Faucheuse guidée par GPS RTK

Cette faucheuse robot est capable de couper l'herbe de façon entièrement automatique sur un parcours déterminé à l'avance. Grâce au guidage par GPS RTK le parcours est reproduit à chaque tonte avec une précision meilleure que 10 centimètres.

Difficulté **Difficile** | Durée **12 mois** | Catégories **Électronique, Machines & Outils, Robotique**

Coût **750 EUR (€)**

Sommaire

Introduction
Video d'introduction
Étape 1 - Les objectifs
Étape 2 - Présentation générale
Étape 3 - Mode d'emploi (1/4)
Étape 4 - Mode d'emploi (2 /4)
Étape 5 - Mode d'emploi (3/4)
Étape 6 - Mode d'emploi (4/4)
Étape 7 - Partie mécanique
Étape 8 - Le GPS RTK (1/3)
Étape 9 - Le GPS RTK (2/3)
Étape 10 - Le GPS RTK (3/3)
Étape 11 - La boussole (1/3)
Étape 12 - La boussole (2/3)
Étape 13 - La boussole (3/3)
Étape 14 - Principe de navigation avec GPS et boussole
Étape 15 - Proposition d'ajout d'un asservissement de position.
Étape 16 - Les deux types de virages
Étape 17 - Partie électrique (1/2)
Étape 18 - Partie électrique (2/2)
Étape 19 - Le programme de pilotage Arduino
Étape 20 - La barre de coupe et sa gestion
Étape 21 - Quelle suite donner ? Quelles améliorations ?
Commentaires

Introduction

On va décrire ici une faucheuse robot capable de couper l'herbe de façon entièrement automatique sur un parcours déterminé à l'avance. Grâce au guidage par GPS RTK le parcours est reproduit à chaque tonte avec une précision meilleure que 10 centimètres (mon expérience) . Le pilotage est basé sur une carte Aduino Mega, complétée par quelques shields de commande moteurs, d'accéléromètres et boussole ainsi que d'une carte mémoire.

C'est une réalisation non professionnelle, mais qui m'a permis de me rendre compte des problèmes rencontrés en robotique agricole. Cette toute jeune discipline est en train de se développer rapidement, aiguillonnée par les nouvelles lois sur la réduction des désherbants et pesticides. Voici par exemple un lien vers le dernier salon de robotique agricole de Toulouse (https://www.fira-agtech.com/). Certaines entreprises comme Naio technologies fabriquent déjà des robots opérationnels (https://www.naio-technologies.com/).
En comparaison, ma réalisation est bien modeste mais elle permet tout de même de comprendre l'intérêt et les défis de façon ludique.
....Et puis elle fonctionne vraiment ! ..... et peut donc être utilisée pour couper l'herbe autour de sa maison, tout en préservant son temps libre...

Même si je ne décris pas la réalisation jusque dans les derniers détails, les indications que je donne sont précieuses pour celui qui voudrait se lancer. N'hésitez pas à poser des questions ou à faire des suggestions, ce qui me permettra de compléter ma présentation au bénéfice de tous.

Je serais vraiment ravi que ce type de projet puisse donner le goût de l'ingénierie à des bien plus jeunes que moi.... afin d'être prêts pour la grande robolution qui nous attend....
D'ailleurs ce type de projet conviendrait parfaitement à un groupe de jeunes motivés dans un club ou un fablab, pour s'exercer à travailler en groupe projet, avec architectes mécanique, électrique, software chapeautés par un ingénieur système, comme dans l'industrie.

La video d'introduction:
Dans la première partie, on voit la machine se déplacer en ligne droite entre deux points du fichier de fauchage. Les inégalités de terrain obligent à de petites corrections de trajectoire. Au bout de la ligne droite, deux types de virages sont possibles: autour du centre de rotation situé au milieu des deux roues arrière ou bien autour d'une roue de façon à réaliser des bandes de fauchage contiguës. Bien que tournant à 1200 tr/min, les 3 lames de cutter des disques de coupe sont bien visibles grâce à l'effet stroboscopique.

Ce qui frappe également, ce sont les grandes roues et la lenteur de déplacement. Les grandes roues sont nécessaires pour se déplacer sur des terrains comportant des irrégularités importantes (les 3 roues sont motorisées). La lenteur de déplacement est liée au besoin de couple important nécessaire pour se déplacer sur des prés pentus.

Matériaux

- Faucheuse_guid_e_par_GPS_RTK_Liste_GPS.pdf
- Faucheuse_guid_e_par_GPS_RTK_ListeMeca.pdf

Outils

Étape 1 - Les objectifs

Le but est de fabriquer un prototype opérationnel de faucheuse capable de faucher l'herbe de manière autonome sur des terrains pouvant comporter des irrégularités importantes (prés plutôt que pelouses).

Le confinement dans le champ ne peut pas être basé sur une limitation par barrière physique ou par fil guide enterré comme pour les robots de tonte pour les pelouses. Les champs à faucher sont en effet variables et de surface importante.

Pour la barre de coupe, l'objectif est de maintenir la pousse de l'herbe à une certaine hauteur après une première tonte ou débroussaillage obtenus par un autre moyen.

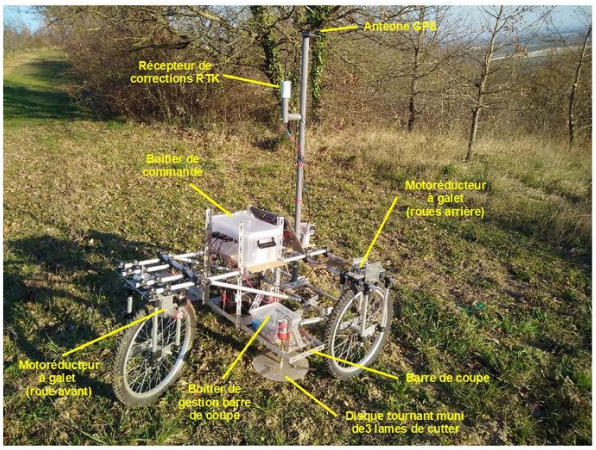
Étape 2 - Présentation générale

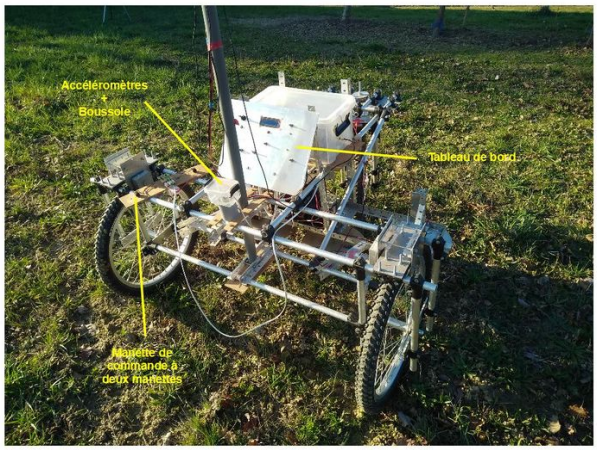
Le système se compose d'un robot mobile et d'une base fixe.

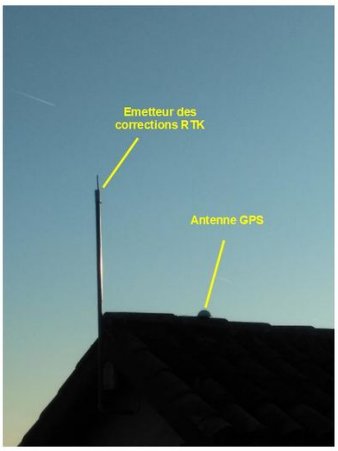
Sur le robot mobile on trouve:
- Le tableau de bord
- Le boîtier de commande général incluant une carte mémoire.
- la manette de commande manuelle
- Le GPS configuré en "rover" ainsi que le récepteur de corrections RTK
- 3 roues motorisées
- Les moteurs à galets des roues
- la barre de coupe constituée de 4 disques tournants portant chacun 3 lames de cutter en périphérie (largeur de coupe de 1 mètre)
- le boîtier de gestion de la barre de coupe
- les batteries

Dans la base fixe on trouve le GPS configuré en "base" ainsi que l'émetteur des corrections RTK. On note que l'antenne est placée en hauteur de façon à rayonner sur quelques centaines de mètres autour de la maison. De plus, l'antenne GPS est en vue de tout le ciel sans aucune occultation par des bâtiments ou par de la végétation.

Les modes Rover et base des GPS seront décrits et expliqués dans la partei GPS.





