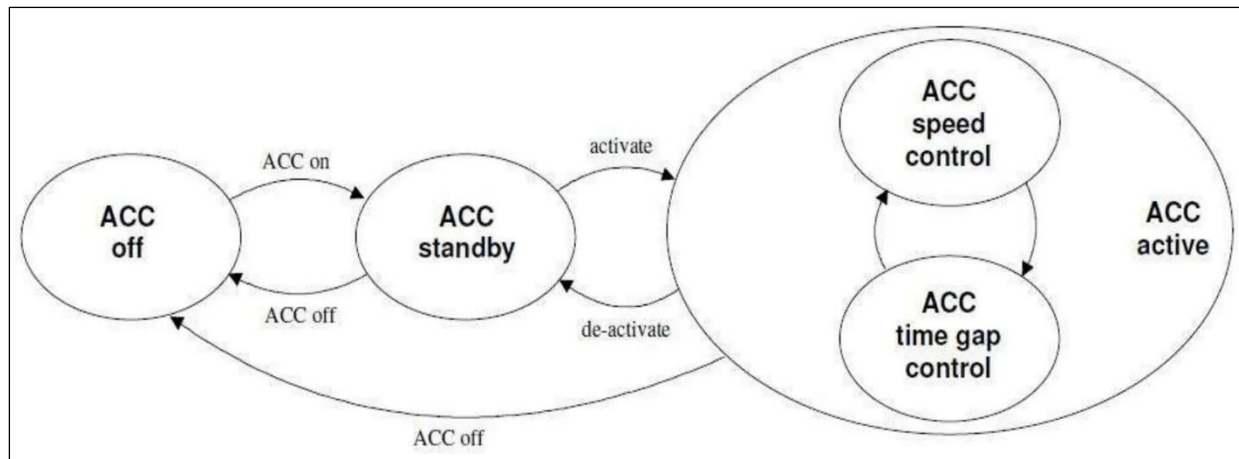


Figure 1.2 - Les états du système ACC et ses Transitions

1.3 Disposition physique

Comme le montre la figure 2, le système ACC se compose d'une série de composants et de systèmes interconnectés. La méthode de communication entre les différents modules se fait via un réseau de communication série connu sous le nom de Controller Area Network (CAN).

Module ACC – La fonction principale du module ACC est de traiter les informations radar et de déterminer si un véhicule à l'avant est présent. Lorsque le système ACC est en « contrôle de l'intervalle de temps », il envoie des informations aux modules de commande du moteur et de commande de frein pour contrôler l'espace entre le véhicule ACC et le véhicule cible.

Module de commande du moteur (Engine Control Module) – La fonction principale du module de commande du moteur est de recevoir des informations du module ACC et du groupe d'instruments et de contrôler la vitesse du véhicule en fonction de ces informations. Le module de commande du moteur contrôle la vitesse du véhicule en contrôlant l'accélérateur du moteur.

Module de commande de frein (Brake Control Module) - La fonction principale du module de commande de frein est de déterminer la vitesse du véhicule en appliquant les freins en réponse à une demande du module ACC. Le système de freinage est une amélioration électronique à la demande du module ACC. Le système de freinage est hydraulique avec ABS et n'est pas un frein par fil.

Groupe d'instruments (Instrument Cluster) – La fonction principale du groupe d'instruments est de traiter les commutateurs de croisière et d'envoyer leurs informations à l'ACC et aux modules de commande du moteur. Le groupe d'instruments affiche également des messages texte et des témoins lumineux pour informer le conducteur de l'état du système ACC.

CAN – Controller Area Network – Le Controller Area Network (CAN) est un réseau standard automobile qui utilise un bus à deux fils pour transmettre et recevoir des données. Chaque nœud du réseau a la capacité de transmettre 0 à 8 octets de données dans une trame de message.

Commutateurs de croisière (Cruise Switches) – Les commutateurs de croisière sont montés sur le volant et permettent au conducteur de commander le fonctionnement du système ACC. Les boutons qui permettent au conducteur de commander l'ACC comprennent :

- **'On'** : place le système dans l'état 'ACC standby'
- **'Off'** : annule le fonctionnement de l'ACC et place le système dans l'état 'ACC off'
- **'Set +'** : active l'ACC et établit la vitesse définie ou accélère
- **'Coast'** : décélère
- **'Resume'** : reprendre pour régler la vitesse
- **'Time Gap +'** : augmenter l'écart
- **'Time Gap -'** : réduire l'intervalle

Interrupteurs de frein (Brake Switches) – Il y a deux interrupteurs de frein, l'interrupteur de frein 1 (BS1) et l'interrupteur de frein 2 (BS2). Lorsque l'un ou l'autre des interrupteurs de frein est activé, le fonctionnement du régulateur de vitesse est désactivé et le système passe à l'état « Veille ACC ».

Feux de freinage (Brake Lights) – Lorsque le module de commande de frein applique les freins en réponse à une demande d'ACC, il allume les feux de freinage pour avertir les véhicules derrière le véhicule ACC qu'il ralentit.

1.4 Aperçu opérationnel

L'interface conducteur du système ACC est très similaire à un système de régulateur de vitesse conventionnel. Le conducteur actionne le système via un ensemble de commutateurs sur le volant. Les commutateurs sont les mêmes que pour un système de régulateur de vitesse conventionnel, à l'exception de l'ajout de deux commutateurs pour contrôler l'intervalle de temps entre le véhicule ACC et le véhicule cible. De plus, il existe une série de messages texte qui peuvent être affichés sur le groupe d'instruments pour informer le conducteur de l'état du système ACC. Le conducteur engage le système ACC en appuyant d'abord sur l'interrupteur ON qui place le système dans l'état « Veille ACC ». Le conducteur appuie ensuite sur l'interrupteur Set pour entrer dans l'état « ACC actif », auquel point le système ACC commence à contrôler le véhicule à la vitesse définie par le conducteur en fonction de l'environnement de circulation.

