الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
جامعة سعد دحلب بالبليدة
UNIVERSITE SAAD DAHLEB DE BLIDA

كلية الهندسة - دائرة الإلكترونيك
FACULTE DES SCIENCES DE L'INGENIEUR
DEPARTEMENT D'ELECTRONIQUE

PROJET DE FIN D'ETUDES

En vue de l'obtention du diplôme
Master en électronique

Sur le thème

Commande d'un ballon dirigeable par le microcontrôleur ATtiny2313V

Option : Traitement d'information et système électronique TISE

Proposé et Dirigé par :
Mr. H.SALHI

Présenté par :
Mr. AHFIR M'hammed
Mr. AHMED HADJALLAH Hamza

Septembre 2011

Dédicace

JE DEDIE CE MODESTE TRAVAIL :
À MA TRES CHERE MERE, POUR SES SACRIFICES DEPUIS QU'ELLE M'A MIS AU MONDE, ET QUI N'A PAS CESSE DE M'ENCOURAGER, DE ME SOUTENIR DANS LES MOMENTS DIFFICILES ET QUI A SU M'ENTOURER DE TOUTE SON AFFECTION ET SON AMOUR POUR QUE JE PUISSE REUSSIR.
À MON PERE, QUI M'A TOUJOURS SOUTENU ET AIDE A AFFRONTER LES DIFFICULTES.
À MES FRÈRES ET SŒURS QUI ONT TOUJOURS ÉTÉ À MON COTÉS ET JE LES SOUHAITE LE BONHEUR ET LA RÉUSSITE DANS LEUR VIE.

À TOUS LES MEMBRES DE MA FAMILLE: ONCLES ET COUSINS.
À MON BINOME ET TOUTE SA FAMILLE.
À MES AMIS DE MASTER TISE ET D'AUTOMATIQUE QUI J'AI PARTAGÉ LES PLUS BEAUX MOMENTS DE MA VIE.

Ahmed Hadjallah

HAMZA

Dédicace

Je dédie ce modeste travail tout d'abord à ma mère sans laquelle je ne serais pas ce que je suis.

A mon père qui nous a tant donné sans jamais rien demander.

A mes frères et à mes soeurs qui ont su m'épauler dans les moments difficiles.

À tous les membres de ma famille : oncles et cousins cotés.

À MON BINOME « hamza »et TOUTE SA FAMILLE.

A tous mes amis qui étaient avec moi dés le début de mon parcours « Ahmed, Omar, Yacine ».

Sans oublier mes amis de Master TISE et d'Automatique.

Merci à toutes les personnes qui m'ont aidé de près ou de loin.

AHFIR M'hammed

REMERCIEMENTS

Nous remercions Dieu de nous avoir donné la force et le courage pour réaliser ce modeste travail.

Nous remercions Monsieur SALHI.H qui a suivi de très près ce travail, pour son aide, ses orientations pédagogiques dans l'élaboration de ce mémoire, et tous les conseils qu'il nous a prodigué pendant toute la durée de ce travail.

Nous remercions aussi Monsieur NADJMI.O.

Nos remerciements vont à tous ceux qui nous portent de l'estime et qui nous ont soutenus d'une manière ou d'une autre pour l'élaboration de ce travail.

Enfin, nous ne pourrons terminer ces remerciements sans remercier nos famille pour leurs aides, compréhensions, encouragements et soutiens, qu'elles nous ont apportés tout le long de nos études et à toutes nos amies.

Résumé :

Ce travail porte sur l'étude et la comparaison des techniques de régulation et les implémenter dans un microcontrôleur (ATtiny2313V) pour commander un ballon dirigeable.

Deux types de commandes sont étudiés: une commande PID et une commande par mode glissant.

Mot clés : Dirigeable autonome, commande PID, commande a structure variable (mode glissant), microcontrôleur ATTINY 2313V.

Summary:

This work involves the study and comparison of control techniques and their implemention on a microcontroller (ATtiny2313V) to control an airship. Two types of controls are studied: a PID control and sliding mode control.

Key words: autonomous airship, PID control, variable structure control (sliding mode), microcontroller ATTINY 2313V.