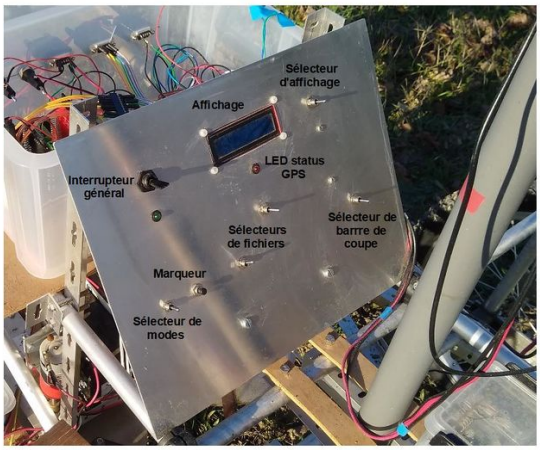


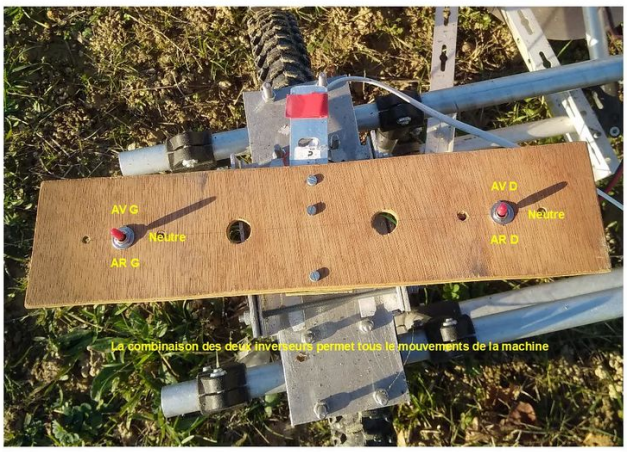
Étape 3 - Mode d'emploi (1/4)

Je propose de prendre connaissance du robot à travers son mode d'emploi qui fait bien apparaître toutes ses fonctionnalités.

Description du tableau de bord:
- Un interrupteur général
- un premier sélecteur à 3 positions permet de sélectionner les modes de fonctionnement: mode déplacement manuel, mode enregistrement de parcours, mode fauchage
- Un bouton poussoir sert de marqueur. On verra ses utilisations.
- Deux autres sélecteurs à 3 positions servent à sélectionner un numéro de fichier parmi 9. On dispose donc de 9 fichiers de fauchage ou d'enregistrements de parcours pour 9 champs différents.
- Un sélecteur à 3 positions est dédié à la commande de la barre de coupe. Une position OFF, une position ON, une position commande programmée.
- Afficheur deux lignes
- un sélecteur 3 positions pour définir 3 affichages différents
- une LED qui indique l'état du GPS. LED éteinte, pas de GPS. LED clignotante lentement, GPS sans corrections RTK. LED clignotante rapide, corrections RTK reçues. LED allumée, verrouillage GPS sur la plus grande précision.

Enfin, **la manette de commande** manuelle est munie de deux sélecteurs à 3 positions. Celui de gauche commande la roue gauche, celui de droite commande la roue droite.



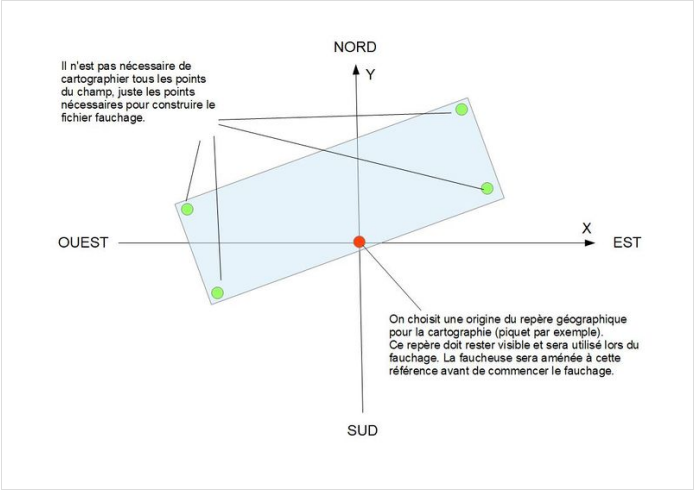


Étape 4 - Mode d'emploi (2 /4)

Mode pilotage manuel (GPS non nécessaire)
Après mise ON et sélection de ce mode avec le sélecteur de mode, le pilotage de la machine se fait avec la manette de commande.
Les deux sélecteurs à 3 positions ont un ressort de rappel qui les ramène toujours en position milieu, correspondant à l'arrêt des roues.
Quand on pousse les leviers de gauche et de droite vers l'avant les deux roues arrière tournent et la machine va tout droit.
Quand on tire les deux leviers en arrière, la machine recule tout droit.
Quand on pousse un levier vers l'avant, la machine tourne autour de la roue arrêtée.
Quand on pousse un levier vers l'avant et l'autre vers l'arrière, la machine tourne autour d'elle-même, en un point situé au milieu de l'axe joignant les roues arrière.
La motorisation de la roue avant s'ajuste automatiquement en fonction des deux commandes passées sur les deux roues arrière.

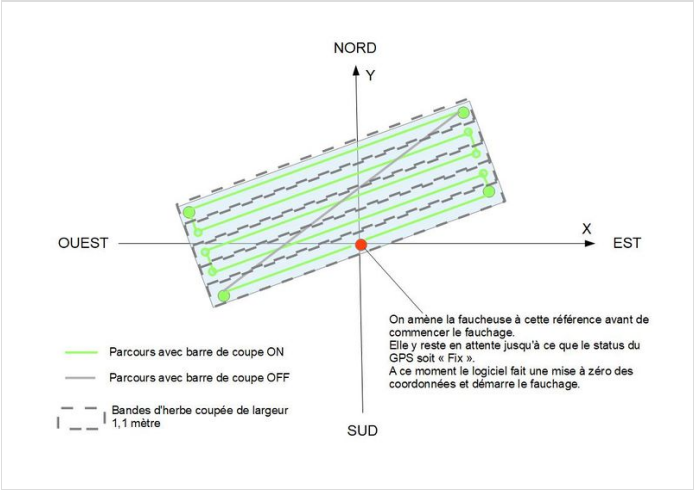
Enfin, dans le mode manuel on peut également faucher de l'herbe. Pour cela, après avoir vérifié que personne ne se trouve à proximité des disques de coupe, on met ON le boîtier de gestion de la barre de coupe (interrupteur "hard" pour la sécurité). On place ensuite le sélecteur de coupe du tableau de bord sur ON. A cet instant les 4 disques de la barre de coupe se mettent en rotation.

Étape 5 - Mode d'emploi (3/4)



Mode enregistrement de parcours (GPS nécessaire)
- Avant de commencer à enregistrer un parcours, on définit arbitrairement un point de référence pour le champ et on le marque par un petit piquet. Ce point sera l'origine des coordonnées dans le repère géographique (photo)
- On sélectionne ensuite le numéro de fichier dans lequel le parcours va être enregistré, grâce aux deux sélecteurs du tableau de bord.
- On met la base ON
- On vérifie alors que la LED d'état GPS se met à clignoter rapidement.
- On quitte le mode manuel en plaçant le sélecteur de mode du tableau de bord sur la position enregistrement.
- On amène alors la machine manuellement à la position du point de référence. Précisément c'est l'antenne GPS qui doit se trouver au dessus de ce repère. Cette antenne GPS est située au dessus du point centré entre les deux roues arrière et qui est le point de rotation de la machine sur elle-même.
- On attend que la LED donnant le statut du GPS soit maintenant allumée sans clignotement. Cela indique que le GPS est à sa précision maximum ("Fix" GPS).
- On marque la position origine 0,0 en appuyant sur le BP marqueur du tableau de bord.
- On se dirige alors vers le point suivant que l'on veut cartographier. Dès qu'il est atteint, on le signale à l'aide du marqueur.
- Pour mettre fin à l'enregistrement on repasse en mode manuel.

Étape 6 - Mode d'emploi (4/4)



Mode fauchage (GPS nécessaire)
Il faut d'abord préparer le fichier de points que la machine doit parcourir pour faucher tout le champ sans laisser de surface non coupée. Pour cela on récupère le fichier enregistré dans la carte mémoire et à partir de ces coordonnées, à l'aide par exemple d'Excel, on génère une liste de points comme sur la photo. Pour chacun des points à atteindre on indique si la barre de coupe est ON ou OFF. Comme c'est la barre de coupe qui consomme le plus de puissance (de 50 à 100 Watts suivant l'herbe), il faut veiller à couper la barre de coupe lors de la traversée d'un champ déjà fauché par exemple.
La carte de fauchage étant générée, on remet la carte mémoire sur son shield dans le tiroir de commande.

Il ne reste plus alors que mettre ON la base et se rendre sur le champ à faucher, juste au dessus du repère de référence. On met ensuite le sélecteur de mode sur "Fauchage".
A ce moment la machine attendra toute seule le verrouillage GPS RTK en "Fix" pour faire une mise à zéro des coordonnées et commencer à faucher.
Lorsque le fauchage sera terminé, elle reviendra seule au point de départ, avec une précision d'une dizaine de centimètres.

Pendant le fauchage, la machine se déplace en ligne droite entre deux points consécutifs du fichier de points. La largeur de coupe est de 1,1 mètre. Comme la machine a une largeur entre roues de 1 mètre et peut tourner autour d'une roue (voir video), il est possible de faire des bandes de fauchage contiguës. C'est d'une grande efficacité !