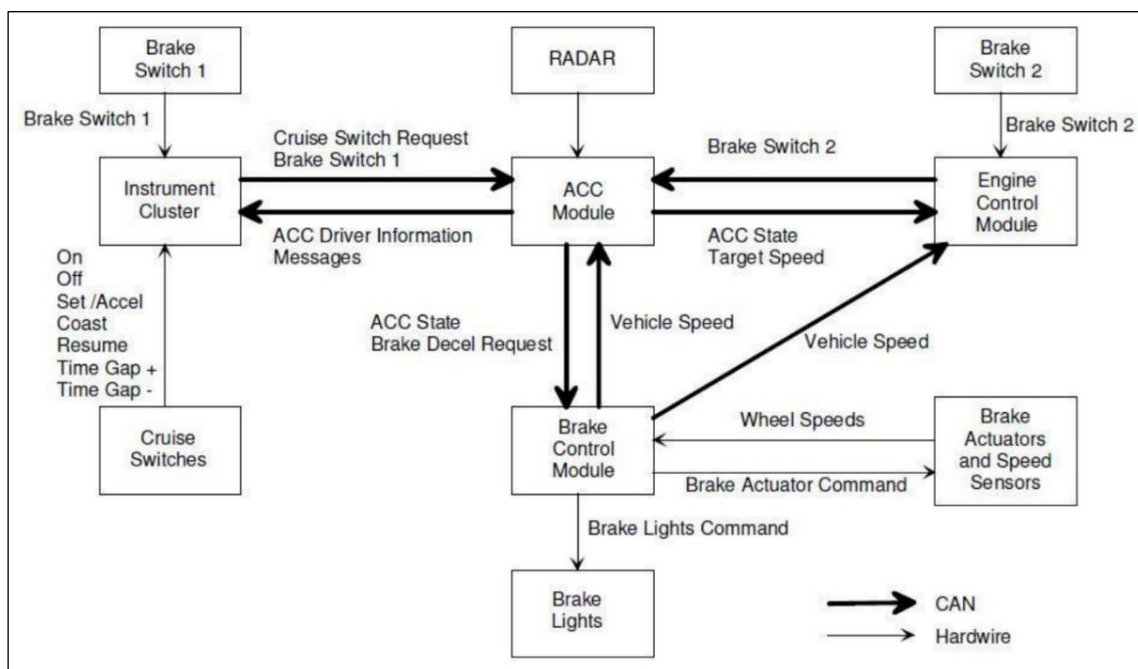


Figure 1.3 - Aperçu opérationnel

1.4.1 Initialisation

Lorsque la clé de contact est en position d'arrêt, aucune alimentation n'est appliquée à l'un des systèmes. Lorsque la clé est mise en position de marche, le système ACC s'initialise à l'état « ACC off ».

1.4.2 Activation du régulateur de vitesse

Entrer en “ACC standby” – Avant que le régulateur de vitesse actif puisse être activé, le conducteur doit d’abord entrer en “ACC standby”. Ceci est effectué par le conducteur en appuyant sur le bouton ACC On. Si aucune défaillance du système n’est présente, le système ACC passera à l’état « ACC standby ».

Saisie de “ACC actif” – Le conducteur passe à l’état « ACC actif » en appuyant sur le bouton « Set » ou « Resume ». Si une vitesse définie antérieure est présente dans la mémoire, le système utilise cette valeur antérieure comme vitesse cible lorsque la fonction Reprendre est enfoncée, sinon la vitesse actuelle au moment où le bouton Définir est enfoncé deviendra la vitesse cible. Les conditions suivantes doivent être remplies pour que le système entre en « ACC actif » en réponse aux interrupteurs de croisière :

- Brake Switch 1 (Contacteur de frein 1) = frein non appliqué
- Brake Switch 2 (Contacteur de frein 2) = frein non appliqué
- Vitesse du véhicule ≥ 40

Lors de l’activation de la commande ACC active, la vitesse du véhicule est contrôlée soit pour maintenir une vitesse définie, soit pour maintenir un écart de temps avec un véhicule en marche avant, selon la vitesse la plus basse.

1.4.3 Fonctionnement en mode de contrôle de vitesse (ACC speed control)

Le fonctionnement dans ce mode est équivalent à celui d’un contrôle de vitesse conventionnel. Si aucun véhicule à l’avant n’est présent dans l’intervalle de temps (Time Gap) ou l’autorisation du système, la vitesse du véhicule est maintenue à la vitesse cible. Le système de commande du moteur contrôle la puissance du moteur pour maintenir la vitesse du véhicule à la vitesse cible (Target Speed).

1.4.4 Fonctionnement en mode de suivi (contrôle de l’intervalle de temps ACC)

Le système ACC entre en mode de suivi ou « ACC time gap control » si le radar détecte un véhicule à l’avant ou dans la distance de dégagement. Pendant ce mode de fonctionnement, le système ACC envoie une vitesse cible au module de commande du moteur et des commandes

de décélération au module de commande de frein pour maintenir l'intervalle de temps défini entre les véhicules.

Contrôle de décélération – Le système ACC décélère le véhicule en abaissant la vitesse cible envoyée au module de commande du moteur et en envoyant une commande de décélération de freinage au module de commande de frein. Pendant les événements de décélération, le module de commande de frein active les feux de freinage.

Contrôle d'accélération - Le système ACC accélère le véhicule en augmentant la vitesse cible envoyée au module de commande du moteur. Le module de commande du moteur essaie d'accélérer le véhicule pour maintenir la vitesse cible.

Réglage de l'écart de temps – Le conducteur peut régler l'écart de temps via les commutateurs « Time Gap + » et « Time Gap – ». L'appui sur l'interrupteur Time Gap + augmente la valeur de l'intervalle de temps et donc l'écart entre les deux véhicules. L'appui sur l'interrupteur Time Gap – entraîne une diminution de la valeur de l'intervalle de temps et donc de la distance entre les deux véhicules.

Réaction à un véhicule lent ou arrêté - Des situations peuvent se produire telles que le système ACC n'est pas en mesure de maintenir l'écart de temps dans l'autorité de décélération du système (0,2 g). L'espace entre le véhicule ACC et le véhicule avant peut diminuer rapidement ou la vitesse minimale du véhicule de 25 km/h peut être atteinte. Dans ces situations, le système ACC passe en « ACC standby » et alerte le conducteur en affichant un message texte « Intervention du conducteur requise » sur le groupe d'instruments et en allumant un carillon audible. Si les freins étaient appliqués par le système ACC, ils seront lentement relâchés. À ce stade, le conducteur doit prendre le contrôle du véhicule.