ملخص:

هذا العمل يتضمن دراسة و مقارنة تقنيات التحكم وتنفيذها في متحكم(ATiny2313V) للسيطرة على منطاد.
يتم دراسة نوعين من الضوابط: متحكم PID و طريقة متحكم منزلقة.

كلمات مفتاحيه : المنطاد الذاتي ، التحكم PID ،تحكم هيكل متغير (وضع الانزلاق) ، متحكم ATiny2313.

Table des matières :

Introduction générale :

Les drones ou UAV (Unmanned Aerial Vehicles) sont des engins volants sans pilote capables de mener à bien une mission de façon autonome ou semi-autonome. Leur utilisation est d'abord militaire pour des missions de reconnaissances ou de surveillance. En effet, ils sont bien adaptés pour la réalisation de missions qui mettraient potentiellement un équipage en danger ou qui nécessitent une permanence sur zone. Des applications civiles font leur apparition comme la surveillance du trafic autoroutier, la prévention des feux de forêts, la récolte de données météorologiques ou bien encore l'inspection d'ouvrages d'art. Le développement de plates-formes robotiques volantes connaît un essor croissant depuis quelques années en raison de la miniaturisation toujours plus poussée des capteurs et des actionneurs et surtout grâce à la possibilité d'embarquer des cartes de commande plus performantes capables d'exécuter la masse de calcul nécessaire au contrôle des robots volants.[6]

On peut distinguer deux variétés d'engins volants autonomes :

-Les plus légers que l'air : tels que les dirigeables.

-Les plus lourds que l'air : tels que les drones avions ou les drones hélicoptères. [1]

Figure 1 : Les engins volants autonomes

Pendant des décennies, les chercheurs de tous les domaines s'intéressaient ces systèmes aéronautiques. Plus particulièrement, Les chercheurs automaticiens qui se sont penchés sur le problème de la commande des UAV. En effet, ce sont des systèmes complexes non linéaires multi variables, instables dans certaines plages de vol et présentent une dynamique fortement couplée. L'application des commandes développées en théorie n'est pas toujours systématique. Cependant, les phases de validation dans les conditions réelles s'avèrent nécessaires surtout en l'absence de modèles précis ce qui implique l'utilisation de commandes robustes. On

entend par là, les commandes qui conservent les performances en présence d'erreurs de modélisation et de perturbations extérieures. [7]

Historique des dirigeables : [4] [8]

Les ballons dirigeables ont été les premiers aéronefs de l'humanité, comme leurs aspect fantasmatique continuent d'alimenter des discussions, voyons ce qu'il en est des ballons et dirigeables, et ce qu'il en sera dans l'avenir. Les grandes lignes de cette histoire sont utiles à connaître, notamment pour mettre en perspective le potentiel des ballons dirigeables.

Les premiers essais, en 1783 menèrent à une conception moderne du dirigeable par le générale **Meusnier** dont le ballon étant de forme ellipsoïde mini d'un gouvernail, mais à l'époque aucun moteur n'existait.

C'est en 1852 que l'ingénieur français **Henri Giffard** teste le premier ballon dirigeable, ce dernier ressemble à une montgolfière allongée, il a une enveloppe souple remplie d'hydrogène et propulsé par un moteur à vapeur.

En 1900, on voit apparaître le premier dirigeable à enveloppe rigide (un cadre solide fait en métal et recouvert d'une toile), appelé **Zeppelin**, ils ont étés bien amélioré: ils vont plus vite (130 km/h) et deviennent plus grands (jusqu'à 200 mètres de long). Utilisés principalement comme moyen de transport, ils servent également à des taches militaires : repérer l'ennemi et bombarder les villes.

Peu à peu, les dirigeables sont délaissés suite aux nombreux accidents. Les dirigeables, gonflés à l'hydrogène sont inflammables. Une petite étincelle suffit pour causer une explosion. La catastrophe la plus célèbre : celle du dirigeable Hindenburg. Chargé de 190 millions de litres d'hydrogène, il a explosé près de New York en 1937. Cela a mis un terme à l'utilisation de l'hydrogène. Depuis, on remplit les ballons dirigeables d'hélium, un gaz ininflammable et plus sécuritaire que l'hydrogène.