Étape 7 - Partie mécanique

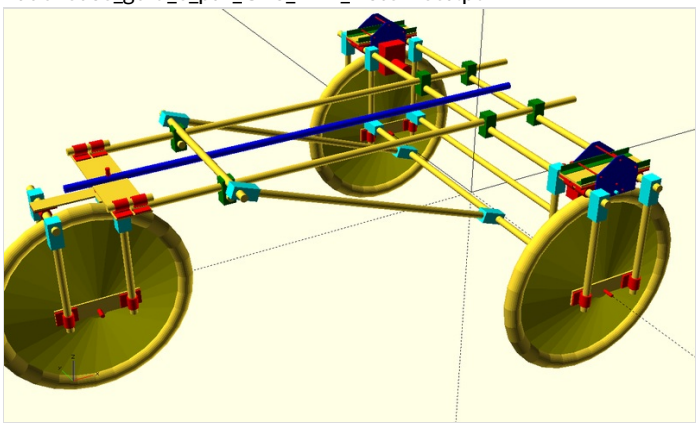
La structure du robot
Le robot est construit autour d'une structure en treillis de tubes d'aluminium, ce qui lui confère une bonne rigidité. Ses dimensions sont environ de 1.20 mètre de long, 1 mètre de large et 80 cm de haut.

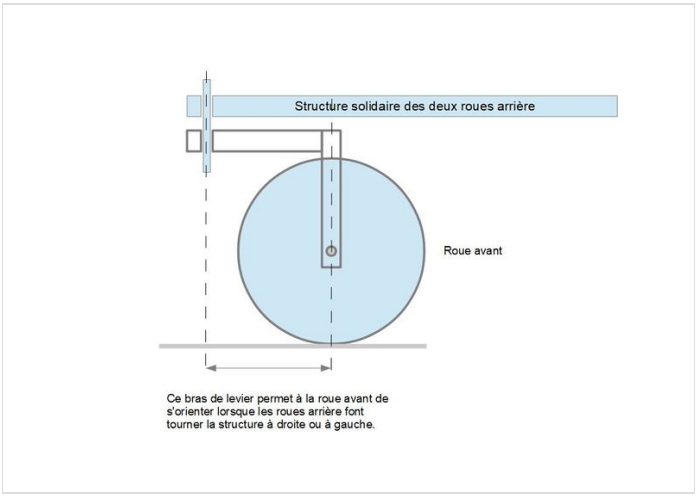
Les roues
Elle peut se déplacer grâce à 3 roues de vélo enfant de diamètre 20 pouces: Deux roues arrière et une roue avant semblable à la roulette des chariots de supermarché (photos 1 et 2). C'est le mouvement relatif des deux roues arrière qui assure son orientation.

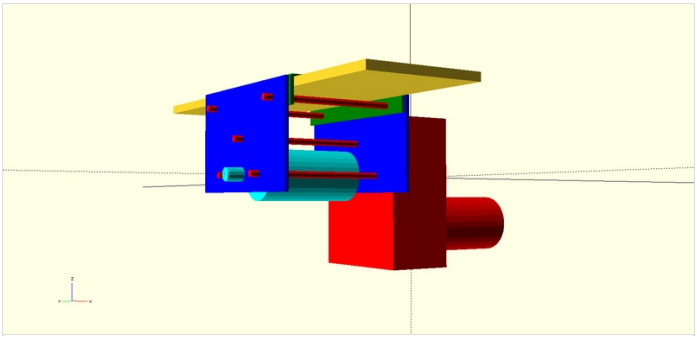
Les moteurs à galet
A cause des irrégularités de terrain, il est nécessaire de disposer de couples importants et donc d'un grand rapport de réduction. Dans ce but j'ai utilisé le principe de galet appuyant sur la roue, comme sur un solex (photos 3 et 4). La réduction importante permet de garder la machine stable dans une pente, même lorsque l'on coupe l'alimentation des moteurs. En contrepartie, la machine avance lentement (3 mètres / minute)...mais l'herbe pousse lentement....
On peut noter que les 3 roues sont motorisées avec des moteurs à galet. La motorisation de la roue avant permet d'éviter le blocage de cette roue directrice lorsqu'elle tombe dans un creux important.

Pour la conception mécanique j'ai utilisé le logiciel de dessin OpenScad (logiciel à scripts très efficace). En parallèle pour les plans de détail j'ai utilisé Drawing de OpenOffice.

Enfin voici la liste de sites où les différents éléments mécaniques ont été achetés:
Faucheuse_guid_e_par_GPS_RTK_ListeMeca.pdf







Étape 8 - Le GPS RTK (1/3)

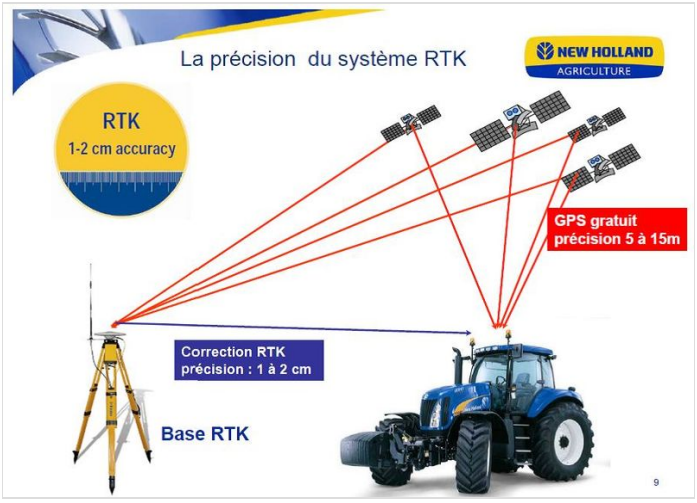
GPS simple

Le GPS simple (photo 1), celui qui est dans notre voiture n'a une précision que de quelques mètres. Si on enregistre la position indiquée par un tel GPS maintenu fixe pendant une heure par exemple, on va observer des fluctuations de plusieurs mètres. Ces fluctuations sont dues à des perturbations de l'atmosphère et de la ionosphère, mais aussi à des erreurs d'horloge des satellites et des erreurs de meure du GPS lui-même. Il ne convient donc pas pour notre application.

GPS RTK

Afin d'améliorer cette précision, on utilise deux GPS situés à une distance inférieure à 10 Km (photo 2). Dans ces conditions, on peut considérer que les perturbations de l'atmosphère et de la ionosphère sont identiques sur chaque GPS. Ainsi la différence de position entre les deux GPS n'est plus perturbée (différentiel). Si maintenant on fixe un des GPS (la base) et si on place l'autre sur un véhicule (le rover), on obtiendra précisément le déplacement du véhicule par rapport à la base sans perturbations. De plus ces GPS effectuent une mesure de temps de vol beaucoup plus présise que les GPS simples (mesures de phase sur la porteuse).
Grâce à ces améliorations, on obtiendra une précision de mesure centimétrique pour le déplacement du rover par rapport à la base.

C'est ce système RTK (Real Time Kinematic) que nous avons choisi d'utiliser.



Étape 9 - Le GPS RTK (2/3)

J'ai acheté 2 circuits GPS RTK (photo 1) à la société Navspark.
Ces circuits sont montés sur un petit circuit imprimé équipé de broches au pas de 2.54 mm qui se monte donc directement sur les plaques d'essai.
Comme le projet est localisé dans le sud-ouest de la France, j'ai choisi des circuits travaillant avec les constellations de satellites américains GPS ainsi que la constellation russe Glonass.
Il est important d'avoir le maximum de satellites afin de bénéficier de la meilleure précision. Dans mon cas, j'ai couramment entre 10 et 16 satellites.

On doit également acheter
- 2 adaptateurs USB, nécessaire pour relier le circuit GPS à un PC (tests et configuration)
- 2 antennes GPS + 2 câbles adaptateurs
- une paire d'emetteurs-récepteurs 3DR afin que la base puisse émettre ses corrections vers le rover et le rover les recevoir.

