



PROJET
I14

IMITATION D'UN HUMAIN PAR UN ROBOT
HUMANOÏDE GRÂCE À UN SYSTÈME DE CAPTURE
DE MOUVEMENT

Arthur Le Monnier et Thomas Caminati
Tuteur : Mai Nguyen

Résumé

Dans le cadre de la 3^{ème} année du cursus FIP à Telecom Bretagne, et plus particulièrement de l'UV MGP320, nous avons réalisé un projet de développement technique lié à nos domaines de compétences. Ce rapport présente les aspects organisationnels et techniques de notre démarche sur le sujet suivant : Imitation d'un humain par un robot humanoïde grâce à un système de capture de mouvement.

Nous présentons les outils techniques utilisés :

- Plateforme de capture de mouvement Xsens
- Simulateur de robot Vrep
- Script développé pour faire communiquer et animer le robot à partir des captures de mouvement

Nous présentons également les outils, la logique de gestion de projet mise en place pour réaliser ce développement ainsi que les aléas rencontrés lors de cette période.

Malgré la non-finalisation du développement à l'heure de la rédaction de ce rapport, notre travail apporte des éléments de réponse à la problématique.

Sommaire

Résumé.....	1
Sommaire	2
Contexte de l'étude.....	3
Cahier des charges fonctionnel	4
Démarche utilisée.....	5
Lot 1 : Apprentissage.....	7
Tâche 3 : Prendre connaissances du contexte et de l'environnement du projet	7
Tâche 4 : Prise en main des logiciels à utiliser et des langages de programmation	7
V-REP	7
MVN Studio / Xsens.....	8
Lot 2 : Recherche et conception d'une solution	11
Tâche 5 : Recherche les moyens de communication entre les capteurs et le robot	11
Tâche 6 : Etude de la solution la plus viable.....	11
Lot 3 : Développement de la solution	12
Tâche 7 : Développement de la solution.....	12
Répartition réelle des tâches.....	13
Identification des rôles de chaque membre du groupe	14
Bilan critique du travail et des résultats.....	14
Table des figure	16
Bibliographie.....	16
Annexes	17
Analyse de la différence entre les deux diagrammes de Gantt	17

Contexte de l'étude

Le sujet de notre projet est « I14 : Imitation d'un humain par un robot humanoïde grâce à un système de capture de mouvement » et est encadré par Mai Nguyen. Notre travail s'inscrit dans le cadre d'un projet plus grand qui est de proposer à des cabinets de kinésithérapeutes une solution pour accompagner les patients dans la pratique d'exercices de kinésithérapie à domicile qu'ils soient réalisés par le robot sur le patient ou bien que le robot puisse contrôler la bonne réalisation des exercices par le patient et le corriger.

A Télécom Bretagne, une solution a déjà été proposée avec comme moyen de capter les déplacements une caméra Kinect. Cependant comme il s'agit d'un dispositif grand public et non de capteurs de mouvements professionnels, le traitement est insuffisamment précis.

Pour pallier à ce manque de précision, l'utilisation de vrais capteurs de mouvements a été proposée. C'est ainsi que notre projet fut présenté, nous allons devoir utiliser un dispositif de capteurs développé par la société Xsens pour enregistrer les exercices des patients. Enfin, la solution développée devra les traiter et les faire reproduire par le robot.

Cahier des charges fonctionnel

Intitulé du projet : Imitation d'un humain par un robot humanoïde grâce à un système de capture de mouvement

Objectifs :

- Enregistrer les mouvements avec des capteurs de mouvements
- Traiter les données apportées par ces capteurs
- Traduire ces données pour que le robot reproduise les mouvements

Contraintes :

- Délai pour le projet 3 semaines
- Utilisation des capteurs de mouvements Xsens
- Utilisation du simulateur V-Rep pour simuler le robot Poppy
- Utilisation du robot Poppy physique si la solution est un succès sur le simulateur
- Temps entre le moment où le client fait ces exercices et le moment où le robot les reproduit inférieur à 2 secondes au maximum (< 1sec de préférence)
- Le langage à utiliser n'est pas imposé mais l'utilisation de Python est préférée.
- Seul le haut du corps jusqu'au bassin sera à simuler

Livrables à rendre :

- Rapport sur le projet (max. 15 pages) pour le 8 février
- Diaporama de présentation du projet pour le 10 février
- Documentation de la solution pour le 12 février

Démarche utilisée

Pour mettre en œuvre notre solution, nous avons créé différentes tâches. Nous avons aucune connaissance des matériels utilisés donc il fallait commencer par une première étape indispensable de prise en main des logiciels et technologies nécessaires à notre projet. Ensuite, il a fallu chercher comment faire le lien entre les capteurs Xsens et le robot Poppy en regardant notamment les différentes sorties possibles via le logiciel de Xsens. Après avoir étudié les solutions possibles, nous avons dû en choisir une et la mettre en œuvre puis tester si cette solution correspondait aux contraintes du cahier des charges.