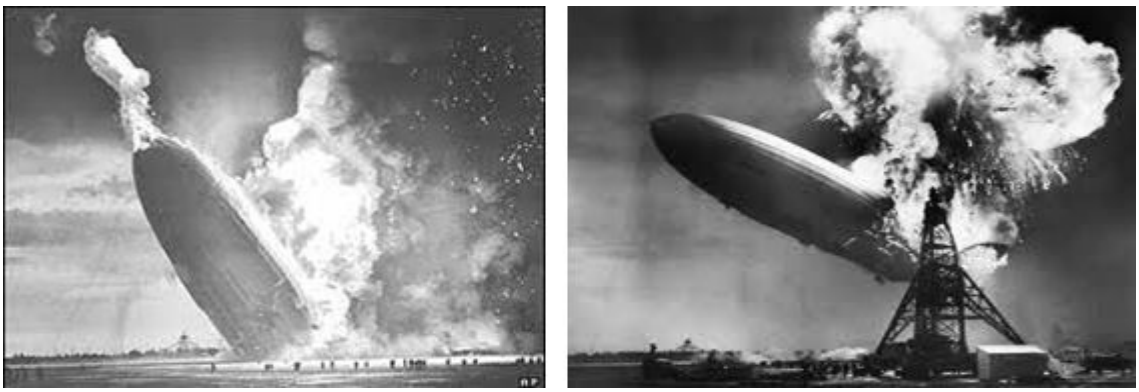


Figure 2 : Incendie du dirigeable Hindenburg.

On trouve trois sortes de ballons dirigeables : souple (c'est grâce au gaz que la forme du dirigeable est maintenue. Celui de Good Year en est un exemple), semi-rigide (sa forme est maintenue grâce à la pression des gaz et à des structures dans le ballon), rigide (il a une structure rigide comme l'Hindenburg).

Les ballons dirigeables font partie des UAV, et depuis une petite dizaine d'années, ils connaissent un certain succès dans la communauté robotique. De nombreux projets de ballons dirigeables autonomes ont récemment vu le jour dans différents laboratoires. Ces engins ont une place un peu à part, et présentant vraisemblablement des intérêts opérationnels dans certains contextes. Ils suscitent beaucoup d'intérêts à cause des missions réalisées, tel que le vol stationnaire et le suivi d'une trajectoire, plusieurs projets sont actuellement en cours afin de développer ces véhicules, parmi eux on cite :

SASS LITE :

C'est le premier projet de ballon autonome ("Small Aerostat Surveillance System, Low Intensity Target Exploitation") mené par la société américaine Applied Research Associates dans un contexte militaire. Le projet a été initié à la fin des années 80, et en 1993 différents ballons de 300 à 600 m^3, munis d'un moteur thermique de 35 chevaux, ont réalisés des déplacements autonomes de plusieurs dizaines de kilomètres, pendant plusieurs heures. Malheureusement aucune information technique sur les lois de commandes utilisées n'est donnée dans les publications.

Aurora :

Ce projet est sans doute le plus connu dans la communauté des roboticiens, et aussi un de ceux qui produit le plus de résultats (ce projet a été présenté en détail aux JNRR 2001). Initié par Alberto Elfes à la fin des années 90 à l'Information Technology Center de Campinas (Brésil), le projet concerne le contrôle autonome du vol d'un ballon, et a d'entrée inclut l'instrumentation et le développement d'un prototype. Les développements matériels et logiciels qui ont consisté à "robotiser" ce ballon y installant les capteurs nécessaires (GPS, compas, gyromètres...), des travaux ont été menés sur la modélisation dynamique et bien entendu sur le contrôle de vol. À notre connaissance, seuls des contrôleurs de type PID pour asservir des paramètres de vol (cap, altitude) et rallier des points de passage ont été effectivement testés en conditions réelles.

Figure 3 : Le ballon dirigeable Aurora.

Lotte :

Ce projet est mené par l'Université de Stuttgart depuis le début des années 90. Il a d'abord consisté à réaliser un ballon d'une longueur de 16 m et de 105 m^3 de volume, et différents développements technologiques associés, notamment l'utilisation de cellules photovoltaïques. De sérieuses études aérodynamiques ont ensuite été menées (modélisation, identification de paramètres en soufflerie, développement de contrôleurs). Malheureusement peu de communications internationales sont publiées autour de ce projet, dont nous ne connaissons aujourd'hui pas précisément l'avancement.

Figure 4 : Le ballon Lotte de l'Université de Stuttgart.