Ci-joint la liste des achats avec des liens internet vers les revendeurs:
Faucheuse_guid_e_par_GPS_RTK_Liste_GPS.pdf

Étape 10 - Le GPS RTK (3/3)

La notice du GPS que l'on trouve sur le site de Navspark permet de mettre en œuvre progressivement les circuits.
http://navspark.mybigcommerce.com/content/NS-HP-GL-User-Guide.pdf

Sur le site de Navspark on trouvera également
- le logiciel à installer sur son PC Windows pour visualiser les sorties des GPS et programmer les circuits en base et en rover.
- Une description du format des données GPS (phrases NMEA)

Tous ces documents sont en anglais mais ils sont relativement faciles à comprendre. Dans un premier temps, la mise en œuvre se fait sans le moindre circuit électronique grâce aux adaptateurs USB qui fournissent également toutes les alimentations électriques.
La progression est la suivante:
- Test des circuits individuellement qui fonctionnent comme des GPS simples. La visualisation en nuage de ponts montre une stabilité de quelques mètres.
- Programmation d'un circuit en ROVER et l'autre en BASE
- Constitution d'un système RTK en reliant les deux modules par un simple fil. La visualisation en nuage de ponts montre une stabilité relative ROVER/BASE de quelques centimètres !
- Remplacement du fil liant BASE et ROVER par les émetteurs-récepteurs 3DR. Là encore le fonctionnement en RTK permet une stabilité de quelques centimètres. Mais cette fois BASE et ROVER ne sont plus reliés par un lien physique.....
- Remplacement de la visualisation PC par une carte Arduino programmée pour recevoir les données GPS sur une entrée série... (voir plus bas)

Nota: Sur les photos fournies à l'étape 2 on voit que le GPS du rover se trouve au sommet d'un mât. C'est une précaution que j'avais prise pour éviter les effets d'écran de mon corps près de la machine lors de la mise au point. En fait, cette précaution n'est plus utile lorsque la mise au point est terminée et que le rover évolue seul sans présence humaine autour.

Étape 11 - La boussole (1/3)

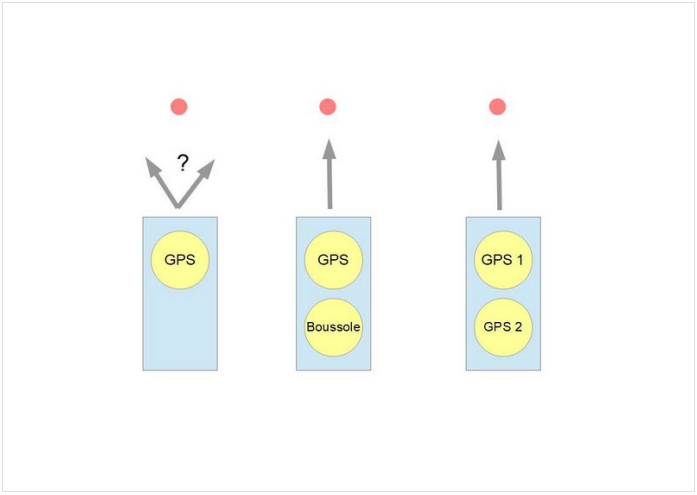
Un GPS ne donne que sa propre position dans le repère géographique sans aucune indication de direction. Cela ne suffit pas pour se diriger efficacement vers un autre point à atteindre, comme le suggère la figure 1.

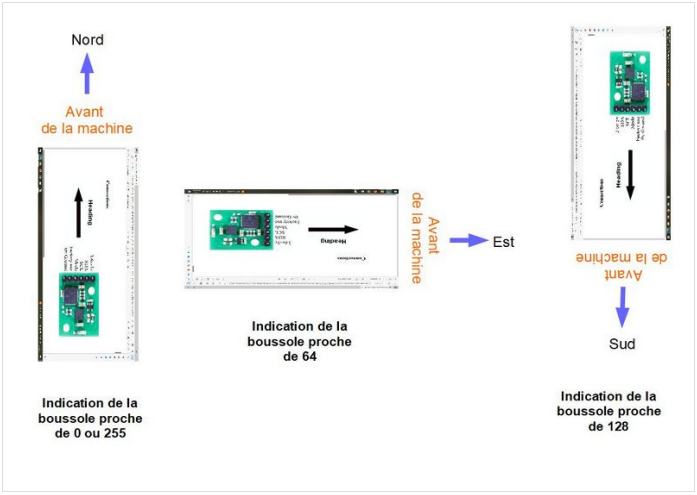
L'idée qui vient naturellement est d'adjoindre une boussole à ce GPS. Dans ce cas il sera nécessaire de caler l'indication de la boussole avec le repère géographique.

Comme le montre également la figure 1, on peut aussi remplacer la boussole par un GPS additionnel. Dans ce cas la différence de position entre les deux GPS permet de calculer directement l'orientation par rapport au repère géographique, sans soucis de recalage de repères. Je n'ai pas choisi cette solution par souci d'économie.

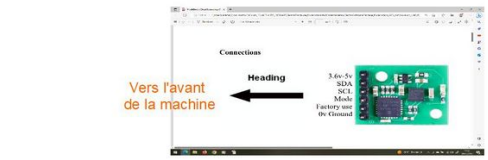
La boussole que j'ai utilisée est la CMPS11. Actuellement ce modèle est remplacé par la CMPS12, basée sur un circuit BNO055, qui est très proche. cmps12.pdf (robot-electronics.co.uk)

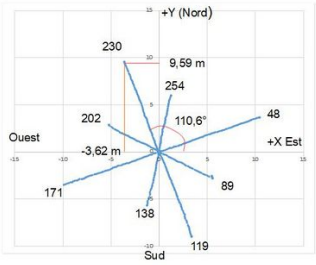
J'ai monté cette boussole sur ma machine en orientant vers l'avant la direction « HEADING » fournie par la documentation. La figure 2 montre alors comment est la lecture de la boussole en fonction de l'orientation de la machine. Le Nord est le nord magnétique, celui indiqué par une boussole à aiguille magnétique.





Étape 12 - La boussole (2/3)





Coordonnées		indication		angle
repère géographique		boussole		repere geographique
X (m)	Y (m)	LSB		(°)
-3,62	9,59	230		110,680
3,36	-8,98	119		-69,486
10,42	3,67	48		19,403
-9,91	-3,48	171		-160,651
1,22	5,99	254		78,488
-1,27	-5,66	138		-102,647
-5,22	2,89	202		151,029
5,53	-2,74	89		-26,357

Pourquoi "caler" les indications de la boussole sur le repère GPS ?

Le repère dans lequel le GPS donne les positions est le repère géographique basé sur l'axe de rotation de la terre. Dans ce repère, la direction +Y (figure 1) correspond au Nord géographique. Un déplacement le long de cet axe vers +Y correspond à une augmentation pure de la latitude. De même un déplacement vers +X correspond à une augmentation pure de la longitude Est.

En comparaison, la boussole indique le Nord magnétique qui est en général décalé d'une valeur variable dans le temps, la déclinaison magnétique (Déclinaison magnétique terrestre — Wikipédia (wikipedia.org)). Actuellement cette différence entre Nord géographique et Nord magnétique est de l'ordre du degré.

Actuellement, la boussole étant montée avec HEADING vers l'avant, lorsque la machine se déplace vers le Nord géographique (+Y) on peut s'attendre à une indication de la boussole proche de 0 – 255.

En réalité il n'en est pas tout à fait ainsi pour plusieurs raisons :
- Des offsets dus à la déclinaison magnétique et à l'erreur de calage en rotation de la boussole sur la machine.
- des variations plus ou moins cycliques sans doutes dues aux perturbations magnétiques des différents moteurs. (voir la suite).

C'est donc la raison pour laquelle il faut établir expérimentalement une relation entre les indications de la boussole et les directions dans le repère du GPS.

Étape 13 - La boussole (3/3)

Comment "caler" les indications de la boussole sur le repère GPS ?

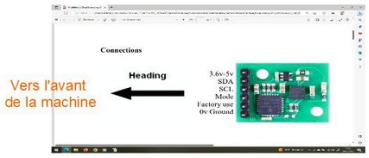
On utilise pour cela le mode enregistrement de parcours (voir étape 5), en faisant se déplacer la machine selon un certain nombre de rayons couvrant la totalité de l'azimut de 360°. Des repères au sol permettent de se guider afin d'assurer un pilotage manuel en ligne droite.

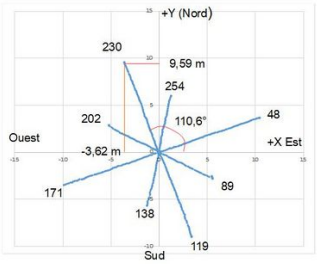
La figure 1 montre par exemple ce que j'ai fait obtenu avec ma machine après fixation de la boussole dans sa position définitive. Toutes les 3 seconde, sont enregistrées une valeur position GPS et une valeur boussole.

On peut alors simplement représenter ces positions GPS traduites en mètres dans le repère XY géographique et en même temps porter la valeur moyenne des valeurs boussole obtenues lors du parcours de chaque rayon.
Toutes ces valeurs peuvent être écrites selon deux colonnes:
- celle des angles calculés dans le repère XY pour atteindre un point choisi (par exemple 110.6)
- celle des indications boussole que l'on aura si la machine se dirige vers ce point (par exemple 230)

Il ne reste plus qu' à trouver la fonction qui va nous faire passer des angles dans le repère géographique GPS aux indications de la boussole. C'est ce que réalise la fonction **boussole (angle)** dont le programme est donné sur la figure 2. Dans l'exemple donné, on peut vérifier que l'on retrouve bien les indications boussole à mieux que 2 LSB

La valeur **but = boussole (angle)** est ensuite utilisée par les fonctions de rotation de la machine (figure 3).