Nous obtenons donc la démarche suivante :

N° Lot ou tâche	Intitulé
Lot 0	Gestion de Projet
Tâche 1	Gestion des délais
Tâche 2	Documentation
Lot 1	Apprentissage
Tâche 3	Prendre connaissance du contexte et de l'environnement du projet
Tâche 4	Apprentissage des logiciels à utiliser et des langages de programmation
Lot 2	Recherche et conception d'une solution
Tâche 5	Recherche les moyens de communication entre les capteurs et le robot
Tâche 6	Étude de la solution la plus viable
Lot 3	Développement de la solution
Tâche 7	Développement de la solution

Figure 1 : Tableau des lots et tâches effectuées

Dans notre planification, nous avons aussi dû prendre en compte la rédaction de la documentation et nous avons donc la planification suivante :

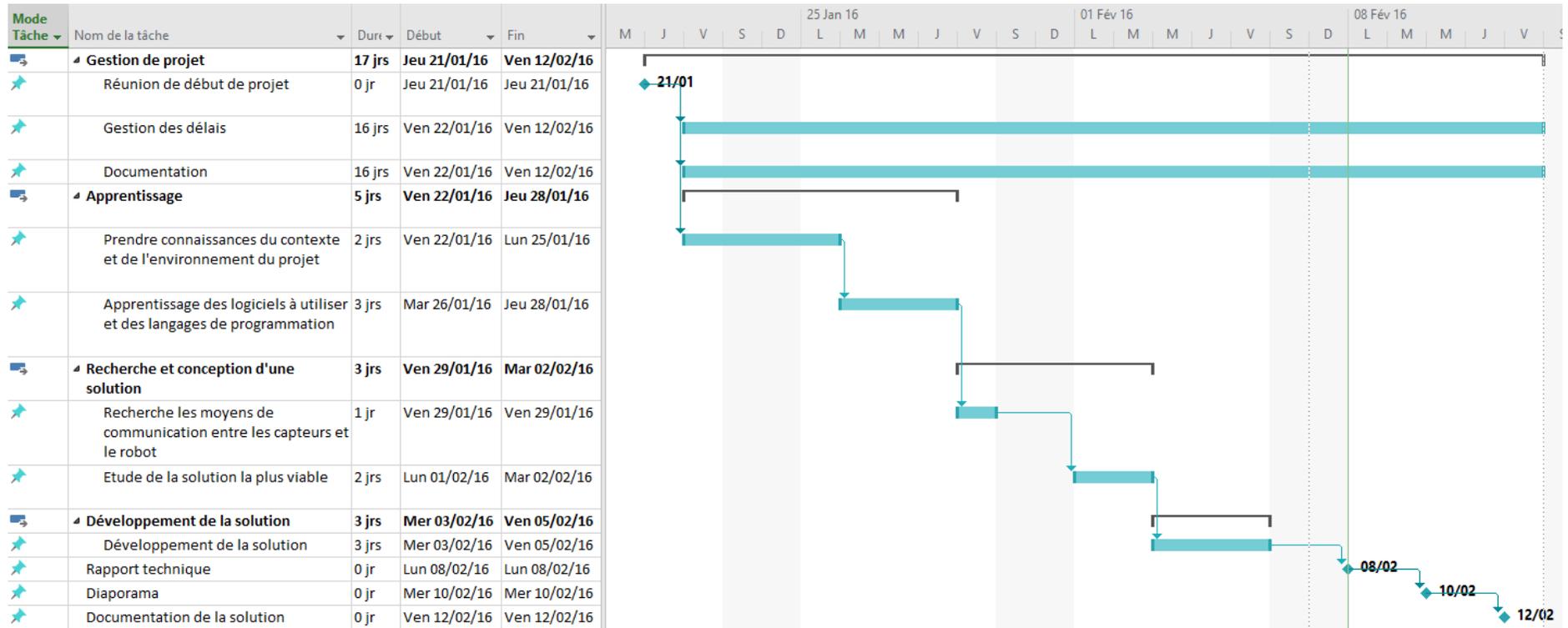


Figure 2 : Diagramme de Gantt initial de notre projet

Lot 1 : Apprentissage

Tâche 3 : Prendre connaissance du contexte et de l'environnement du projet

La première partie de ce lot s'est déroulée sans difficulté, nous avons étudié la documentation liée à l'utilisation de V-Rep et des capteurs Xsens.

Pour le logiciel V-Rep, il s'agissait principalement d'un tutoriel qui nous montrait comment le code Python se répercutait sur la simulation de Poppy.

Quant à Xsens, il y avait plus de documentation et nous avons principalement regardé comment disposer les capteurs sur une personne et comment calibrer et utiliser le logiciel MVN qui est lié aux capteurs Xsens.

