

# Compte-rendu de réunion numéro 6

Date : Mercredi 8 mars 2017

<b>Objectif(s)</b>	Réunion de planification du sprint et des livrables
<b>Participants</b>	Michaël LAUER, Michel TAIX, Marine BOUCHET, Bruno DATO, Thibaut AGHNA-TIOS, Tristan KLEMPKA, Thibault LAGOUTE
<b>Auteur</b>	Marine Bouchet

## Sujets abordés

Nous avons discuté de l'avancement du projet et de ce qu'il était possible de réaliser dans le temps restant pour les différentes briques :

- Détection
- Localisation
- Trajectoire
- Commande et Odométrie
- Commande haut-niveau

Pour finir, nous avons évoqué la première étape du projet, l'oral et le stand.

## Existant et Tâches à réaliser

Nous sommes convenus des différentes parties à concevoir pour chaque brique.

### Localisation et Détection

La détection des amers se fait via la bibliothèque `AR_TRACK_ALVAR`. Elle permet facilement de retourner l'id et la position, dans le repère caméra du repère lié à l'objet (param extrinsèques). Il est important de notifier que cette bibliothèque permet aussi de lire plusieurs `ar_markers` sur une même image – cette fonctionnalité est à implémenter.

Après cette étape, nous déterminons l'emplacement de notre robot dans le repère local du marker reconnu afin de connaître sa position dans le repère global de la map. Cette étape marche actuellement avec un seul marker et est en cours de généralisation.

Plusieurs décisions et observations ont été faites :

- Une série de tests sur les limites de la bibliothèque détection des `ar_marker` (distance min, max et angle de visibilité maximum) ainsi que la confiance en la position relative retournée pour un marker détecté.
- Réfléchir à la possibilité d'un filtre de Kalman qui faciliterait le processus d'intégration des observations visuelles – et, en utilisant la brique déjà faite `navigation_stack`. Il s'agira notamment de définir le vecteur d'états cachés :  $(x, y, \theta, v, dv, ddv)$  du robot. Faire des observations à l'arrêt – `STATIC LOOK`, pour rendre le filtre linéaire, en perdant les paramètres de vitesses, serait plus facile. Pour la prédiction de la position, quand on ne voit pas d'amers, nous éviterons de faire des estimations trop souvent pour éviter d'accumuler trop d'erreurs : *periode* > 100ms par exemple (adapté à la vitesse du robot).
- Une carte de visibilité des amers devra être produite afin de savoir, pour chaque position s'il est possibles de voir un ou des amers – avec les paramètres du premier point.
- On positionnera les amers à la même hauteur que la caméra pour passer en deux dimensions.

### Commande Haut Niveau, Trajectoire et Commande/Odométrie

Le robot se déplace actuellement dans une carte avec des obstacles connus. Il suit une trajectoire pour atteindre un but prédéfini dans la brique `navigation_stack` avec les paramètres adéquats dans la

limite de l'odométrie et des roues. Des tests ont été réalisés pour l'évitement d'obstacles et les passages entre les portes.

Plusieurs décisions et observations ont été faites :

- Pour l'odométrie, l'erreur systématique de chacune des roues est à définir en testant en ligne droite (grande pour réduire l'impact de l'erreur du démarrage – nous ne contrôlons l'accélération) et en cerle. L'erreur quand le robot tourne devra être gérée en définissant la vitesse la plus adéquate, ni trop grande ni trop petite, pour éviter à la fois les sauts et le patinage. Un rapport devra être fourni.
- La brique navigation.stack ne permet pas de gérer les obstacles non connues, dynamique. Il va falloir, dès que le cas se présente, reprendre la main, et utiliser une brique de navigation locale que nous devrons coder pour éviter cet obstacle – simplement dans un premier temps (exemple, tourner à droite pour le contourner) et un algorithme de "table top" dans un deuxième temps.
- la navigation se fera d'un point vers un amér – dans le champs de vision de l'amér en question plus précisément. S'il ne le voit pas, il tourne sur lui même et s'arrête pour récupérer la position relative de celui-ci. Si deux amér peuvent être détectés, il en détecte un, se tourne et détecte le deuxième.
- A l'initialisation nous admettrons que la position du robot sera soit connue, soit dans l'angle de vue d'un amér.

## Autres

### Première étape

Pour le tp de "recherche balle", il faudra essayer d'effectuer la recherche en boucle fermée et le rendre le plus robuste possible.

### Gestion de projet et Présentation

Le bilan de projet devra être court, détaillé et contenir une vidéo. Un manuel d'utilisation sera fourni et le code commenté. Un oral blanc sera réalisé et les diapositives envoyées avant la présentation. Le début de la dernière semaine sera consacrées à ses tâches ainsi qu'à la préparation du stand pour la journée EEA.

**Prochaine réunion : Mardi 21 mars 2017 entre 10h et 12h**

**Oral : Mercredi 22 mars 2017 après-midi**

**Stand : Jeudi 23 mars 2017 après-midi**