Coordonnées		indication		angle
repère géographique		boussole		repere geographique
X (m)	Y (m)	LSB		(°)
-3,62	9,59	230		110,680
3,36	-8,98	119		-69,486
10,42	3,67	48		19,403
-9,91	-3,48	171		-160,651
1,22	5,99	254		78,488
-1,27	-5,66	138		-102,647
-5,22	2,89	202		151,029
5,53	-2,74	89		-26,357

byte boussole (float angle)

```
byte boussole(float angle)      // Transformation d'un angle d'azimut en direction boussole
{
int angle_boussole;

if (angle > 180) angle = angle - 360;  // l'angle doit toujours être compris entre 180° et -180°
if (angle < 180) angle = angle + 360;  // l'angle doit toujours être compris entre 180° et -180°
if (angle < 0) {angle = angle + 360 ;}

angle_boussole = -255. / 360. * angle + 61 ;        // offset = 61 lié à  l'orientation de la boussole sur la machine
angle_boussole = angle_boussole - sin(6.28*(angle -30.)/380.)* 8. ; // correction erreur cyclique fittée sur des mesures

 if (angle_boussole > 255) {angle_boussole = angle_boussole - 255;}
 if (angle_boussole <0 ) {angle_boussole = angle_boussole + 255;}

  return angle_boussole;
}
```

```
void rotationCentre (byte but)      // Tourne autour d'un point jusqu'à ce que la boussole indique but = boussole (angle)
                                    // angle est calculé dans le repère géographique à partir des positions GPS.
                                    // boussole est la fonction qui contient le « calage » de la boussole / repère GPS
{
   lectureBoussole();
   int difference = but - RepBoussole[0];
   if (abs(difference) > 128)  difference = -difference;  // si la différence dépasse 128 en module, il faut inverser le sens

   if (difference < -1)  // RepBoussole[0] diminue quand on tourne vers la gauche
   {
     do
     {
       commandeMoteurs(0,2); // rotation vers la gauche
       lectureBoussole();
       affichageFauchage();
       delay(100);
     } while (abs(RepBoussole[0] -  but) > 1);
     commandeMoteurs(1,1);
   }

   if (difference > 1)    // RepBoussole[0] augmente quand on tourne vers la droite
   {
     do
     {
       commandeMoteurs(2,0);    // rotation vers la droite
       lectureBoussole();
       affichageFauchage();
       delay(100);
     } while (abs(RepBoussole[0] -  but) > 1);
     commandeMoteurs(1,1);
   }
```

Étape 14 - Principe de navigation avec GPS et boussole

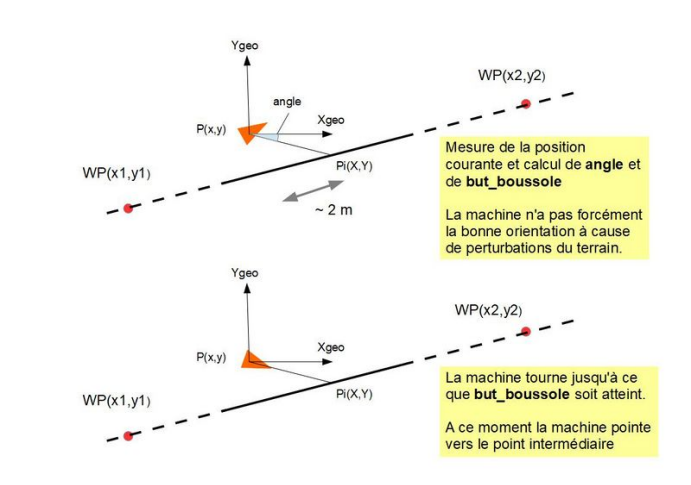
Le déplacement de la machine se fait de WP en WP selon des lignes droites.

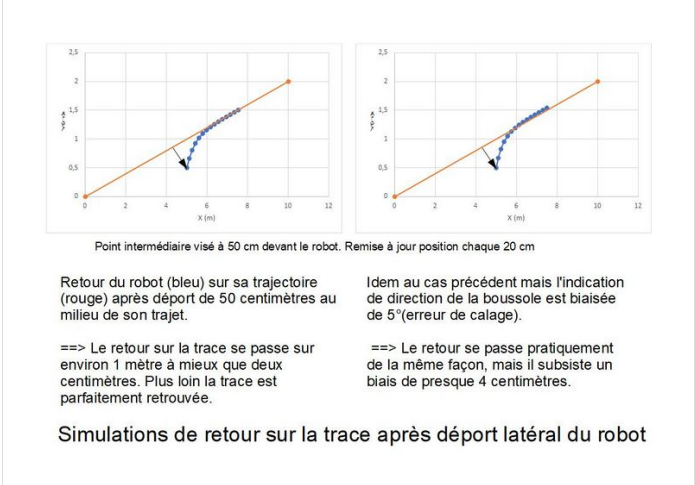
Au départ d'un WP, la boussole indique la direction du prochain WP à atteindre. Cependant, afin d'éviter
- les erreurs de direction qui croissent avec la distance parcourue
- les dérives latérales par glissement (en situation de dévers)
des points intermédiaires situés sur la droite joignant les deux WP sont calculés. Ces points intermédiaires sont situés à environ deux mètres à l'avant de la machine, comme montré sur la figure 1.

Ensuite, à chaque position courante de la machine P(x,y) on calcule l'**angle** (par rapport à Xgéo) nécessaire pour atteindre ce point intermédiaire. On calcule également l'indication de la boussole correspondant à cet angle, **but_boussole**. C'est à ce moment que sert la boussole "calibrée" par rapport au repère géographique. La machine tourne jusqu'à ce que l'indication de la boussole soit égale à **but_boussole.** Ainsi la machine pointera vers le point intermédiare, quelle que soit l'orientation de départ de la machine.

Les calculs de ces points intermédiaires et de l'angle de correction sont explicités dans :
Faucheuse_guid_e_par_GPS_RTK_PrincipeSuiviLigne_3.pdf

Ce principe de navigation donne de très bons résultats. Lors de tests, j'ai déplacé latéralement la machine de 50 cm et j'ai pu constater qu'elle réagissait rapidement pour rallier la ligne des WP (voir figure 2).
Si la machine était munie de capteurs de distance à l'avant (ce n'est pas le cas actuellement) on pourrait facilement lui faire contourner un obstacle et revenir ensuite automatiquement sur la ligne des WP.
La simulation de la figure 2 montre cependant que les erreurs boussole se traduisent par des écarts de position par rapport à la ligne des WP.





Étape 15 - Proposition d'ajout d'un asservissement de position.

Ce système n'est pas implanté dans la machine actuelle. J'ai fait des simulations qui montrent que grâce à cet asservissement PID (Correction Proportionnelle, Intégrale et Dérivée) on pourrait compenser les écarts de position induits par les erreurs de la boussole (voir paragraphe précédent).

La figure 1, montre qu'une erreur boussole de l'ordre de 5° peut introduire une erreur de suivi de ligne de presque 9 cm. C'est dommage car on gaspille ainsi la précision du GPS pourtant proche de 2 cm.

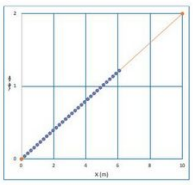
La figure 2 explique un principe d'asservissement de position PID qui permettrait de verrouiller le robot sur la ligne des WP, même en cas d'imprécision de la boussole ou d'évolution dans le temps de son calage (masses magnétiques, champ magnétique moteurs).

La figure 3 explique le calcul de l'erreur de position « d » du robot nécessaire au calcul de la correction.

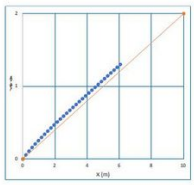
La figure 4 montre la simulation de cette correction avec ce principe d'asservissement. Une simple correction Proportionnelle et Intégrale ramène les erreurs au niveau de celles du GPS.

En conclusion, en combinant un asservissement angulaire avec la boussole et un asservissement de position par rapport à la ligne suivie, on bénéficie à la fois d'une grande efficacité et d'une très bonne précision directement reliée au GPS RTK.
Je recommanderais donc à ceux qui construisent leur propre robot, de l'implanter dans leur logiciel.

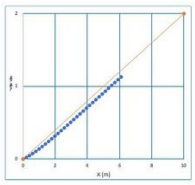
Effet d'un décalage angulaire de la boussole (simulation)



Quand la boussole indique parfaitement la direction demandée, le robot suit exactement la ligne rouge entre deux WP.

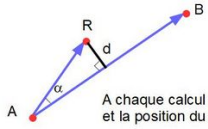


Avec une erreur boussole de 5°, le déplacement n'est plus rectiligne et le robot se trouve à presque 9 cm de la ligne des WP



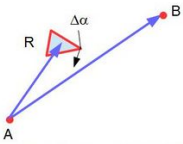
Avec une erreur boussole de -5°, le robot se déplace maintenant à presque 9 cm de la ligne des WP, mais de l'autre côté.