Tâche 4 : Prise en main des logiciels à utiliser et des langages de programmation

Cette tâche s'est révélée plus longue que prévu puisque nous ne pouvions pas installer le logiciel MVN sur l'ordinateur que l'on nous avait fourni pour des raisons d'incompatibilité au niveau du système d'exploitation.

Nous avons donc reçu l'autorisation de l'installer sur nos machines personnelles puisque la clé servant à activer le logiciel est de type dongle, c'est-à-dire que le logiciel ne peut fonctionner que si la clé est branchée à l'ordinateur qui souhaite le lancer. Cependant, un autre problème est apparu puisque la clé dongle n'était pas activée et que nous n'avions pas la clé produit nécessaire pour son activation. Suite à quelques jours nous avons reçu cette clé et avons pu poursuivre cette tâche comme il se devait.

Comme nous savions que nous allions perdre du temps sur cette tâche à la suite de la deuxième difficulté, nous avons décidé de commencer en parallèle le Lot 2 et plus particulièrement la tâche 5 puisque nous pouvions utiliser la documentation de Xsens pour trouver des pistes de solutions.

Voici la présentation des deux logiciels que nous avons utilisés :

- **V-REP**

V-Rep est un logiciel très utilisé pour simuler les mouvements de tous les types de robot, y compris le robot open source Poppy que nous utilisons. Les contrôles de mouvement peuvent être programmés en C/C++, Python, Java, Lua, Matlab, Octave or Urbi. Nous avons choisi Python comme conseillé dans le cahier des charges pour sa simplicité d'écriture. En lançant le code contrôlant Poppy, on peut visualiser le résultat dans une fenêtre de simulation comme on peut le voir dans la figure ci-dessous.

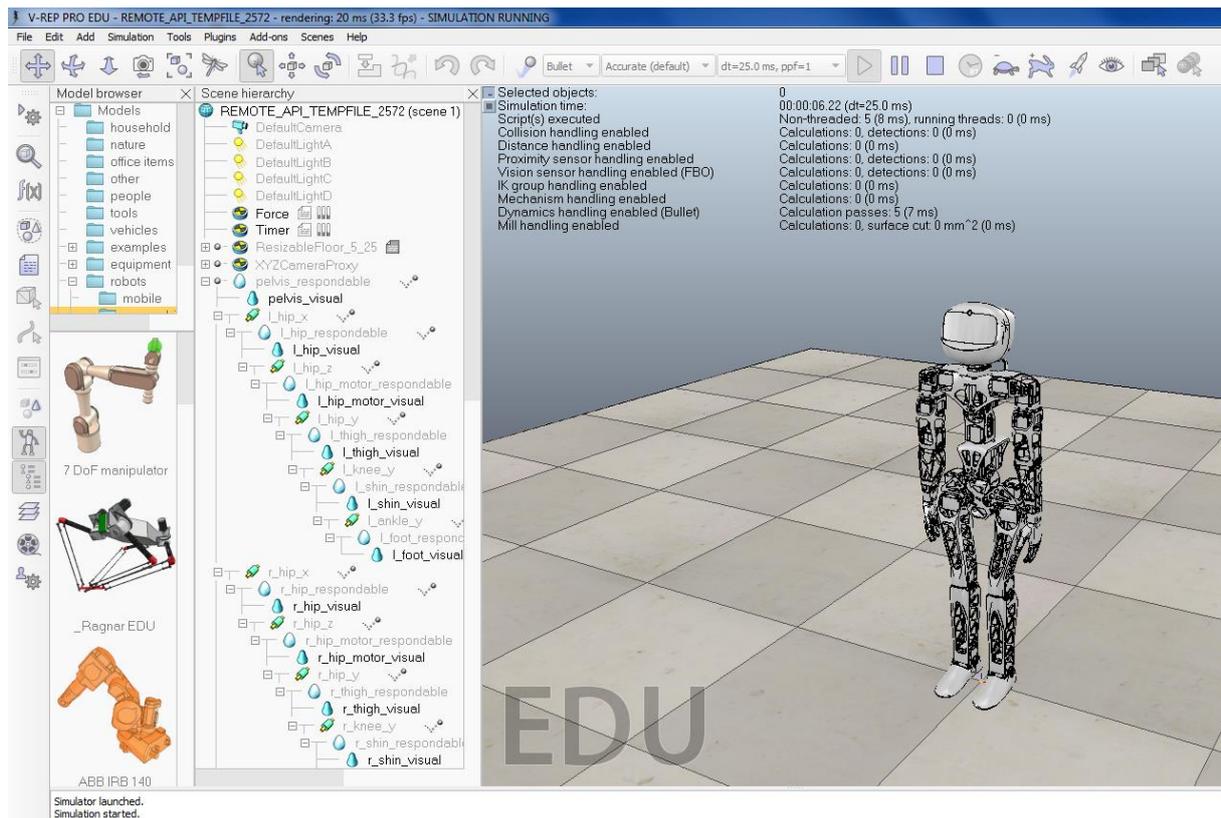


Figure 3 : Logiciel de simulation Vrep

Pour notre projet, nous allons utiliser le résultat du logiciel Xsens et coder les mouvements correspondants en Python pour que le robot les reproduise dans le simulateur.