Karma :

Le Laas a commencé à s'intéresser aux ballons dirigeables autonomes en 2000, et a développé un prototype dont le premier vol a eu lieu en Mars 2003. Outre le développement du prototype et son exploitation pour mener des travaux sur la cartographie de l'environnement survolé,

différents travaux relatifs au contrôle de vol ont été menés : modélisation du système et identification des paramètres du modèle, développement de différents contrôleurs, et plus récemment planification de trajectoire. Tous ces travaux ont été évalués en simulation, mais aucun n'a encore débouché sur un contrôle effectif du ballon, qui est toujours télé opéré lors de ses vols.

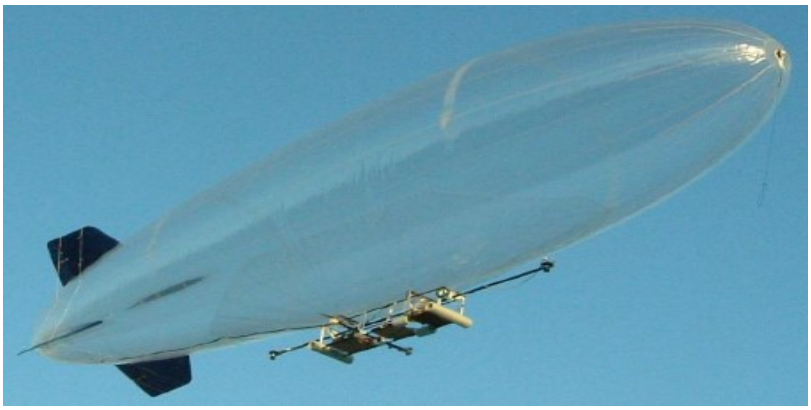


Figure 5 : Le ballon Karma du Laas

Autres projets

De très nombreux autres projets de ballons dirigeables autonomes ont été et sont menés dans d'autres laboratoires et méritent d'être mentionnés : le projet mené au LSC (Laboratoire des Systèmes Complexes), où les travaux portent surtout sur la planification de trajectoires faisables et leur suivi. Le projet Alpha mené entre l'ENST et l'AnimatLab et avec la société Airstar, dans lequel des contrôleurs sont générés par des techniques évolutionnistes. Un projet est mené à l'université Hagen en Allemagne. Deux projets sont en genèse à l'Université des Andes à Bogotta (projet "Uran") et à l'Institut Supérieur de Robotique au Portugal Cette liste est loin d'être exhaustive, et ne mentionne notamment pas les projets de ballons autonomes "indoor" (qui ont été notamment menés au laboratoire GRASP d'Upenn et à Berkeley - projets qui semblent avoir été abandonnés).

L'objet du présent travail est l'implémentation de différentes techniques de commande sur un microcontrôleur AVR (attiny2313) pour commander un ballon dirigeable et comparer les résultats obtenus avec une simulation en utilisant le SIMULINK de MATLAB.

Ce manuscrit s'articule autour de quatre chapitres :

Chapitre I : Nous présentons des généralités et définitions permettant de se familiariser avec les ballons dirigeables et le principe de fonctionnement du microcontrôleur attiny2313.

Chapitre II : il est consacré à la commande du ballon dirigeable en utilisant la méthode classique (PID) et par mode de glissement.

Chapitre III : concerne l'implémentation des lois de commande sur le microcontrôleur attiny2313.

Chapitre IV : Ce chapitre illustre les résultats de simulation de différentes commandes PID et mode de glissement appliqué au système et une comparaison entre ces deux techniques de commande.

Après l'application des différentes commandes, nous avons finalisé notre travail par une conclusion générale donnant une vue complète sur les résultats obtenus. Des perspectives sont ensuite suggérées pour les futurs travaux dans ce domaine.

Pour contrôler un processus dynamique, il est nécessaire de le modéliser.

1.1 : Modélisation du dirigeable :

Dans cette partie, nous présentons une analyse de la physique du système étudiés dans ce travail. Tout d'abord, une brève description du véhicule réel est faite.

Deuxièmement, le modèle non-linéaire du dirigeable.

