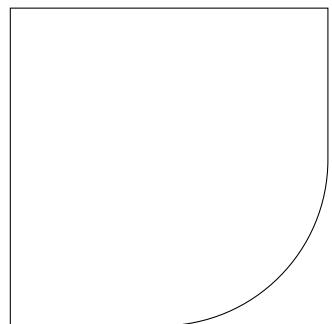


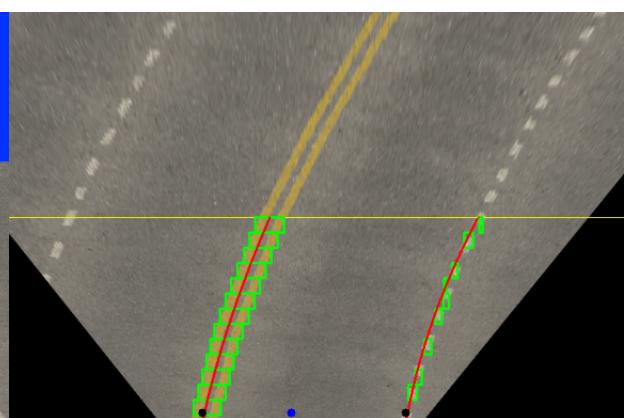
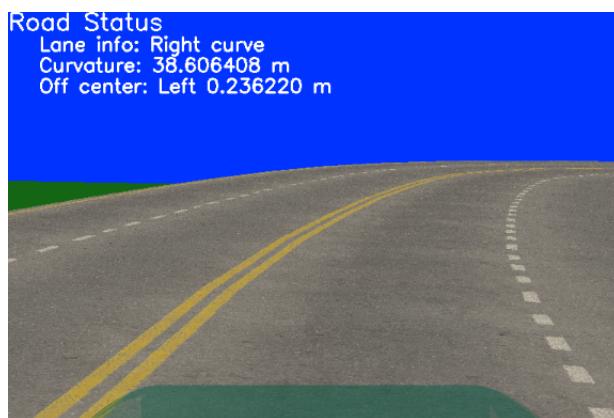
ENSTA Bretagne
2, rue François Verny
29806 BREST cedex
FRANCE
Tel +33 (0)2 98 34 88
00
[www.ensta-bretagne.
fr](http://www.ensta-bretagne.fr)



Middleware
UE 4.1
April 24, 2020



Véhicule autonome suiveur de lignes



Colin Baumgard, Ludovic Diguet, Hamid Hacene,
Corentin Lemoine, Antonin Lizé

Contents

Introduction	2
Abstract	2
1 Présentation du projet	3
1.1 Problématique	3
1.2 Objectifs	3
2 Support matériel	4
2.1 Présentation de la voiture	4
2.2 Adaptation de l'architecture pour le projet	4
3 Simulation sous V-REP	5
3.1 Modélisation dynamique de la voiture	5
3.2 Création de la piste	5
3.3 Interface avec ROS	5
4 Choix techniques	6
4.1 Architecture fonctionnelle	6
5 Pipeline de contrôle	6
5.1 Architecture <i>C2</i>	6
5.2 Window Lane Detection	6
6 Résultats	6
7 Perspectives et améliorations	7
Conclusion	7
Annexe1	7
Annexe2	7

Introduction

Ce projet s'inscrit dans le cadre des enseignements de l'ENSTA BRETAGNE dispensés aux élèves de deuxième année de la filière *Robotique Autonome* au sein de l'U.E 4.1 : *Middleware*.

L'objectif est d'utiliser des connaissances acquises au long de l'année pour réaliser un projet robotique concret. Ce projet permet de toucher à de nombreux domaines : CAO et conception mécanique, électronique, OS, réseau, conception logiciel, OpenCV... De plus, travailler sur un même code en groupe n'est pas évident, il est donc important de s'y former.

Abstract

In this article presents the realisation of an autonaumous 1:10 scale car. We will particularly talk about the software implementation. We have achied our goal using a rpi board and a camera and OpenCV to process images. ROS is used to have a modular code.