- **MVN Studio / Xsens**

MVN Studio est le logiciel dédié à l'utilisation des capteurs de mouvements Xsens. On trouve un capteur pour chaque partie importante du corps, ceux dont nous allons avoir besoin sont :

- Tête
- Épaules
- Sternum
- Bras
- Avant-bras
- Bassin

Nous n'allons pas simuler le bas du corps pour notre projet donc seuls ces capteurs (haut du corps) nous sont utiles.

Avant de lancer la synchronisation des capteurs, il faut prendre les mesures de la personne qui les portera. Ces mesures peuvent être enregistrées pour pouvoir être réutilisées lors de prochaines captures de mouvements. Ensuite nous devons placer les capteurs à des endroits

spécifiques. Le tutoriel de Xsens est très bien fait pour commencer une première utilisation (<https://tutorial.xsens.com/>).



Figure 4 : Pose des capteurs Xsens

Une fois les capteurs placés il suffit de les synchronisés à la station (cela se fait de manière automatique) et de les calibrés grâce à la pose N aussi expliqué dans le tutoriel. Nous pouvons alors visualiser le squelette de la personne et enregistrer ses mouvements grâce à un bouton Record.

Une fois l'enregistrement fait il suffit d'exporter le fichier d'enregistrement ou de le transmettre en datagramme. La deuxième méthode peut être réalisée en même temps que l'enregistrement.