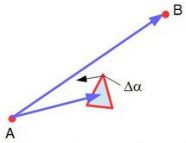
Principe de l'asservissement sur la ligne des WP



A et B sont deux WP dont les coordonnées sont lues dans le fichier fauchage.
R est la position courante du robot.

A chaque calcul de position du robot, on calcule également la distance d et la position du robot par rapport à la ligne AB (à droite ou à gauche).



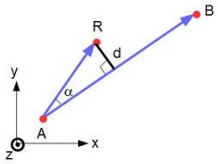


A chaque pas de calcul, on rajoute à l'angle calculé pour la boussole un angle $\Delta\alpha$ dans le sens indiqué par la flèche, de façon à ramener le robot vers la ligne AB, malgré les erreurs boussole.

$\Delta\alpha$ est calculé à partir de la valeur de l'écart d de façon à réaliser un asservissement PID :

$$\Delta\alpha = Kp \cdot d + Kd \cdot \delta d / \delta t + Ki \cdot \int d \cdot \delta t$$

Calcul de l'erreur de position d



A et B sont deux WP dont les coordonnées sont lues dans le fichier fauchage.

R est la position courante du robot

Il faut déterminer si le robot est à droite ou à gauche de la ligne AB et à quelle distance d (erreur de position)

$$\overrightarrow{AR} \wedge \overrightarrow{AB} = \begin{vmatrix} X_R - X_A \\ Y_R - Y_A \\ 0 \end{vmatrix} \wedge \begin{vmatrix} X_B - X_A \\ Y_B - Y_A \\ 0 \end{vmatrix} = \begin{vmatrix} 0 \\ 0 \\ (X_R - X_A) \times (Y_B - Y_A) - (X_B - X_A) \times (Y_R - Y_A) \end{vmatrix}$$

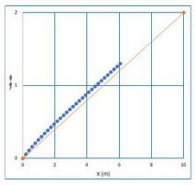
$$\left|\overrightarrow{AR} \wedge \overrightarrow{AB}\right| = \left|\overrightarrow{AB}\right| \times \left|\overrightarrow{AR}\right| \times \sin(\alpha) = \left|\overrightarrow{AB}\right| \times d$$

$$d = \frac{(X_R - X_A) \times (Y_B - Y_A) - (X_B - X_A) \times (Y_R - Y_A)}{\left|\overrightarrow{AB}\right|}$$

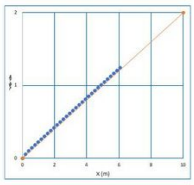
Si d > 0, R est à droite de AB

Si d < 0, R est à gauche de AB

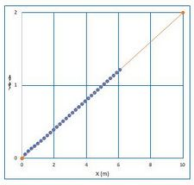
Compensation du biais boussole par un asservissement proportionnel et intégral (simulation)



On veut compenser cette erreur de position due à une erreur boussole de 5°

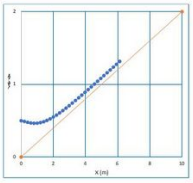


L'application d'une correction proportionnelle réduit l'erreur mais laisse une erreur de traînée.

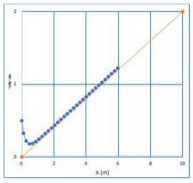


L'ajout d'un terme intégral à la correction permet de supprimer l'erreur de traînée.

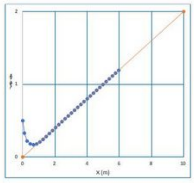
Réponse de l'asservissement à un décalage brutal (simulation)



On veut compenser cette erreur de position due à un écart de position de 50 cm en présence d'une erreur boussole de 5°



L'application d'une correction proportionnelle réduit l'erreur mais laisse une erreur de traînée.



L'ajout d'un terme intégral à la correction permet de supprimer l'erreur de traînée.

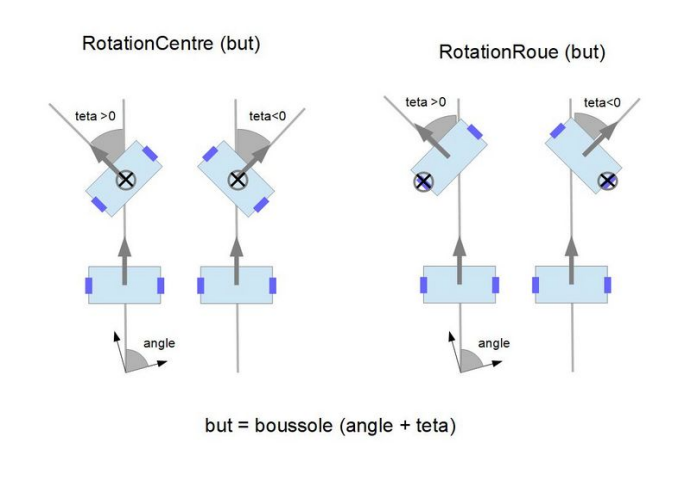
Étape 16 - Les deux types de virages

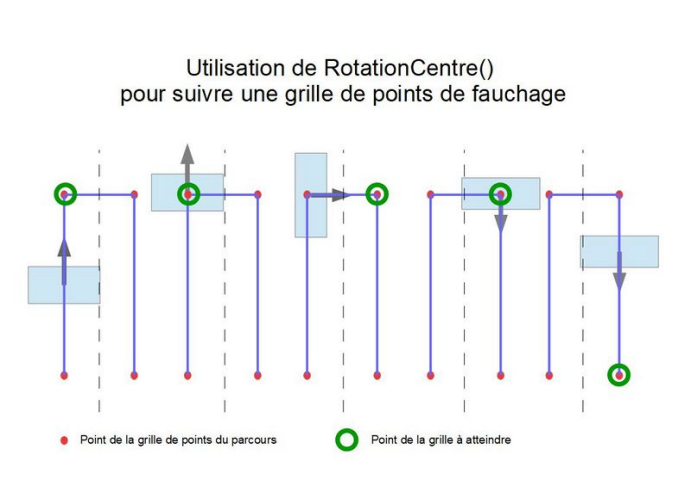
Deux types de virages sont définis (figure 1):
- virages autour du centre situé entre les deux roues arrière
- virage autour d'une roue, soit la roue gauche pour tourner à gauche, soit la roue droite pour tourner à droite
Les fonction boussole() est expliquée à l'étape 13.

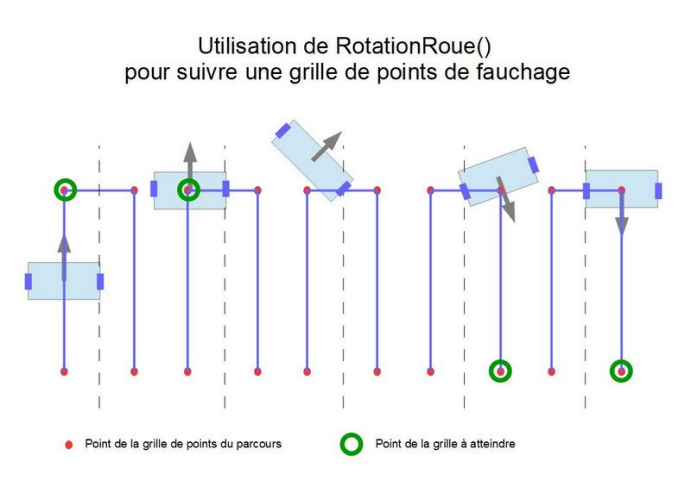
Le virage autour du centre est d'usage général. Il peut cependant être utilisé pour parcourir une grille de fauchage en bandes contigües (figure 2). Tous les points doivent être atteints les uns après les autres, comme le montre la figure 2..

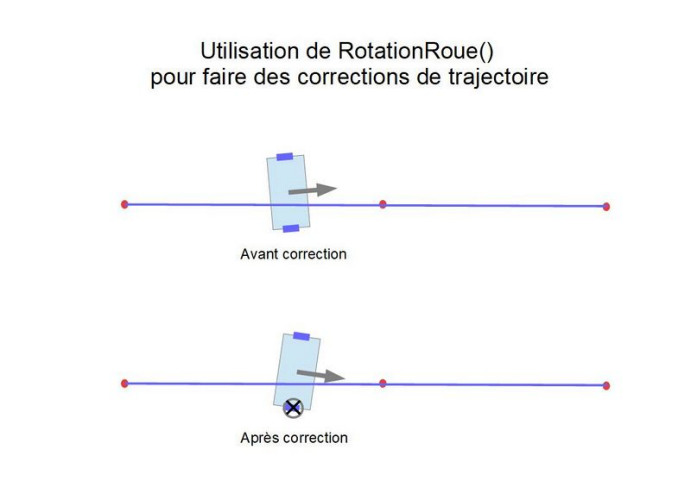
Toutefois le virage autour d'une roue est mieux adapté pour réaliser précisément des bandes contiguës pendant le fauchage (figure 3). Par rapport au virage autour du centre, il s'avère plus précis et plus fluide. Par contre ici l'un des points en bout de ligne est omis, car inutile (N=N+1 dans le programme).