1.1.1 Description du dirigeable : [9]

Un ballon dirigeable est une enveloppe de mylar de forme ellipsoïde remplie de gaz d'hélium, pour l'ascenseur, il a des ailettes de stabilisation du mouvement et une gondole. La gondole, attachée au fond du sac, contient le récepteur radio, des moteurs, deux propulseurs latéraux. Le propulseur de poupe est placé dans la nageoire inférieure de la queue. L'emplacement des propulseurs est illustré à la figure suivante.

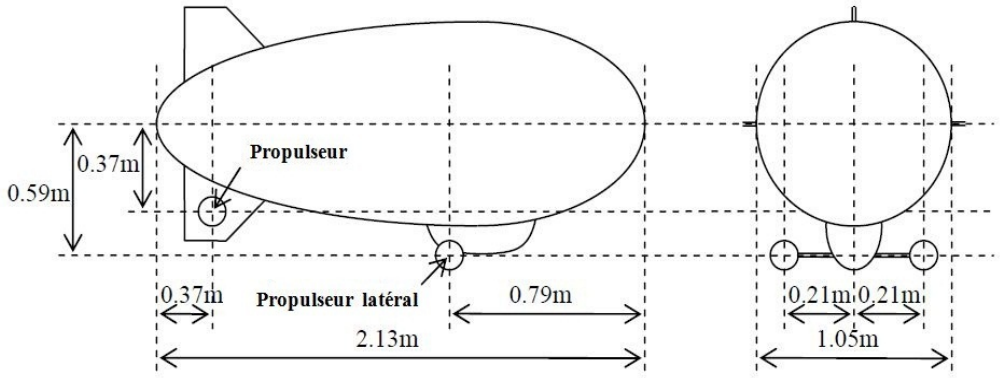


Figure1.1 : dimensions du dirigeable et position des propulseurs [2]

1.1.2 Modèle non linéaire du dirigeable :

Ces engins volants sont assimilés à des corps rigides, cette hypothèse permet de les modéliser de manière simplifiée par la méthode de Newton-Euler. [8]

Les équations du modèle décrivent les lois qui régissent le comportement du ballon dirigeable dans l'espace. Elles modélisent ainsi deux aspects distincts : cinématique et dynamique. [9]

1.1.3 Modèle cinématique : [2]

La configuration du dirigeable est caractérisée par sa position et son orientation par rapport à un repère global. Ce qui nécessite la considération d'un repère local fixé à un point appartenant au dirigeable, généralement il s'agit du centre de gravité ou volumique.

La position et l'orientation du ballon sont définies dans le repère référentiel où la rotation de la terre est négligée ; sa surface est supposée plane.

Les repères qui viennent d'être décrits sont présentés dans la figure suivant :

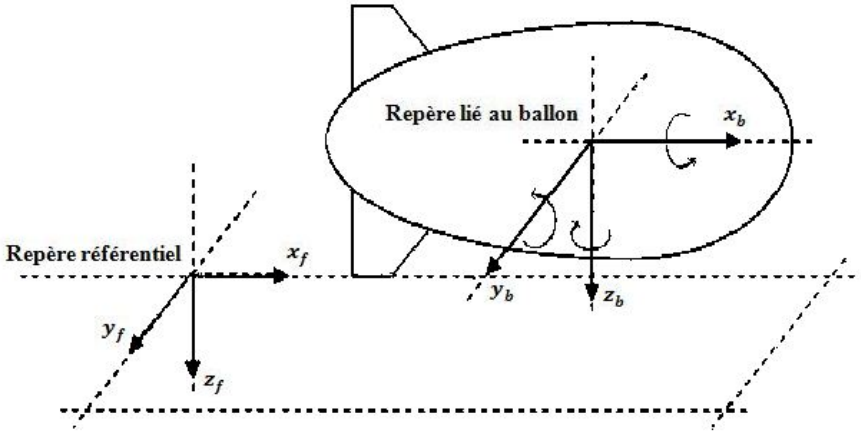


Figure1.2 : représentation des repères

Dans le repère référentiel la position et l'orientation du ballon sont exprimés par le vecteur

$$\eta = [\eta_1, \eta_2]^T$$

Avec $\eta_1 = [x\ y\ z]^T$ représente les trois composantes de position.

$\eta_2 = [\phi\ \theta\ \psi]^T$ représente les trois composantes d'orientation appelées respectivement angle de roulis, de tangage et de lacet.

