

30/06/2021

BETAÏLE

Antonin

Compte-rendu Hebdomadaire Bras

Manipulateur

Poursuite du projet

TABLE DES MATIERES

EXPLICATIONS :	2
DEPLOIEMENT DU NIRYO ONE :	2
- ASSEMBLAGE ET MATERIEL	3
<i>Intégration de la caméra :</i>	3
<i>Design d'un nouveau préhenseur :</i>	3
- SYSTEME EMBARQUE	3
<i>Alimentation du bras :</i>	3
<i>Création d'un nouvel étage pour le Turtlebot :</i>	3
- GESTION DU PROBLEME D'URDF	4
- CORRECTION DE POSITIONNEMENT	4
<i>Correction verticale :</i>	4
DEPLOIEMENT DU TURTLEBOT :	5
- INTERACTIONS UTILISATEUR	5
<i>Interface vocale :</i>	5
<i>La commande vocale :</i>	5
- LE DOCKING	5
<i>Le pupitre de docking :</i>	5
<i>Relance après docking :</i>	6
DEPLOIEMENT DU MEUBLE INTELLIGENT :	6
- INTEGRATION ROS	6
<i>Commande du meuble intelligent :</i>	6
- CAPTEURS SUPPLEMENTAIRES	7
<i>Ajout d'une caméra USB :</i>	7

Explications :

Mon apport au projet « Bras manipulateur pour une personne en grande perte d'autonomie » n'a pas permis de poursuivre toutes les pistes explorées par le précédent stagiaire ni toutes celles que j'ai moi-même pu mettre en avant. C'est pourquoi je vais tenter de répertorier ici tous les objectifs qui n'ont pas été complétés ni même abordés lors de mon stage mais qui mériteraient un suivi plus poussé

Déploiement du Niryo One :

- Assemblage et matériel

Intégration de la caméra :

Les choix de l'entreprise Niryo dans le placement de la caméra et de sa connectique ont été une source de problème matériel. Les mouvements divers du bras ont abîmé le câble USB sur le long terme, rendant impossible l'utilisation de la caméra. Après changement du câble, le positionnement de ce dernier ne semble plus pouvoir l'endommager. Cependant, l'utilisation de scotch uniquement a été requise pour arriver à ce résultat. L'idéal serait de repenser la façon dont il serait possible d'intégrer la caméra ou de protéger le câble

Design d'un nouveau préhenseur :

Les préhenseurs du Niryo sont suffisamment efficaces pour déplacer les petits objets de couleur fourni avec les accessoires. Cependant, puisque le scénario d'utilisation du système prévoit de faire transporter au bras des objets du quotidien tels que des tasses remplies de boisson, il est essentiel de pouvoir assurer la préhension des objectifs. Créer un nouvel outil plus performant pourrait s'avérer être une étape incontournable pour assurer le bon déroulé d'une action.

- Système embarqué

Alimentation du bras :

Le bras fonctionnant toujours sur secteur et aucune recherche n'ayant été menée pour en savoir plus sur la façon dont je pouvais alimenter le bras sur batterie, tout le travail d'intégration du bras n'a pas été réalisé. Il faut choisir une batterie et voir comment intégrer le Niryo à la base mobile avec les contraintes d'autonomie et de place qui concernent le système.

Création d'un nouvel étage pour le Turtlebot :

Puisque le bras est toujours contraint par la présence de son câble d'alimentation je n'avais pas prévu de le faire porter par le turtlebot. Cependant, l'avancement des fonctionnalités du système et des interactions entre le bras et la base mobile a rendu possible le fonctionnement du système dans un rayon restreint défini par la longueur du câble du Niryo (en théorie). Il serait donc possible de fixer le bras sur un autre étage du turtlebot pour tester le fonctionnement du système en conditions réelles (et ainsi finaliser le premier prototype).

- Gestion du problème d'URDF

Mettre un terme aux messages d'erreur :

Plusieurs fois abordée, la question des messages d'erreur sur la fenêtre du bringup du Niryo n'a pas été résolue. Il serait utile de pouvoir remédier à ce problème.

- Correction de positionnement

Correction verticale :

Il est en principe possible que le workspace ne se trouve pas à la hauteur attendue par rapport à la base du Niryo. Dans ce cas de figure l'offset renseigné au robot ne permettrait pas au bras de rapprocher une approche convenable de l'objet. Dans le cas où le workspace serait trop haut, le bras finirait même par forcer le mouvement alors que la pince aurait déjà touché le workspace. Cette situation pourrait endommager le système, et renverser l'objet, ce qui est à proscrire.