1.4.5 Transition entre les modes de contrôle de vitesse et suivi

Le système ACC passe automatiquement du contrôle de vitesse à l'intervalle de temps (suivi). Le mode de fonctionnement est déterminé par la plus faible de la vitesse réglée pour le mode de contrôle de vitesse et la vitesse cible pour maintenir l'écart entre le véhicule ACC et un véhicule avant.

Fondamentalement, si aucun véhicule n'est présent dans la distance de dégagement, le système fonctionnera en mode Speed Control, sinon, il fonctionnera en mode Time Gap.

1.4.6 Annulation du fonctionnement du régulateur de vitesse

Le fonctionnement du régulateur de vitesse peut être annulé par l'opérateur ou automatiquement via le système ACC. L'une ou l'autre des conditions suivantes désactivera l'ACC :

- La pédale de frein est enfoncée
- Le bouton « Off » est enfoncé
- Vitesse du véhicule < 25 km/h (ou selon les spécifications du constructeur)
- Une défaillance du système ACC est détectée

1.5 Conclusion

Le régulateur de vitesse adaptatif (ACC) représente une évolution significative par rapport au régulateur de vitesse conventionnel. Grâce à l'utilisation d'un radar avant et à une gestion électronique avancée, il permet au véhicule de s'adapter automatiquement au trafic environnant en maintenant une vitesse cible ou un intervalle de temps (time gap) constant avec le véhicule qui précède. Le système fonctionne de manière autonome dans deux modes principaux :

- le **mode contrôle de vitesse** (speed control), équivalent au régulateur classique en l'absence de véhicule détecté ;
- le **mode suivi** (time gap control), qui assure un suivi fluide et sécurisé en ajustant continuellement l'accélération et la décélération.

L'intégration étroite entre le module ACC, le moteur, le système de freinage, le réseau CAN et l'interface conducteur (commutateurs au volant et affichages au tableau de bord) permet une intervention très progressive et naturelle, tout en conservant au conducteur la possibilité de reprendre le contrôle à tout moment (freinage manuel, bouton OFF, ou situation limite). Les principales limites du système sont clairement identifiées : autorité de décélération limitée (environ 0,2 g dans la plupart des cas), impossibilité de gérer certaines situations d'urgence ou de trafic très dense, vitesse minimale d'activation/fonctionnement, et nécessité d'une intervention humaine lorsque ces limites sont atteintes (passage en standby + alerte). En résumé, le régulateur de vitesse adaptatif constitue aujourd'hui l'une des briques fondamentales des systèmes d'aide à la conduite de niveau 2 (conduite semi-autonome), apportant un réel gain de confort sur longs parcours et sur autoroute, tout en constituant une étape importante vers les générations futures de véhicules à conduite de plus en plus automatisée.

Chapitre 2 : Conception et réalisation de l'application de simulation ACC

2.1 Introduction

Ce chapitre décrit en détail la réalisation logicielle de l'application de simulation du Régulateur de Vitesse Adaptatif (Adaptive Cruise Control – ACC). Il présente les bibliothèques Python utilisées, l'architecture de programmation adoptée, ainsi que la conception de l'Interface Homme-Machine (IHM) telle qu'implémentée dans l'application finale.

L'objectif est de montrer comment le cahier des charges a été concrètement traduit en une application fonctionnelle, modulaire et interactive, permettant l'observation du comportement de l'ACC et des échanges sur le bus CAN.

2.2 Bibliothèques Python utilisées

Le développement de l'application repose sur l'utilisation conjointe de bibliothèques graphiques et de bibliothèques standards Python. Ce choix permet d'assurer à la fois une **interface riche et interactive**, ainsi qu'une **implémentation claire et structurée de la logique de contrôle et de simulation**.

2.2.1 PySide6 (Qt for Python)

La bibliothèque **PySide6** constitue le cœur graphique de l'application. Elle permet la création de l'interface utilisateur, la gestion des événements et le rendu graphique avancé. Grâce à Qt, il est possible de concevoir une interface moderne comprenant plusieurs vues, des widgets personnalisés (jauges, visualisation radar, barres de commande), ainsi qu'une gestion fluide des interactions utilisateur.

PySide6 est également utilisé pour la gestion du temps via les **timers Qt**, ce qui permet de synchroniser l'actualisation de l'interface avec la simulation temps réel du système ACC.

La bibliothèque **PySide6** est utilisée pour la conception de l'interface graphique et la gestion événementielle. Elle permet :

- la création de fenêtres et de widgets, la gestion des événements utilisateur (clics, sliders, boutons), le rendu graphique avancé (dessins personnalisés, animations), et l'utilisation de timers pour la mise à jour temps réel.

Rôle principal : Interface Homme-Machine (Dashboard, CAN Monitor).

```
from PySide6.QtCore import Qt, QTimer, QObject, Signal
from PySide6.QtGui import QPainter, QColor, QPen, QFont
from PySide6.QtWidgets import (
    QApplication, QMainWindow, QWidget, QPushButton,
    QLabel, QSlider, QComboBox, QTableWidgetItem
)
```

Figure 2.1 - Importation des bibliothèques PySide6

2.2.2 Bibliothèques standard Python

Les bibliothèques standard sont utilisées pour la logique interne et la simulation :

- **math** : calculs physiques (vitesse, accélération, angles),
- **time** : horodatage et gestion du temps réel,
- **struct** : encodage et décodage des messages CAN,
- **dataclasses** : définition claire des structures de données,
- **enum** : définition des états de l'ACC,
- **collections.deque** : stockage circulaire des trames CAN.

```
import math
import time
import struct
from dataclasses import dataclass
from enum import IntEnum
from collections import deque
```

Figure 2.2 - Importation des bibliothèques standard

2.3 Architecture générale du programme

Bien que l'application soit développée dans un **fichier unique (main.py)**, son architecture interne est fortement structurée. Cette organisation permet de reproduire le fonctionnement d'un **système automobile distribué**, tout en conservant une lisibilité et une maintenabilité élevées.