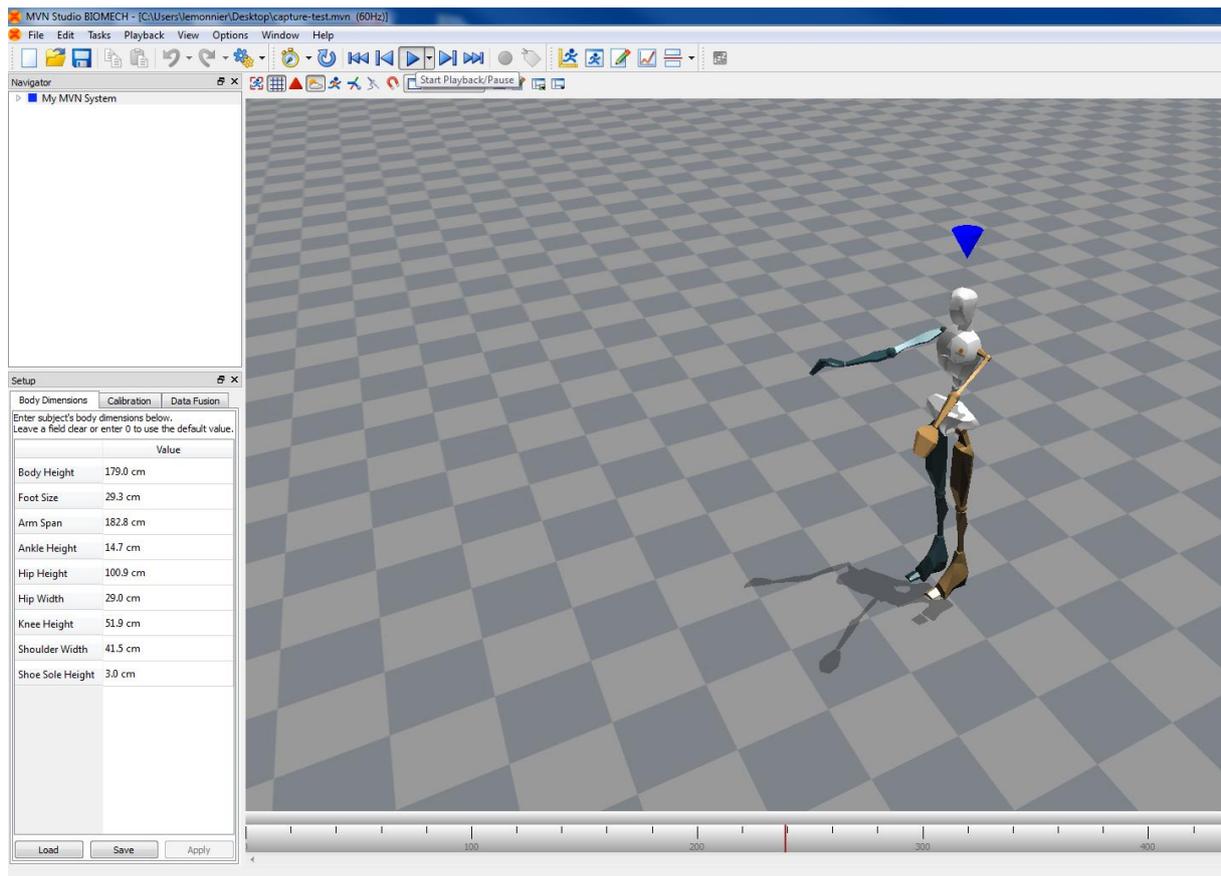


Figure 5 : Logiciel de capture de mouvement Mvn studio

Lot 2 : Recherche et conception d'une solution

Tâche 5 : Recherche les moyens de communication entre les capteurs et le robot

Nous avons donc effectué cette tâche en parallèle de la tâche 4. Durant cette tâche nous avons trouvé différentes solutions.

Les différentes sorties possibles étaient :

- Exportation aux formats
 - .BVH (Biovision Hierarchical data)
 - .FBX (Motion Builder)
 - .MVNX – (MVN Open XML format)
 - .C3D export (coordinate 3D)
- Transmission de données de mouvement en continu sur un réseau local en UDP

Nous avons regardé un peu plus en détail la transmission de données en continu pour voir si l'on pouvait retrouver facilement les mouvements de chaque membre et nous avons trouvé un document nous expliquant le contenu des datagrammes qui sont envoyés.

Tâche 6 : Etude de la solution la plus viable

Cette tâche a été raccourcie, car nous aurions manqué de temps sur la partie développement sinon. Nous avons donc décidé, en accord avec notre encadrante, que nous partirions sur la solution de transmission en continu comme nous possédions le détail du contenu des datagrammes envoyés. De plus pour éviter de perdre davantage de temps nous allons utiliser notre solution sur un enregistrement et pas en temps réel pour éviter de devoir mettre à chaque fois la tenue avec les capteurs. Théoriquement cela devrait être aussi rapide en temps réel et ce sera testé si jamais nous avons du temps disponible avant la soutenance.

Lot 3 : Développement de la solution

Tâche 7 : Développement de la solution

Cette tâche consiste à développer la brique de liaison entre le dispositif de capteur de mouvement Xsens et le simulateur Vrep. En effet, nous avons réussi à capturer les mouvements et à les sauvegarder d'une part et nous savons commander le robot dans le simulateur à partir d'un script python d'autre part. Il s'agit maintenant d'établir la liaison entre ces deux composants.

Comme évoqué précédemment, l'envoi des informations relatives aux captures de mouvement est fait via UDP sur un réseau local ou bien sur le même équipement. Afin de réceptionner ces informations, nous utilisons dans notre script une socket UDP qui écoute sur le port 9763 défini par défaut, mais paramétrable sur le MVN studio.

Afin de fiabiliser le traitement des données de captures, nous allons stocker dans un tampon les datagrammes qui arrivent à une fréquence élevée lors du streaming des mouvements. Ainsi, il n'y aura pas de perte de datagramme.

Dès lors, il convient de retrouver dans ces datagrammes, les informations utiles pour la reproduction par Poppy des mouvements capturés. Toutes les positions du robot sont la conséquence de la position des différents actionneurs sur un angle précis. Ainsi, il suffit de transmettre pour chaque moteur l'angle à adopter et le robot décrit la bonne position à l'instant t. Les datagrammes peuvent contenir différents types d'informations au caractère divers et notamment des angles décrits par les différents membres du squelette étudié. Il nous faut donc faire correspondre ces deux jeux de coordonnées pour mettre en mouvement le robot.

À l'heure de la rédaction de ce rapport et malgré la documentation, nous n'avons pas encore réussi à récupérer les données pertinentes dans les datagrammes. Nous avons réussi à identifier certaine séquence en utilisant la conversion décimale vers hexadécimal puis vers ASCII mais pas celles qui nous sont utiles. Nous avons bon espoir qu'avec le temps supplémentaire jusqu'à la soutenance nous arriverons à finaliser le projet pour atteindre les objectifs du cahier des charges. Il semble que cela représente seulement quelques lignes de codes mais que nous n'avons pas encore identifiées.

Répartition réelle des tâches

Comme nous le montre le diagramme de Gantt si dessous, nous avons beaucoup changé l'organisation des tâches.