1 Présentation du projet

1.1 Problématique

Nous avons pu acquérir au fil du semestre quatre une connaissance théorique et pratique du *Middleware ROS* (Robot Operating System). Pour rappel, un *Middleware* est un outil qui permet d'échanger des informations entre différentes applications et programmes via des techniques et des messages standardisés. ROS est un *Middleware* développé par l'**OPEN ROBOTICS**.

Le but de ce projet au sein de l'U.E est de maîtriser les concepts fondamentaux de ROS et d'arriver à les implémenter sur un robot. Il s'agit d'appliquer une démarche d'ingénierie afin de proposer des solutions adéquates en prenant en compte les différentes dimensions techniques du problème : *hardware* et *software*.

La problématique de départ se résume ainsi à "rendre un véhicule autonome pour accomplir un tour complet du terrain d'athlétisme de l'école". Les véhicules mis à disposition sont des voitures à l'échelle 1/10^{ème} construite par différents groupes au semestre précédent.



(a) Terrain de l'*ENSTA Bretagne*

(b) Voiture utilisée pour le projet

Figure 1: Problématique du projet

1.2 Objectifs

L'objectif principal de notre équipe est de répondre à la problématique tout en proposant une approche différente des autres groupes pour apporter une diversité aux contenus des projets.

Ainsi, nous nous sommes fixé ces deux objectifs supplémentaires :

- Proposer une approche réaliste qui s'inspire des voitures autonomes existantes sur le marché ou en développement ;

- Développer essentiellement en C++ pour garantir la portabilité des algorithmes et l'implémentation sur d'autres structures ;
- Utiliser le plus possible les solutions et algorithmes proposés par ROS afin d'exploiter au maximum les possibilités offertes par ce *Middleware*.

Compte tenu de l'évolution de la pandémie du *Covid-19*, et suite à la fermeture de l'*ENSTA Bretagne*, nous avons revu la problématique et les objectifs du projets. À cause d'une avarie de matériel, l'objectif principal ne peut être atteint : il s'agit désormais de construire une simulation pour valider les différents algorithmes qui seront proposés.

2 Support matériel

2.1 Présentation de la voiture

Sed sed porttitor risus. Curabitur elit erat, aliquet non orci et, lacinia egestas libero. Suspendisse tristique diam eu efficitur tincidunt. Praesent rhoncus venenatis sapien, nec efficitur elit mattis et. Duis nec felis convallis nulla semper gravida a sit amet magna. Ut enim sapien, vehicula vel ultricies quis, pharetra vitae nunc. In hac habitasse platea dictumst.

-> Expliquer vite fait la construction de la voiture + photo + schéma

2.2 Adaptation de l'architecture pour le projet

Sed sed porttitor risus. Curabitur elit erat, aliquet non orci et, lacinia egestas libero. Suspendisse tristique diam eu efficitur tincidunt. Praesent rhoncus venenatis sapien, nec efficitur elit mattis et. Duis nec felis convallis nulla semper gravida a sit amet magna. Ut enim sapien, vehicula vel ultricies quis, pharetra vitae nunc. In hac habitasse platea dictumst.

-> Citer les sensors qui équiperont la voiture !

-> Expliquer tous les changements apportés à la voiture de base : pièces supplémentaires (support caméra, support Rasp, batterie, support GPS, etc...) avec schéma et pk ces choix techniques de conception !

3 Simulation sous V-REP

3.1 Modélisation dynamique de la voiture

Sed sed porttitor risus. Curabitur elit erat, aliquet non orci et, lacinia egestas libero. Suspendisse tristique diam eu efficitur tincidunt. Praesent rhoncus venenatis sapien, nec efficitur elit mattis et. Duis nec felis convallis nulla semper gravida a sit amet magna. Ut enim sapien, vehicula vel ultricies quis, pharetra vitae nunc. In hac habitasse platea dictumst.

-> Expliquer vite fait la modélisation de la voiture en faisant référence au tp de Benn Zerr.