2.3.1 Principe ECU + Bus CAN

Le principe fondamental adopté est celui d'une **architecture orientée calculateurs (ECU)**. Chaque fonction du système ACC est implémentée sous la forme d'un ECU logiciel indépendant, communiquant exclusivement via un **bus CAN virtuel**.

Cette approche présente plusieurs avantages :

- elle reflète fidèlement l'architecture des systèmes embarqués automobiles réels,
- elle permet de séparer clairement la logique de chaque sous-système,
- elle facilite l'analyse et la validation des échanges de données.

Les ECU implémentés couvrent l'ensemble de la chaîne fonctionnelle du système ACC, depuis les entrées conducteur jusqu'à l'affichage des informations au tableau de bord

Chaque fonction du système est modélisée sous forme d'un **ECU logiciel**, communiquant uniquement via un **bus CAN virtuel**.

Les ECU implémentés sont :

- DriverInputsECU, RadarECU, ACCECU, EngineECU, BrakeECU, ClusterECU.

2.3.2 Bus CAN virtuel

Le bus CAN virtuel constitue l'élément central de communication entre les ECU. Il permet l'envoi, la réception et la journalisation des trames CAN de manière similaire à un réseau CAN réel. Chaque trame est identifiée par un identifiant d'arbitrage et contient une charge utile de 8 octets, conformément au standard CAN. Ce bus virtuel joue également un rôle pédagogique essentiel, car il permet l'observation directe des échanges entre calculateurs via la vue **CAN Monitor**.

Le bus CAN virtuel permet :

- l'envoi de trames,
- la mémorisation de la dernière trame par identifiant, la journalisation complète pour l'analyse.

```
@dataclass
class CANFrame:
    seq: int
    ts: float
    arb_id: int
    data: bytes
    src: str
```

Figure 2.4 - Structure
d'une trame CAN

```
def send(self, arb_id: int, data: bytes, src: str):
    self._seq += 1
    frame = CANFrame(self._seq, time.time(), arb_id, data, src)
    self._log.append(frame)
```

Figure 2.3 - Envoi d'une trame CAN

2.4 Programmation de la logique ACC

La logique de contrôle du régulateur adaptatif est implémentée dans un ECU dédié, appelé **ACCECU**. Cet ECU constitue le cœur décisionnel du système et applique strictement les règles définies dans le cahier des charges.

2.4.1 Machine à états

Le fonctionnement du système ACC est basé sur une **machine à états finis**. Chaque état représente un mode de fonctionnement précis du régulateur, et les transitions entre états sont déclenchées par des événements tels que l'action du conducteur, la détection d'un véhicule cible ou les contraintes de sécurité.

Cette approche garantit un comportement déterministe, lisible et conforme aux systèmes industriels.

La logique ACC est implémentée à l'aide d'une **machine à états finis**, conforme au cahier des charges.

États implémentés :

- ACC_OFF
- ACC_STANDBY
- ACC_ACTIVE_SPEED_CONTROL
- ACC_ACTIVE_TIME_GAP_CONTROL
- DRIVER_INTERVENTION_REQUIRED

```
class AccState(IntEnum):
    ACC_OFF = 0
    ACC_STANDBY = 1
    ACC_ACTIVE_SPEED_CONTROL = 2
    ACC_ACTIVE_TIME_GAP_CONTROL = 3
    DRIVER_INTERVENTION_REQUIRED = 4
```

Figure 2.5 - Définition des états ACC

2.4.2 Algorithme de régulation

Deux algorithmes de régulation sont implémentés selon le contexte de conduite. En absence de véhicule cible, le système fonctionne en **Speed Control**, maintenant la vitesse à la valeur de consigne définie par le conducteur. Lorsque le radar détecte un véhicule cible, le système bascule automatiquement en **Time Gap Control**, où la distance inter-véhiculaire est régulée à partir d'un temps de suivi configurable.

Deux modes sont implémentés :

- **Speed Control** : maintien de la vitesse consigne,
- **Time Gap Control** : maintien d'une distance cible basée sur le temps inter-véhicule.

```
target_distance = d0 + time_gap * ego_speed
gap_error = measured_distance - target_distance
acc_cmd = 0.05 * gap_error + 0.65 * relative_speed
```

Figure 2.6 - Calcul de la distance cible (Time Gap)

Une **limitation stricte de décélération à 0,2g** est appliquée. En cas de dépassement, le système passe en **Driver Intervention Required**.

2.5 Simulation du véhicule et de l'environnement

Afin d'évaluer le comportement de l'ACC, un modèle simplifié mais réaliste du véhicule et de son environnement a été implémenté.

2.5.1 Modèle du véhicule ego

Le véhicule ego est modélisé par un mouvement longitudinal intégrant la vitesse, l'accélération et la position. Les effets de freinage, d'accélération et de résistance sont pris en compte, tandis qu'un filtrage temporel limite les variations brusques d'accélération afin d'assurer un comportement fluide et réaliste.

Le véhicule ego est simulé selon un modèle longitudinal incluant :

- accélération,
- freinage,
- résistance aérodynamique,
- limitation du jerk pour le confort.

```
velocity += acceleration * dt
position += velocity * dt
```

Figure 2.7 - Mise à jour du véhicule ego

2.5.2 Véhicule cible et radar

Le véhicule cible est simulé de manière indépendante et peut suivre différents scénarios de conduite. Le radar calcule en temps réel la distance longitudinale et la vitesse relative, informations essentielles pour le fonctionnement du régulateur adaptatif.

Le radar calcule :

- la présence du véhicule cible,
- la distance longitudinale,
- la vitesse relative.

```
distance = lead_position - ego_position  
relative_speed = lead_speed - ego_speed
```

Figure 2.8 - Génération de la mesure radar

2.6 Interface Homme-Machine (IHM)

L'interface graphique constitue un élément central de l'application, car elle permet de visualiser le comportement du système ACC de manière intuitive et pédagogique.

2.6.1 Simulation Dashboard

Le **Simulation Dashboard** offre une vue synthétique du système en fonctionnement. Il affiche la vitesse du véhicule, l'état du régulateur, les commandes moteur et frein, ainsi qu'une visualisation 2D de la route et du radar. Cette vue permet au conducteur (utilisateur) d'interagir directement avec le système via les boutons ACC et les pédales virtuelles.