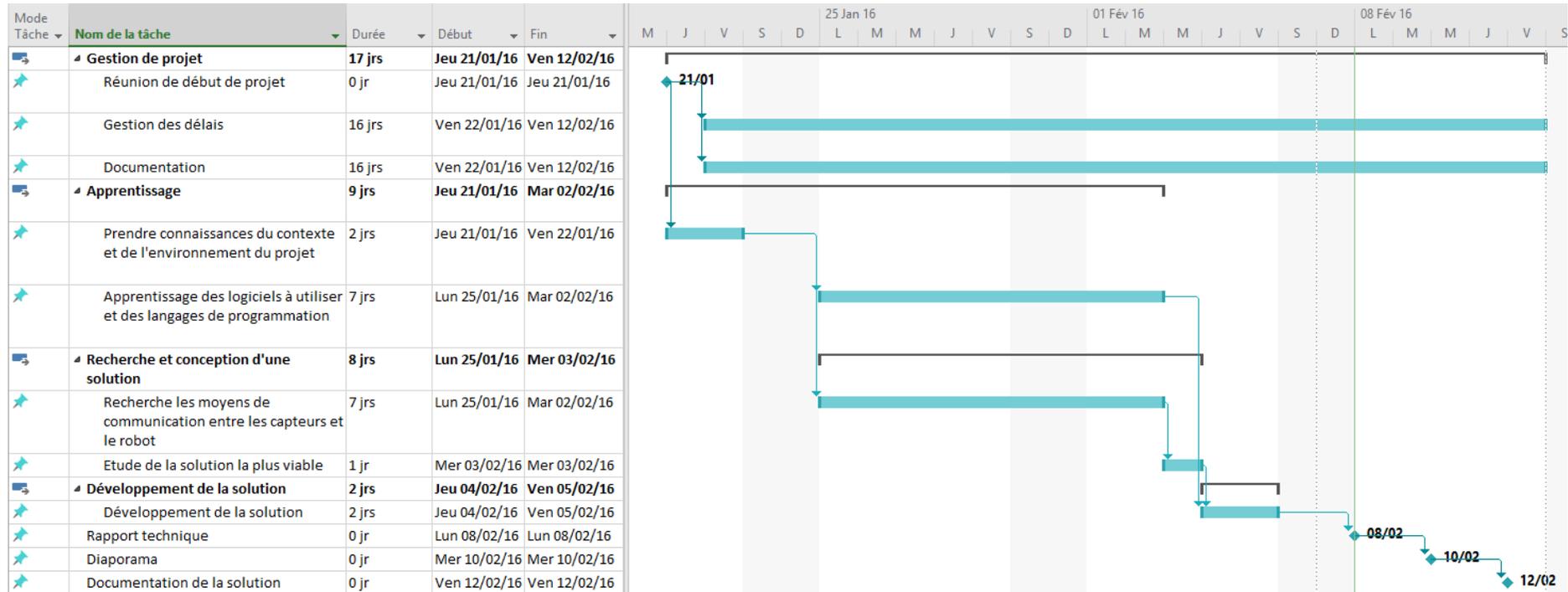


Figure 6 : Diagramme de Gantt effectif

Identification des rôles de chaque membre du groupe

Nous avons essayé de séparer les responsabilités selon si elles sont techniques ou si elles correspondent à de la gestion de projet. Ainsi Arthur s'occupait plus de la technique et Thomas plus de la gestion. Cependant pour ce qui est du travail dans chaque tâche nous nous répartissions au cas par cas selon les besoins.

N° Lot ou tâche	Intitulé	Responsable
Lot 0	Gestion de Projet	Thomas Caminati
Tâche 1	Gestion des délais	Thomas Caminati
Tâche 2	Documentation	Thomas Caminati
Lot 1	Apprentissage	Thomas Caminati
Tâche 3	Prendre connaissance du contexte et de l'environnement du projet	Arthur Le Monnier
Tâche 4	Apprentissage des logiciels à utiliser et des langages de programmation	Thomas Caminati
Lot 2	Recherche et conception d'une solution	Arthur Le Monnier
Tâche 5	Recherche les moyens de communication entre les capteurs et le robot	Arthur Le Monnier
Tâche 6	Etude de la solution la plus viable	Arthur Le Monnier
Lot 3	Développement de la solution	Arthur Le Monnier
Tâche 7	Développement de la solution	Arthur Le Monnier

Figure 7 : Répartition des rôles dans le projet

Bilan critique du travail et des résultats

Tout d'abord, ce projet nous a permis d'appréhender de nouveaux outils qui sont autant de technologies différentes de celles que nous utilisons dans le cadre professionnel. Abordant des thématiques différentes comme la capture de mouvement ou la robotique, notre travail nous a conduit à nous informer sur les problématiques associées à ces thématiques. Grâce à ce projet nous avons eu la chance d'accéder à du matériel professionnel tel que la solution Xsens. Nous avons également découvert la plateforme open source humanoïde Poppy.

N'ayant jamais utilisé le langage Python, nous avons pu acquérir de l'expérience sur ce langage grâce à notre connaissance théorique de la programmation et ainsi que celle d'autres langages acquises au cours de notre formation.