Enfin, le virage autour d'une roue est aussi utilisé pour réaliser les petites corrections angulaires lors du trajet entre deux WP, afin de pointer vers le point intermédiaire situé au devant de la machine (figure 4). Là aussi c'est une correction beaucoup plus fluide que si RotationCentre était utilisée.









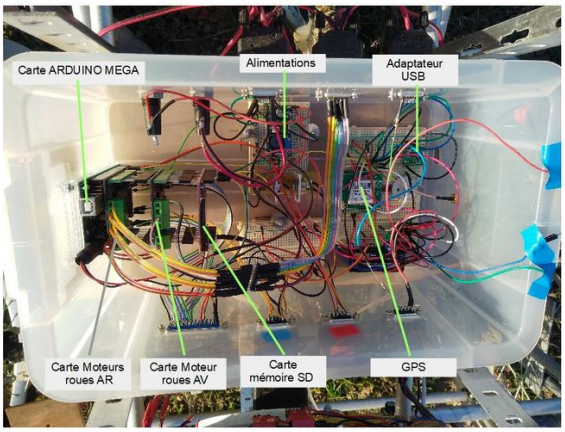
Étape 17 - Partie électrique (1/2)

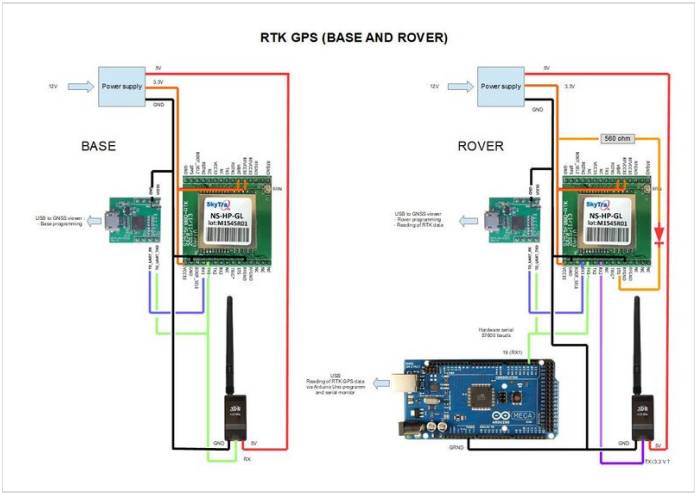
Le boîtier de commande électrique
La photo 1 montre les principales cartes du boîtier de commande qui vont être détaillés ci-dessous.

Câblage du GPS
Les câblages des GPS de la base et de la faucheuse sont donnés sur la figure 2.
On arrive naturellement à ce câblage si on suit la progression de la notice des GPS (voir partie GPS). Dans tous les cas on note la présence de l'adaptateur USB qui permet de programmer les circuits soit en base, soit en rover grâce au logiciel sur PC fourni par NavSpark. Grâce à ce programme, on a également toute les informations de position, nombre de satellites, etc...
Dans la partie faucheuse, la broche Tx1 du GPS est reliée à l'entrée série 19 (Rx1) de la carte ARDUINO MEGA pour recevoir les phrases NMEA (figure 3).
Dans la base la broche Tx1 du GPS est envoyée sur la broche Rx de la radio 3DR pour l'envoi des corrections. Dans la faucheuse les corrections reçues par la radio 3DR sont envoyées sur la broche Rx2 du circuit GPS.

On note que ces corrections et leur gestion sont intégralement assurées par les circuits GPS RTK. Ainsi, la carte Aduino MEGA ne reçoit que des valeurs de position corrigées.

Nota: Le câblage montré ici est celui d'une maquette de labo. Même si ce câblage a permis de tester la machine en conditions réelles avec les vibrations, il est évidemment souhaitable d'en concevoir une version plus durable avec circuits imprimés.





Exemple de phrase NMEA (Emprunt Wikipedia)

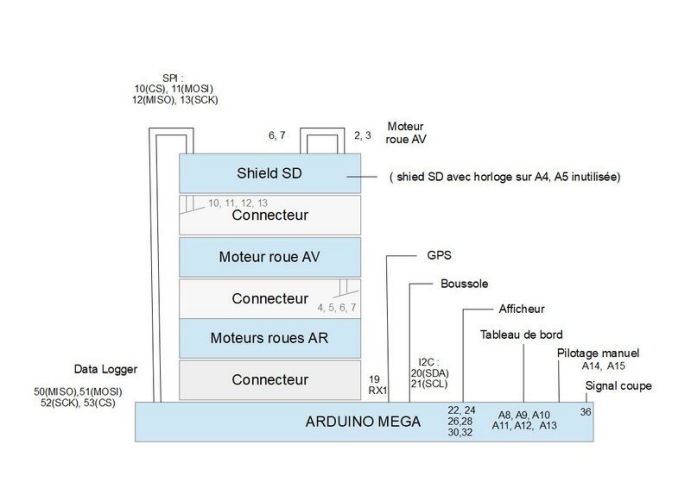
Étape 18 - Partie électrique (2/2)

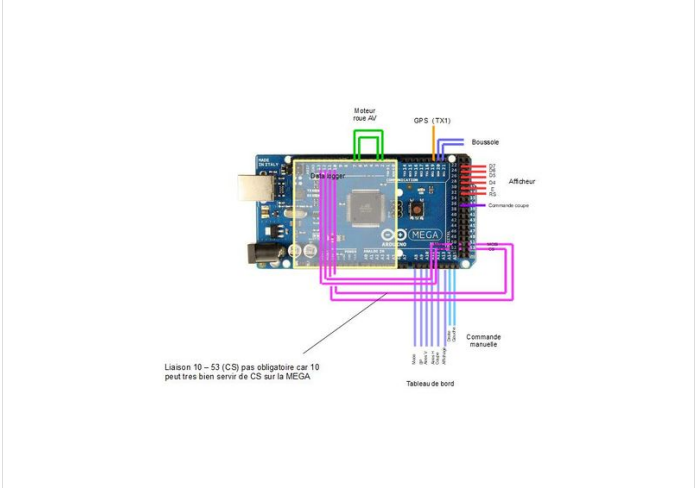
La carte Arduino MEGA et ses shields
- Carte arduino MEGA
- Shield Moteurs roues arrière
- Shield Moteur roue avant
- Shield arte SD

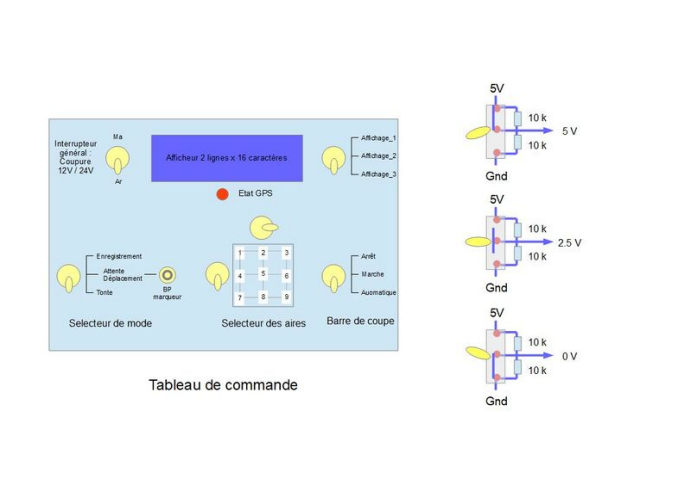
Sur la figure 1, on note que l'on a mis des connecteurs intercalaires entre les cartes pour que la chaleur dissipée dans les cartes moteur puisse s'évacuer. De plus, ces intercalaires permettent de couper des liaisons non désirées entre les cartes, sans avoir à les modifier.

Les figures 2 et 3 montrent comment sont lues les positions des inverseurs du tableau de bord et de la manette de commande.

Et enfin on trouvera le câblage de l'afficheur ici:
Faucheuse_guid_e_par_GPS_RTK_brochage_afficheur_LCM1602C.pdf







Étape 19 - Le programme de pilotage Arduino

La carte microcontrôleur est une Arduino MEGA (UNO n'ayant pas assez de mémoire). Le programme de pilotage est très simple et classique. J'ai développé une fonction pour chaque opération élémentaire à effectuer (lecture tableau de bord, acquisition des données GPS, affichage LCD, commande d'avance ou de rotation de la machine, etc...). Ces fonctions sont ensuite utilisées facilement dans le programme principal. La vitesse lente de la machine (3 mètres / minute) facilite grandement les choses.

Par contre, la barre de coupe n'est pas gérée par ce programme mais par le programme de la carte UNO qui se trouve dans le boîtier spécifique.

Dans la partie SETUP du programme on trouve
- les initialisations de pins utiles de la carte MEGA en entrées ou sorties
- l'initialisation de l'afficheur LCD
- l' initialisation carte mémoire SD
- l' initialisation de la vitesse de transfert de l'interface série hardware vers le GPS
- l' initialisation de la vitesse de transfert de l'interface série vers l'IDE
- l' arrêt des moteurs et de la barre de coupe

Dans la partie LOOP du programme on trouve au début
- les lectures du tableau de bord et de la manette, du GPS, de la boussole et des accéléromètres
- un sélecteur à 3 dérivations, suivant l'état du sélecteur de modes du tableau de bord (manuel, enregistrement, fauchage)

La boucle LOOP est rythmée par la lecture asynchrone du GPS qui est l'étape la plus lente. On revient donc en début de boucle environ toutes les 3 secondes.