3.2 Création de la piste

Sed sed porttitor risus. Curabitur elit erat, aliquet non orci et, lacinia egestas libero. Suspendisse tristique diam eu efficitur tincidunt. Praesent rhoncus venenatis sapien, nec efficitur elit mattis et. Duis nec felis convallis nulla semper gravida a sit amet magna. Ut enim sapien, vehicula vel ultricies quis, pharetra vitae nunc. In hac habitasse platea dictumst.

-> Expliquer comment la piste a été construite et surtout justifier le choix de la texture (vraie route !)

3.3 Interface avec ROS

Sed sed porttitor risus. Curabitur elit erat, aliquet non orci et, lacinia egestas libero. Suspendisse tristique diam eu efficitur tincidunt. Praesent rhoncus venenatis sapien, nec efficitur elit mattis et. Duis nec felis convallis nulla semper gravida a sit amet magna. Ut enim sapien, vehicula vel ultricies quis, pharetra vitae nunc. In hac habitasse platea dictumst.

-> Mettre les équation d'états de la voiture et expliquer quels sont les variables à contrôler

-> Expliquer l'interface avec ROS : lua pour la caméra et le control de steering + speed

-> Maintenant comment rendre cette voiture autonome ?!

4 Choix techniques

4.1 Architecture fonctionnelle

Sed sed porttitor risus. Curabitur elit erat, aliquet non orci et, lacinia egestas libero. Suspendisse tristique diam eu efficitur tincidunt. Praesent rhoncus venenatis sapien, nec efficitur elit mattis et. Duis nec felis convallis nulla semper gravida a sit amet magna. Ut enim sapien, vehicula vel ultricies quis, pharetra vitae nunc. In hac habitasse platea dictumst.

-> Mettre l'arch fct.

-> Expliquer les différentes briques : GPS, PWM, etc..

5 Pipeline de contrôle

5.1 Architecture C2

Sed sed porttitor risus. Curabitur elit erat, aliquet non orci et, lacinia egestas libero. Suspendisse tristique diam eu efficitur tincidunt. Praesent rhoncus venenatis sapien, nec efficitur elit mattis et. Duis nec felis convallis nulla semper gravida a sit amet magna. Ut enim sapien, vehicula vel ultricies quis, pharetra vitae nunc. In hac habitasse platea dictumst.

-> Mettre l'arch C2 (node graphe ??) et expliquer les grandes étapes.

5.2 Window Lane Detection

Sed sed porttitor risus. Curabitur elit erat, aliquet non orci et, lacinia egestas libero. Suspendisse tristique diam eu efficitur tincidunt. Praesent rhoncus venenatis sapien, nec efficitur elit mattis et. Duis nec felis convallis nulla semper gravida a sit amet magna. Ut enim sapien, vehicula vel ultricies quis, pharetra vitae nunc. In hac habitasse platea dictumst.

-> Expliquer toute la pipeline de traitement d'image en détaillant les fonctions et montrer les résultats intermédiaires !

6 Résultats

Sed sed porttitor risus. Curabitur elit erat, aliquet non orci et, lacinia egestas libero. Suspendisse tristique diam eu efficitur tincidunt. Praesent rhoncus venenatis sapien, nec efficitur elit mattis et. Duis nec felis convallis nulla semper

gravida a sit amet magna. Ut enim sapien, vehicula vel ultricies quis, pharetra vitae nunc. In hac habitasse platea dictumst.

-> Mettre les résultats obtenues + lien vers vidéo !

7 Perspectives et améliorations

-> Mettre toutes les améliorations possibles (Docker)

Conclusion

Lorem ipsum dolor sit amet coming Soon !!

Annexe1 : Raspberry comme ordinateur de bord

Nous avons choisi d'installer Raspbian sur notre raspberry, cela nous permet d'avoir un accès simple à la caméra, cependant l'installation de ROS est un peu plus longue, car il faut recompiler les sources.

Pour installer ROS sur notre raspberry pi 3 nous avons suivi la documentation officielle : <http://wiki.ros.org/ROSberryPi/Installing%20ROS%20Kinetic%20on%20the%20Raspberry%20Pi>

Annexe2 : Installation des librairies

Lorem ipsum dolor sit amet coming Soon !!