Le Dashboard permet :

- l'affichage de la vitesse (jauge),
- l'état de l'ACC,
- la visualisation 2D de la route et du radar,
- l'interaction conducteur.

```
self.speedGauge = SpeedGauge()  
self.roadView = RoadView()  
self.accStatusLabel = QLabel("ACC OFF")
```

Figure 2.9 - Construction du Dashboard

Cette vue est destinée à l'analyse fonctionnelle et pédagogique.

2.6.2 CAN Monitor

La vue **CAN Monitor** est dédiée à l'analyse des communications internes du système. Elle permet d'observer chaque trame CAN échangée, de filtrer les messages et d'injecter manuellement des trames afin de tester des situations spécifiques. Cette fonctionnalité renforce fortement l'aspect pédagogique et expérimental du projet.

Le CAN Monitor permet :

- la visualisation des trames CAN en temps réel,
- le filtrage par identifiant ou nom,
- l'injection manuelle de messages.

```
self.table = QTableWidgetItem()
self.table.setColumnCount(7)
self.table.setHorizontalHeaderLabels(
    ["Time", "ID", "Name", "Source", "Data", "Decoded", "Seq"]
)
```

Figure 2.10 - Table de visualisation CAN

2.7 Synchronisation et temps réel

La synchronisation de l'application repose sur une **boucle de simulation à pas fixe** pour la logique physique et une **boucle graphique** pour l'affichage. Cette séparation garantit une simulation stable et indépendante des performances de la machine hôte.

L'application utilise :

- une boucle de simulation à pas fixe (20 ms),
- un timer Qt pour l'actualisation graphique (~60 FPS).

```
self.timer = QTimer()
self.timer.timeout.connect(self.core.step_real_time)
self.timer.start(16)
```

Figure 2.11 - Boucle principale de simulation

2.8 Conclusion

Ce chapitre a présenté de manière détaillée la **réalisation complète de l'application de simulation ACC**, depuis le choix des bibliothèques Python jusqu'à la conception de l'interface graphique finale. L'architecture modulaire basée sur des ECU logiciels et un bus CAN virtuel permet une **représentation fidèle du fonctionnement d'un système ACC réel**, tout en restant adaptée à un cadre académique. L'application obtenue constitue une base solide pour l'analyse expérimentale et la validation fonctionnelle, qui seront abordées dans le chapitre suivant.

Chapitre 3 : Tests de simulation et analyse des résultats

3.1 Introduction

Ce chapitre est consacré à la **phase de validation fonctionnelle et comportementale** du système de régulation de vitesse adaptatif (ACC) développé. L'objectif principal est d'évaluer le **bon fonctionnement du système** à travers différents **scénarios de simulation représentatifs de situations de conduite réelles**, tout en vérifiant la conformité aux exigences définies dans le cahier des charges. L'analyse repose sur une double observation : l'évolution visuelle du système via le **Simulation Dashboard** et l'analyse détaillée des échanges internes via le **CAN Monitor**.

3.2 Méthodologie de test et stratégie de validation

La validation du système ACC a été réalisée à l'aide de **tests par simulation**, méthode couramment utilisée dans le développement des systèmes ADAS avant toute intégration sur véhicule réel. Chaque test consiste à placer le système dans un état initial donné, à appliquer une séquence d'événements (actions conducteur, apparition d'un véhicule cible, variation de vitesse), puis à observer la réaction du système.

Les critères de validation retenus sont :

- le respect des **conditions d'activation et de désactivation** de l'ACC,
- la cohérence des **transitions de la machine à états**,
- la stabilité des commandes moteur et frein,
- le respect des **contraintes physiques** (notamment la limite de décélération),
- la cohérence des **messages CAN échangés** entre les ECU.

Cette méthodologie permet une évaluation complète du système, à la fois du point de vue fonctionnel et du point de vue communication CAN.

3.3 Scénario 1 : Conduite en route libre (Free Drive)

Dans ce premier scénario, aucun véhicule cible n'est présent devant le véhicule ego. Le conducteur active le système ACC après avoir atteint la vitesse minimale requise. Le régulateur passe alors de l'état **ACC STANDBY** à **ACC ACTIVE – Speed Control**.

Les résultats observés montrent que :

- la vitesse du véhicule ego converge vers la vitesse de consigne définie,
- la commande moteur est stable et proportionnelle à l'erreur de vitesse,
- aucune commande de frein n'est générée,
- les trames CAN indiquent l'absence de véhicule cible et un fonctionnement correct du mode Speed Control.

Ce scénario valide le fonctionnement nominal du régulateur adaptatif en absence de contrainte externe.