Dans la dérivation mode normal, la fonction de déplacement est commandée en fonction de la manette et l'affichage est mis à jour environ toutes les 3 secondes (position, état GPS, direction boussole, inclinaison...). Un appui sur le BP marqueur met à zéro les coordonnées de position qui seront exprimées en mètres dans le repère géographique.

Dans la dérivation mode enregistrement, toutes les positions mesurées pendant le déplacement sont enregistrées sur la carte SD (période d'environ 3 secondes). Lorsque un point d'intérêt est atteint, l'appui sur le marqueur est enregistré. dans la carte SD. La position de la machine est affichée toutes les 3 secondes, en mètres dans le repère géographique centré sur le point origine.

Dans la dérivation mode fauchage: La machine a préalablement été amenée au dessus du point de référence. Au basculement du sélecteur de mode sur "fauchage", le programme observe les sorties GPS et en particulier la valeur du flag d'état. Lorsque le flag d'état passe en "Fix", le programme effectue la mise à zéro de la position. Le premier point à atteindre est alors lu dans le fichier fauchage de la mémoire SD. Quand ce point est atteint, le virage de la machine se fait comme indiqué dans le fichier fauchage, soit autour d'une roue, soit autour du centre des deux roues - Voir les explications du paragraphe 15 -

Le processus se répète jusqu'à ce que le dernier point soit atteint (en général point de départ). A ce moment le programme met la machine et la barre de coupe à l'arrêt.

Le fichier du programme de pilotage SW_Faucheuse_pilotage_V2.ino se trouve ici : Faucheuse guid e par GPS RTK SW Faucheuse pilotage V2.ino
Pas mal de commentaires ont été mis mais il est possible d'en rajouter en fonction des demandes d'explications.
Pour examiner ou éditer le fichier .ino de façon plus agréable qu'avec l'éditeur Arduino on peut utiliser l'éditeur Notepad++ (Notepad++ (notepad-plus-plus.org)).

Ci-dessous un exemple de fichier de fauchage:Faucheuse_guid_e_par_GPS_RTK_ExempleFichierFauchage.txt

Étape 20 - La barre de coupe et sa gestion

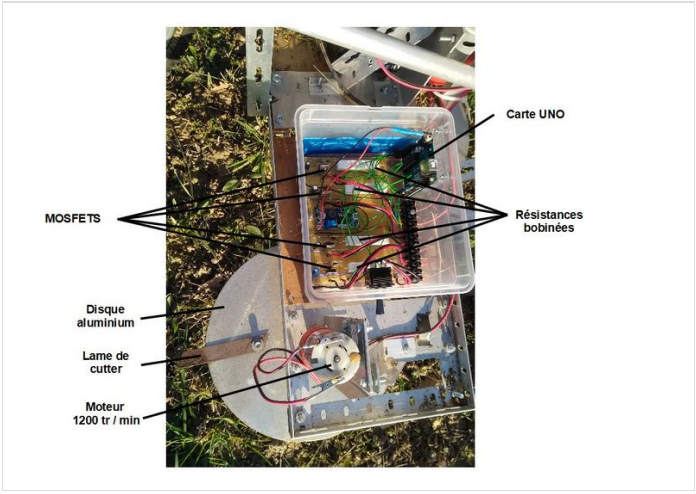
La barre de coupe est constituée par 4 disques tournant à la vitesse de 1200 tr/ minute. Chaque disque est muni de 3 lames de cutter escamotables. Ces disques sont disposés de façon à réaliser une bande de coupe continue de 1.2 mètre de large.

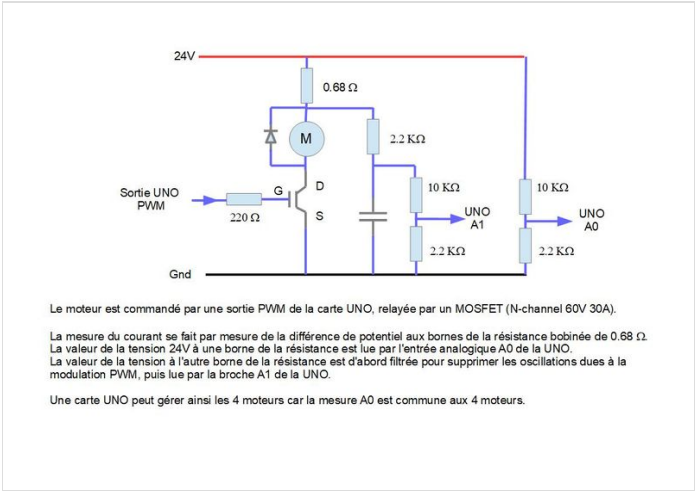
Les moteurs doivent être contrôlés de façon à limiter le courant
- au démarrage, à cause de l'inertie des disques
- lors de la coupe, à cause des blocages par trop d'herbe

Pour cela on mesure le courant dans le circuit de chaque moteur grâce à des résistances bobinées de faible valeur. La carte UNO est câblée et programmée pour mesurer ces courants et envoyer une commande PWM adaptée aux moteurs.

Ainsi, au démarrage, la vitesse augmente progressivement jusqu'à sa valeur maxi en 10 secondes. En cas de blocage par de l'herbe, le moteur s'arrête pendant 10 secondes et refait une tentative de relance pendant 2 secondes. Si le problème persiste, le cycle de 10 secondes de repos et 2 secondes de relance recommence. Dans ces conditions, l'échauffement du moteur reste limité, même en cas de blocage permanent.

Les moteurs se mettent en route ou s'arrêtent lorsque la carte UNO reçoit le signal envoyé par le programme de pilotage. Cependant un interrupteur hard permet de couper le courant de façon fiable pour sécuriser les opérations de maintenance





Étape 21 - Quelle suite donner ? Quelles améliorations ?

Ajouter Galileo à GPS et Glonass

La végétation (arbres) peut limiter le nombre de satellites en vue du véhicule et réduire la précision ou empêcher le verrouillage RTK. On a donc intérêt à utiliser le maximum de satellites simultanément. Il serait donc intéressant de compléter les constellations GPS et Glonass par la constellation Galileo.

On devrait pouvoir bénéficier ainsi de plus de 20 satellites au lieu de 15 au maximum, ce qui permet de mieux s'affranchir de l'écrantage par la couverture végétale.

Il commence à exister des circuits travaillant simultanément avec ces 3 constellations:

- Navspark a sorti un circuit "identique" à celui que j'ai utilisé, mais avec Galilleo en supplément (photo 1) (http://navspark.mybigcommerce.com/ns-hp-gn2-px1122r-multi-band-quad-gnss-rtk-breakout-board/)
- Il existe aussi des shields très compacts (phot 2) qui incluent à la fois le circuit GPS et l'émetteur-récepteur sur le même support https://www.ardusimple.com/product/simplertk2b-starter-kit-mr-ip65/

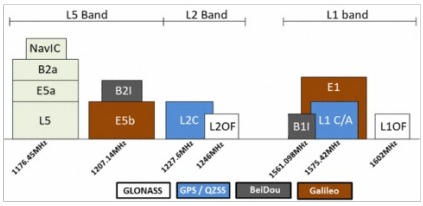
....Mais le prix est bien supérieur à celui des circuits que nous avons utilisés ici.

Compléter le GPS par de l'odométrie et par un LIDAR

Malheureusement, en arboriculture il arrive que la couverture végétale soit très importante (champ de noisetiers par exemple). Dans ce cas, même avec les 3 constellations il se peut que le verrouillage RTK ne soit pas possible.

Il faut donc introduire un capteur qui permettrait de conserver la position même en l'absence momentanée de GPS.

L'utilisation de l'odométrie est une solution simple pour suppléer à la perte du GPS. Son principe consiste à mesurer précisément la rotation des deux roues motrices pour en déduire le trajet de la faucheuse. Malheureusement la précision est limitée par tous les petits glissements des roues par rapport au sol. Je suis en train de tester cette solution en situation de terrain réelle.

L'utilisation d'un LIDAR pourrait également permettre de se repérer dans un verger. Son principe repose sur le balayage du plan horizontal par un faisceau laser émis par une tourelle. Au cours de ce balayage, la distance à tous les objets interceptés est mesurée . Les troncs des arbres sont très faciles à repérer dans ce cas par le faisceau laser et peuvent servir à observer la marche du robot. Voici par exemple un type de LIDAR qui pourrait convenir pour des conditions extérieures (photo3): https://www.robotshop.com/eu/fr/telemetre-laser-360-rplidar-40m.html

Dans les deux cas le GPS reprendrait sa fonction en bout de rangée, au sortir de la couverture végétale.