Pour estimer l'offset il faudrait en principe estimer la longueur du rayon d'un marqueur pour le comparer avec une longueur obtenue avec une photo témoin. La différence de longueur permettrait de savoir si la caméra est au-dessus ou en dessous de la distance d'observation. Et par une simple conversion nous pourrions alors en déduire l'offset idéal.

Déploiement du Turtlebot :

- Interactions utilisateur

Interface vocale :

Actuellement la synthèse vocale n'est activée que si l'utilisateur clique sur un bouton de l'interface web. Il est donc possible d'entendre quand le robot effectue une action mais pas quand il la termine. Pour l'instant des messages d'information sont juste affichés en fonction des terminaisons des actions. Il serait judicieux d'être en mesure de rendre écoutable ces messages pour que la fin des actions soit mieux identifiée

La commande vocale :

Piste non explorée

- Le docking

Le pupitre de docking :

L'idée avait été abordée pendant une réunion et jugée comme convenable.

Le principe était de créer un pupitre à hauteur variable sur lequel seraient placés le marqueur alvar et le workspace du Niryo. Le marqueur serait positionné au pied, à hauteur de la caméra du turtlebot, et le workspace serait sur une tête orientée à l'horizontale.

Le but de ce pupitre serait de donner plus de flexibilité au placement du workspace pour palier au manque de précision de la navigation du turtlebot. Puisqu'avec le node de navigation, il est trop dur de positionner le robot à une coordonnée exacte avec la bonne orientation, seul le docking permet d'arriver à un résultat satisfaisant.

Le docking passant par le node `/automatic_parking_vision` nécessite uniquement la présence d'un marqueur bien visible. Le positionnement du turtlebot n'a donc plus besoin d'être précis puisque le docking peut avoir lieu tant que le robot peut repérer le marqueur en faisant une rotation sur lui-même.

C'est par ce biais que le pupitre offre de la flexibilité au positionnement et à l'utilisateur. En effet, le pupitre peut être déplacé sans que cela implique de revoir les coordonnées des objectifs et le placement du workspace. Puisque le workspace est positionné juste au dessus

du marqueur, le Niryo n'aura pas de problèmes pour réaliser des actions impliquant la vision, si le docking est bien réalisé. Et puisque la hauteur du pupitre est modulable, l'ensemble peut être adjoint à n'importe quel meuble de l'utilisateur.

Ainsi il serait possible d'avoir plusieurs pupitres pour chaque point d'intérêt représenté par un bouton au niveau de l'interface web.

Relance après docking :

Dans le CR n°9 j'indiquais avoir permis de relancer la navigation après le docking. Cependant, l'initialisation de la navigation :

```
/home/chaire/catkin_ws/src/turtlebot3/turtlebot3_navigation/launch/amcl.launch
```

définit une position et une orientation de départ unique.

Ainsi, il faudrait trouver un moyen de sauvegarder la dernière commande de navigation pour la mettre en argument d'une fonction qui lancerait d'autres versions du fichier amcl.launch avec d'autres coordonnées.

Déploiement du meuble intelligent :

- Intégration ROS

Commande du meuble intelligent :

Lors de son stage, Quentin Mérian a pu développer le contrôle des moteurs du tiroir et de l'élévateur du meuble. De mon côté j'ai également pu implémenter les boutons de commande du meuble dans l'interface web avec un menu dédié. Cependant nos travaux n'ont pas pu être mis en commun par manque du temps. Il reste donc à créer un node de contrôle du meuble permettant de prendre en charge les interactions avec l'interface web de la même façon que les nodes `/niryo_control` et `/simple_navigation_goal`. Il disposerait donc d'un switch avec un « case » pour chaque bouton et a minima un subscriber pour recevoir les informations des boutons et un publisher pour envoyer les informations utiles à l'utilisateur.

Chaque « case » du switch doit pouvoir activer les actions du meuble soit en lançant un roslaunch pour chaque node soit en passant par un topic spécifique qui communiquerait avec

un node regroupant toutes les actions. Tout dépend de la façon dont les actions seront implémentées.

- Capteurs supplémentaires

Ajout d'une caméra USB :

Tout comme pour le niryo et le turtlebot, il serait judicieux de pouvoir intégrer une caméra au meuble pour lui permettre de surveiller le bon déroulement des actions en cours. Le flux vidéo serait retransmis sur l'interface web, au niveau du menu du meuble en ajoutant quelques lignes dans le fichier **webui.js** et dans le fichier **index.html**.