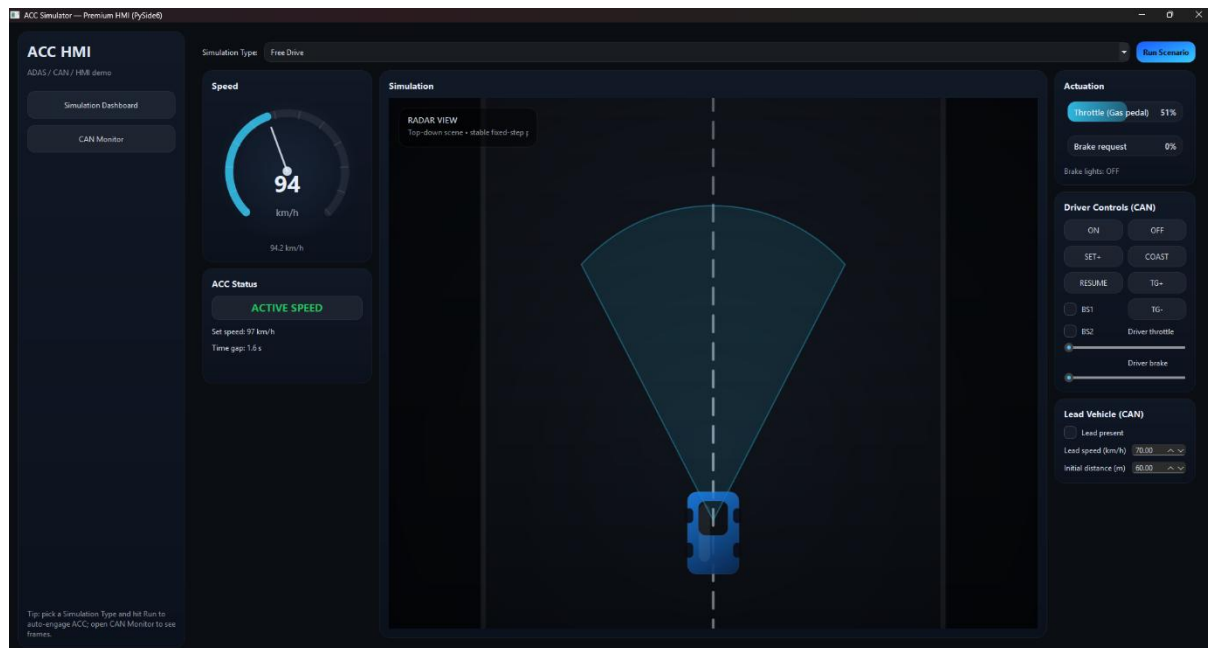


Figure 3.1 - Simulation du régulateur ACC en mode Speed Control (route libre)

ACC Simulator — Premium HMI (PySide6)

ACC HMI
ADAS / CAN / HMI demo

Simulation Dashboard
CAN Monitor

CAN Filter: Filter (e.g. 0x130 or ACC_STATUS or Brake)

ID	Name	Src	Data (hex)	Decoded	Seq
0x140	ENGINE_CMD	EngineECU	0A 00 13 00 00 00 00 00	throttle_pct=10, acc_cmd_mps2=0.19	19530
0x150	BRAKE_CMD	BrakeECU	00 00 00 00 00 00 00 00	brake_pct=0, brake_light=False, decel_cmd_mps2=0.00	19531
0x160	CLUSTER	ClusterECU	C2 03 00 00 3F FC 02 00	speed_kmh=96.20, dist_m=0.00, rel_kmh=-96.10, state=2, brake_light=False	19532
0x102	LEAD_CTRL	LeadControlECU	00 00 00 00 00 00 00 00	sim_type=0, present=False, lead_speed_kmh=0.00, lead_distance_m=70.00	19533
0x101	DRIVER_INPUTS	DriverInputECU	00 00 00 00 00 00 00 78	btn=0, brs=0, thr=0, brk=0, ctr=123	19534
0x110	EGO_STATE	ENV	C2 03 00 00 4D 21 00 00	speed_kmh=96.00, accel_mps2=0.00, pos_m=855.00	19535
0x111	LEAD_STATE	ENV	00 00 00 00 00 00 00 00	present=False, speed_kmh=0.00, pos_m=0.00	19536
0x120	RADAR_OBJECT	RadarECU	00 00 00 F3 FE 00 00 00	present=False, dist_m=0.00, rel_speed_mps=-26.70, lead_speed_kmh=0.00	19537
0x130	ACC_STATUS	ACC	00 00 00 00 00 00 00 00	state=2, msp=0, set_speed_kmh=97.30, time_gap_s=1.60, acc_cmd_mps2=0.19	19538
0x140	ENGINE_CMD	EngineECU	0A 00 13 00 00 00 00 00	throttle_pct=10, acc_cmd_mps2=0.19	19539
0x150	BRAKE_CMD	BrakeECU	00 00 00 00 00 00 00 00	brake_pct=0, brake_light=False, decel_cmd_mps2=0.00	19540
0x160	CLUSTER	ClusterECU	C2 03 00 00 3F FC 02 00	speed_kmh=96.20, dist_m=0.00, rel_kmh=-96.10, state=2, brake_light=False	19541
0x102	LEAD_CTRL	LeadControlECU	00 00 00 00 00 00 00 00	sim_type=0, present=False, lead_speed_kmh=0.00, lead_distance_m=70.00	19542
0x101	DRIVER_INPUTS	DriverInputECU	00 00 00 00 00 00 00 7C	btn=0, brs=0, thr=0, brk=0, ctr=124	19543
0x110	EGO_STATE	ENV	C2 03 00 00 4D 21 00 00	speed_kmh=96.20, accel_mps2=0.00, pos_m=855.70	19544
0x111	LEAD_STATE	ENV	00 00 00 00 00 00 00 00	present=False, speed_kmh=0.00, pos_m=0.00	19545
0x120	RADAR_OBJECT	RadarECU	00 00 00 F3 FE 00 00 00	present=False, dist_m=0.00, rel_speed_mps=-26.70, lead_speed_kmh=0.00	19546
0x130	ACC_STATUS	ACC	02 00 00 00 00 00 00 00	state=2, msp=0, set_speed_kmh=97.30, time_gap_s=1.60, acc_cmd_mps2=0.19	19547
0x140	ENGINE_CMD	EngineECU	0A 00 13 00 00 00 00 00	throttle_pct=10, acc_cmd_mps2=0.19	19548
0x150	BRAKE_CMD	BrakeECU	00 00 00 00 00 00 00 00	brake_pct=0, brake_light=False, decel_cmd_mps2=0.00	19549
0x160	CLUSTER	ClusterECU	C2 03 00 00 3F FC 02 00	speed_kmh=96.20, dist_m=0.00, rel_kmh=-96.10, state=2, brake_light=False	19550

Frame Injection
Message: DRIVER_INPUTS
Arb ID (hex): 0x130
Data (8 bytes hex): 00 00 00 00 00 00 00 00
Send Injected Frame

Tip: pick a Simulation Type and hit Run to auto-engage ACC, open CAN Monitor to see frames.