Malgré une mauvaise gestion des risques et des aléas importants lors du déroulement du projet qui nous ont sensiblement retardés et conduit à la non-finalisation du développement, nous pensons avoir été utile par notre travail de découverte de la solution Xsens. Les connaissances recueillies étant documentées dans le wiki lié à Poppy, notre travail pourra être retrouvé et poursuivi.

Même si Poppy et le dispositif de capture de mouvement associé ne constituent pas encore un assistant de kinésithérapie fiable et opérationnel, ils sont une petite pierre portée à l'édifice de l'amélioration des soins aux patients et de leurs conditions de vie.

Table des figures

Figure 1 : Tableau des lots et tâches effectuées	5
Figure 2 : Diagramme de Gantt initial de notre projet.....	6
Figure 3 : Logiciel de simulation Vrep	8
Figure 4 : Pose des capteurs Xsens.....	9
Figure 5 : Logiciel de capture de mouvement Mvn studio.....	10
Figure 6 : Diagramme de Gantt effectif.....	13
Figure 7 : Répartition des rôles dans le projet	14

Bibliographie

- Tutoriel pour apprendre à utiliser V-Rep avec la simulation du robot Poppy :
<http://nbviewer.jupyter.org/github/poppy-project/poppy-humanoid/blob/master/software/samples/notebooks/Controlling%20a%20Poppy%20humanoid%20in%20V-REP%20using%20pypot.ipynb>
- Wiki de Poppy (seulement accessible via le réseau de Telecom Bretagne) :
<http://10.29.227.201/doku.php?id=poppy>
- Site expliquant les différentes fonctions et utilisation du logiciel MVN :
<https://www.xsens.com/products/mvn-biomech/>
- Tutoriel d'aide à la première utilisation :
<https://tutorial.xsens.com/>
- Documentation concernant la transmission des données de mouvement en continu :
http://issuu.com/xsensmvn/docs/mvn_studio_real-time_network_stream