Figure 3.3 - Trames CAN lors du fonctionnement de l'ACC en mode Speed Control

3.4 Scénario 2 : Suivi d'un véhicule cible (Follow Vehicle)

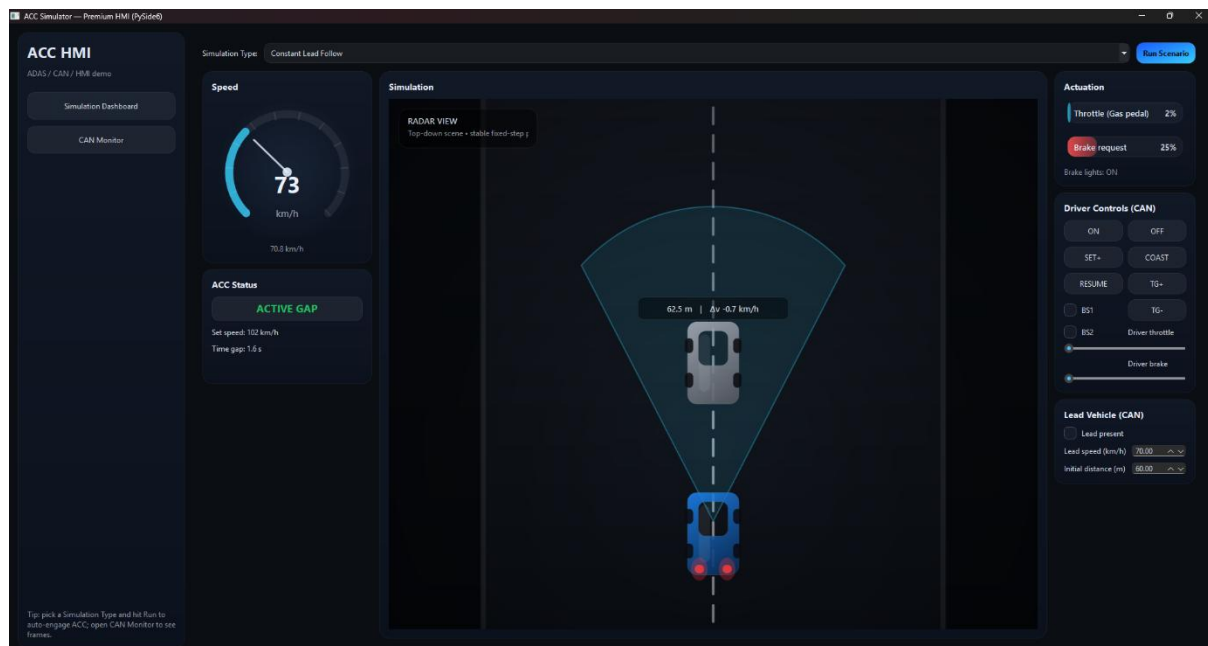


Figure 3.2 - Simulation du régulateur ACC en mode Time Gap Control (suivi de véhicule)

Ce scénario introduit un véhicule cible roulant à une vitesse inférieure à celle du véhicule ego. Dès la détection du véhicule par le radar, le système bascule automatiquement du mode **Speed Control** vers le mode **Time Gap Control**.

L'analyse des résultats montre que :

- la distance inter-véhiculaire converge vers la distance cible calculée à partir du temps de suivi,
- la décélération est progressive et conforme aux limites définies,
- la vitesse du véhicule ego s'adapte à celle du véhicule cible,
- les trames CAN radar et ACC_STATUS confirment la présence du véhicule cible et le changement de mode.

The screenshot shows the 'ACC HMI' simulation interface. On the left, there's a sidebar with 'Simulation Dashboard' and 'CAN Monitor'. The main area displays a list of CAN frames with columns for 'ID', 'Name', 'Src', 'Data (hex)', 'Decoded', and 'Seq'. The frames include engine commands, brake commands, cluster data, lead control data, driver inputs, ego state, lead state, radar object, ACC status, and radar data. The 'Decoded' column shows various parameters like throttle, brake, speed, distance, and status. At the bottom, there's a 'Frame Injection' section with fields for 'Message', 'Arb ID (hex)', and 'Data (8 bytes hex)', and a 'Send Injected Frame' button.

ID	Name	Src	Data (hex)	Decoded	Seq
0x140	ENGINE_CMD	EngineECU	05 00 0A 00 00 00 00 00	throttle_pct=5, acc_cmd_mps2=0.10	35901
0x150	BRAKE_CMD	BrakeECU	00 00 00 00 00 00 00 00	brake_pct=0, brake_light=0, decel_cmd_mps2=0.00	35902
0x160	CLUSTER	ClusterECU	DD 02 1E 02 E9 FF 03 00	speed_kmh=73.30, dist_m=54.20, rel_kmh=-3.20, status=3, brake_light=0	35903
0x102	LEAD_CTRL	LeadControlECU	01 01 BC 02 BC 02 00 00	sim_type=1, present=True, lead_speed_kmh=70.00, lead_distance_m=70.00	35904
0x101	DRIVER_INPUTS	DriverInputECU	00 00 00 00 00 00 00 96	btn=0, bsci=0, thrc=0, bbr=0, cts=150	35905
0x110	EGO_STATE	ENV	DD 02 FA FF B1 08 00 00	speed_kmh=73.30, accel_mps2=-0.06, pos_m=299.30	35906
0x111	LEAD_STATE	ENV	01 00 BC 02 FD 00 00 00	present=True, speed_kmh=70.00, pos_m=358.10	35907
0x120	RADAR_OBJECT	RadarECU	01 1E 02 FF 7F BC 02 00	present=True, dist_m=54.20, rel_speed_mps=-0.90, lead_speed_kmh=70.00	35908
0x130	ACC_STATUS	ACCCECU	03 00 FF 03 10 00 0A 00	status=3, misg=0, set_speed_kmh=102.30, time_gap_s=1.60, acc_cmd_mps2=0.10	35909
0x140	ENGINE_CMD	EngineECU	05 00 0A 00 00 00 00 00	throttle_pct=5, acc_cmd_mps2=0.10	35910
0x150	BRAKE_CMD	BrakeECU	00 00 00 00 00 00 00 00	brake_pct=0, brake_light=0, decel_cmd_mps2=0.00	35911
0x160	CLUSTER	ClusterECU	DD 02 1E 02 E9 FF 03 00	speed_kmh=73.30, dist_m=54.20, rel_kmh=-3.20, status=3, brake_light=0	35912
0x102	LEAD_CTRL	LeadControlECU	01 01 BC 02 BC 02 00 00	sim_type=1, present=True, lead_speed_kmh=70.00, lead_distance_m=70.00	35913
0x101	DRIVER_INPUTS	DriverInputECU	00 00 00 00 00 00 00 97	btn=0, bsci=0, thrc=0, bbr=0, cts=151	35914
0x110	EGO_STATE	ENV	DD 02 FA FF B1 08 00 00	speed_kmh=73.30, accel_mps2=-0.06, pos_m=299.70	35915
0x111	LEAD_STATE	ENV	01 00 BC 02 FD 00 00 00	present=True, speed_kmh=70.00, pos_m=358.40	35916
0x120	RADAR_OBJECT	RadarECU	01 1E 02 FF 7F BC 02 00	present=True, dist_m=54.00, rel_speed_mps=-0.90, lead_speed_kmh=70.00	35917
0x130	ACC_STATUS	ACCCECU	03 00 FF 03 10 00 0A 00	status=3, misg=0, set_speed_kmh=102.30, time_gap_s=1.60, acc_cmd_mps2=0.09	35918
0x140	ENGINE_CMD	EngineECU	05 00 0A 00 00 00 00 00	throttle_pct=5, acc_cmd_mps2=0.09	35919
0x150	BRAKE_CMD	BrakeECU	00 00 00 00 00 00 00 00	brake_pct=0, brake_light=0, decel_cmd_mps2=0.00	35920
0x160	CLUSTER	ClusterECU	DD 02 1E 02 E9 FF 03 00	speed_kmh=73.30, dist_m=54.00, rel_kmh=-3.20, status=3, brake_light=0	35921

Figure 3.4 - Trames CAN lors du suivi d'un véhicule cible en mode Time Gap Control

Ce scénario valide la capacité du système à **adapter automatiquement son comportement** en fonction de l'environnement.

3.5 Scénario 3 : Stop & Go

Le scénario Stop & Go simule une situation de trafic dense dans laquelle le véhicule cible ralentit fortement jusqu'à un arrêt, puis redémarre. Le système ACC doit gérer des phases successives de décélération, de maintien à basse vitesse, puis de ré-accélération. Les résultats montrent que :

- le véhicule ego ralentit de manière fluide et contrôlée,
- la distance de sécurité est maintenue même à basse vitesse,
- la reprise de vitesse est progressive lors du redémarrage du véhicule cible,
- les commandes moteur et frein restent cohérentes et stables.

Ce test met en évidence la **robustesse du système ACC** dans des conditions de conduite complexes et réalistes.

3.6 Scénario 4 : Freinage brusque du véhicule cible

Dans ce scénario critique, le véhicule cible effectue un freinage brusque, imposant une forte décélération au véhicule ego. Le système ACC calcule alors une décélération requise dépassant la limite autorisée de **0,2 g**.

Les observations montrent que :

- le système applique la décélération maximale autorisée,
- l'état **Driver Intervention Required** est activé,
- un message clair est affiché sur l'IHM pour avertir le conducteur,
- les trames CAN indiquent explicitement la limitation de la commande ACC.

Ce scénario valide le respect des **contraintes de sécurité** et la gestion correcte des situations critiques par le système.

3.7 Scénario 5 : Disparition du véhicule cible

Ce scénario simule la disparition soudaine du véhicule cible (changement de voie ou sortie de champ radar). Le système ACC doit alors revenir automatiquement en mode **Speed Control**.

Les résultats observés montrent que :

- la transition entre les modes est fluide et sans instabilité,
- la vitesse du véhicule ego converge à nouveau vers la consigne,
- les messages CAN reflètent correctement la perte de la cible radar.

Ce test confirme la capacité du système à gérer les transitions dynamiques de l'environnement.

3.8 Analyse globale des résultats

L'ensemble des scénarios testés montre que le système ACC :

- respecte les conditions d'activation et d'annulation définies,
- applique correctement les lois de régulation,

- garantit un comportement stable et prévisible,
- assure une communication CAN cohérente entre les différents ECU.

La visualisation simultanée via le Dashboard et le CAN Monitor a permis de vérifier la **correspondance entre le comportement observé et les données échangées**, renforçant la crédibilité des résultats.

3.9 Limites observées et perspectives d'amélioration

Bien que les résultats soient satisfaisants, certaines limites ont été identifiées :

- le radar est modélisé sans bruit ni latence,
- un seul véhicule cible est pris en compte,
- le modèle dynamique reste volontairement simplifié.

Ces limitations sont acceptables dans un cadre pédagogique, mais pourraient être levées dans une version future en introduisant des perturbations, des multi-cibles ou des modèles physiques plus avancés.

3.10 Conclusion

Ce chapitre a permis de valider expérimentalement le **bon fonctionnement du système de régulation de vitesse adaptatif** développé. Les tests réalisés démontrent que le système répond de manière cohérente et conforme aux exigences du cahier des charges, aussi bien dans des situations normales que dans des cas critiques. Les résultats obtenus confirment la pertinence de l'architecture choisie et de l'approche par simulation, ouvrant la voie à des extensions futures du système.

Conclusion générale

Ce projet a permis de concevoir et de mettre en œuvre une **application de simulation du Régulateur de Vitesse Adaptatif (ACC)** reposant sur une architecture logicielle inspirée des systèmes automobiles réels. L'utilisation d'une approche modulaire basée sur des **ECU logiciels** et un **bus CAN virtuel** a permis de reproduire de manière cohérente les interactions entre les différents sous-systèmes du véhicule.

Les résultats obtenus à travers les différents scénarios de simulation démontrent que le système développé respecte les exigences définies dans le cahier des charges, aussi bien en termes de logique de contrôle que de contraintes de sécurité. L'analyse conjointe du **Simulation Dashboard** et du **CAN Monitor** a permis de valider la cohérence entre le comportement observé et les messages CAN échangés, renforçant ainsi la crédibilité de la solution proposée.

Au-delà de la validation fonctionnelle, ce projet constitue un **outil pédagogique pertinent** pour la compréhension des systèmes ADAS et des réseaux de communication embarqués. Des perspectives d'amélioration peuvent être envisagées, telles que l'intégration de modèles capteurs plus réalistes, la gestion de plusieurs véhicules cibles ou l'extension vers d'autres fonctions d'assistance à la conduite. Ainsi, ce travail constitue une base solide pour des développements futurs et une introduction concrète aux problématiques des systèmes embarqués automobiles modernes.