Annexes

Analyse de la différence entre les deux diagrammes de Gantt

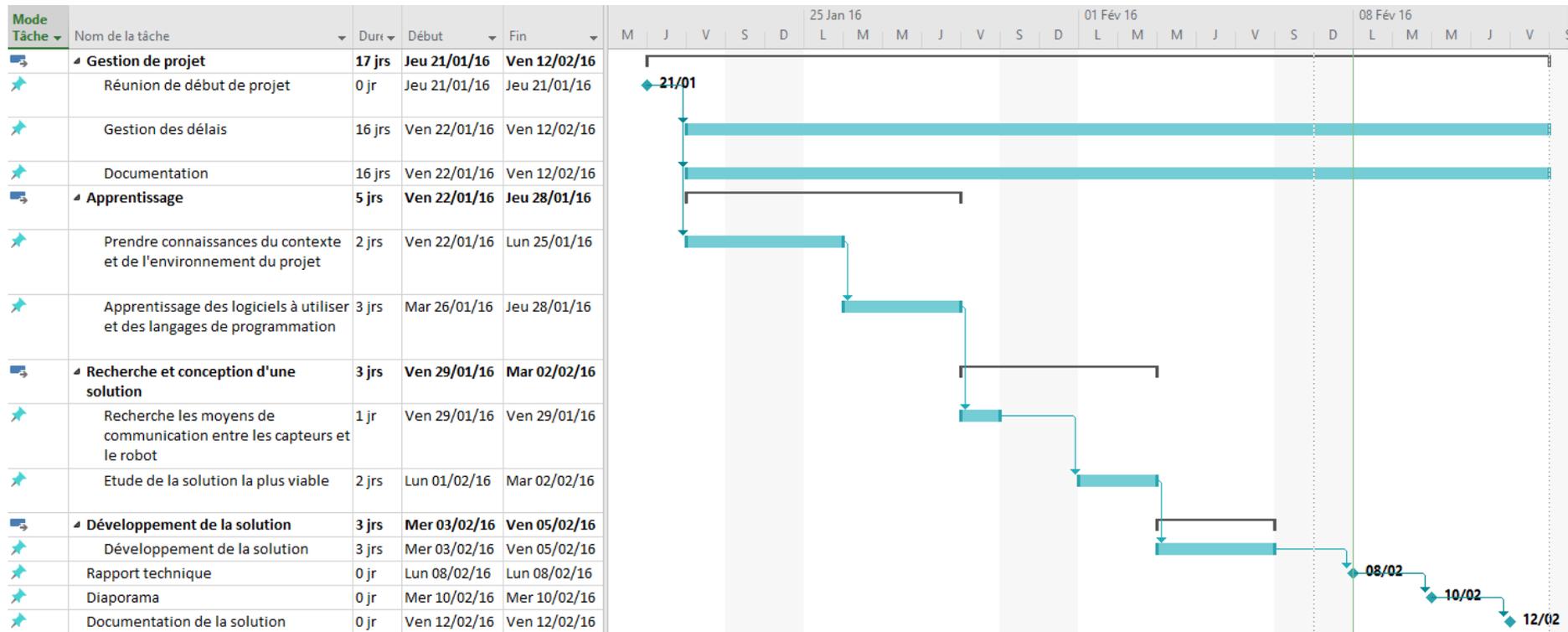


Diagramme de Gantt initial

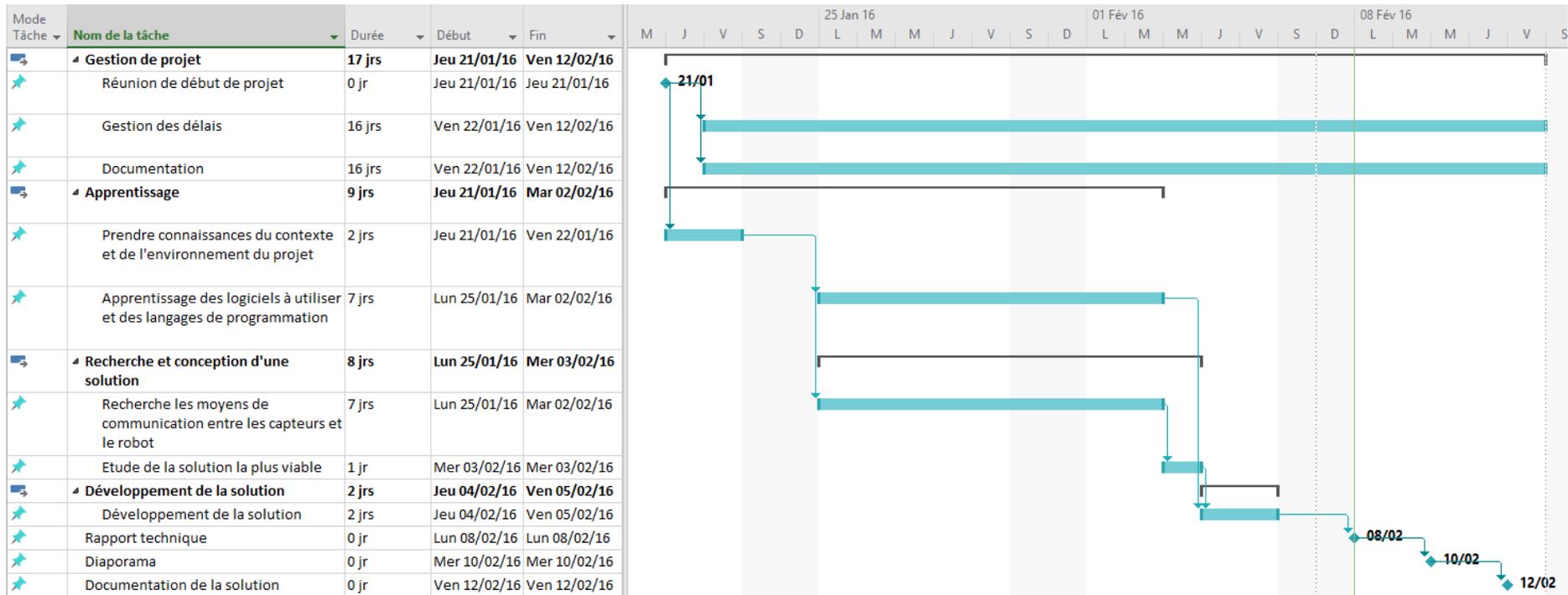


Diagramme de Gantt effectif

Comme on peut le voir, les deux diagrammes de Gantt sont assez différents car nous avons rencontré un problème qui n'a pas été pris en compte dans les risques du projet. Tout d'abord, lors de la tâche 4 d'apprentissage des logiciels, nous n'avons pas anticipé le fait que les problèmes de système d'exploitation puissent arriver (le logiciel étant en 64 bits et l'OS en 32 bits). Ce problème reste assez mineur puisque nous avons rapidement demandé si nous pouvions utiliser nos propres ordinateurs étant donné que l'activation logicielle n'est pas liée à une machine mais à une clé dongle comme expliqué dans la partie concernant la tâche 4.

Le deuxième problème pouvait paraître mineur mais demander du temps à résoudre puisque la clé dongle seule n'était pas suffisante et qu'il nous manquait une licence d'activation. Cependant, nous avons malgré tout continué à travailler en suspendant la tâche 5 (recherche de moyens de communication entre les capteurs et le robot).

On remarque aussi que la tâche 5 a été beaucoup plus longue que prévu à cause du fait que nous ne pouvions pas regarder les différents formats sur le logiciel et voir leur construction.

En définitive, ce changement a pu compenser une partie du retard apporté par la tâche 4 mais se fait quand même ressentir sur le temps passé à rechercher une solution et à la développer.