Dans le repère lié au ballon les vitesses linéaires et angulaires seront définies par le vecteur

$$\nu = [\nu_1, \nu_2]^T$$

Ou $\nu_1 = [\nu_x\ \nu_y\ \nu_z]^T$ représente les trois composantes de vecteur vitesses linéaires.

$\nu_2 = [\omega_x\ \omega_y\ \omega_z]^T$ représente les vitesses angulaires autour des trois axes du repère lié au ballon.

La matrice de rotation qui décrit la transformation entre le repère local et global est donnée par : [2]

$$J_1(\eta_2) = \begin{bmatrix} cos\psi cos\theta & -sin\psi cos\phi + cos\psi sin\theta sin\phi & sin\psi sin\phi + cos\phi cos\psi sin\theta \\ sin\psi cos\theta & cos\psi cos\phi + sin\phi sin\theta sin\psi & -cos\psi sin\phi + sin\theta sin\psi cos\phi \\ -sin\psi & cos\psi cos\phi & cos\theta cos\phi \end{bmatrix} \quad (1.1)$$

Transformation des vitesses :

Les vitesses de translation $\boldsymbol{\dot{\eta}_1}$ et $\boldsymbol{\nu_2}$ du repère local par rapport au repère global exprimées respectivement dans le repère global et local, et les vitesses de rotation $\boldsymbol{\dot{\eta}_2}$ et $\boldsymbol{\nu_2}$ du repère local par rapport au repère global exprimées respectivement dans le repère global et local. Les relations reliant les différentes vitesses sont données par :

$$\dot{\eta}_1 = J_1(\eta_2)v_1 \tag{1.2}$$

$$\dot{\eta}_2 = J_2(\eta_2)v_2 \tag{1.3}$$

Ou $$J_2(\eta_2) = \begin{bmatrix} 1 & sin\phi tg\theta & cos\phi tg\theta \\ 0 & cos\phi & -sin\phi \\ 0 & sin\phi/cos\theta & cos\phi/cos\theta \end{bmatrix} \quad ; \text{Avec } \theta \neq \frac{\pi}{2} + k\pi. \tag{1.4}$$

À partir des équations (1.2) et (1.3), l'expression cinématique du dirigeable est donnée par:

$$\dot{\eta} = J(\eta_2)v \quad \Leftrightarrow \quad \begin{bmatrix} \dot{\eta}_1 \\ \dot{\eta}_2 \end{bmatrix} = \begin{bmatrix} J_1(\eta_2) & [0]_{3\times3} \\ [0]_{3\times3} & J_2(\eta_2) \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \tag{1.5}$$

L'intégration de l'équation $\dot{\eta} = J(\eta)v$ nous donne la position et l'orientation véhicule volant à chaque instant dans le repère inertiel.

1.1.4 Dynamique de dirigeable :

Contrairement aux avions qui se basent totalement sur la vitesse relative de leurs ailes par rapport au vent, les dirigeables utilisent, tout comme les sous marins, la sustentation archimédienne afin développer une grande partie de leur portance. Cette sustentation provient du gaz enfermé par l'enveloppe et qui doit être plus léger que l'air. En général le gaz utilisé est l'Hélium qui est rare. Lors du mouvement du dirigeable, et vu son volume, l'air déplacé par le dirigeable est important. La masse volumique du dirigeable est très proche de celle du fluide, ce qui fait apparaître des termes additifs de masses et inerties ajoutées (appelées aussi masses et inerties virtuelles). [5]

Hypothèses : [5]

• Les déformations de l'enveloppe sont négligeables, afin de permettre d'appliquer la théorie de la mécanique des solides.

• L'invariance de la masse de dirigeable par rapport à la consommation du combustible et par rapport au déplacement des particules de gaz à l'intérieur de l'enveloppe.

• Toutes les perturbations extérieures sont nulles (vent, pression, température. . . etc).

• Le centre de flottabilité est supposé confondu avec le centre volumique.

Modèle Dynamique : [2] [5]

Le modèle dynamique du dirigeable comme étant un corps rigide est obtenu par l'application du formalisme de Newton-Euler, les termes additifs de masses et inerties ajoutées sont modélisées par la théorie de Kirchhoff. Cette dynamique est donnée généralement dans le repère local pour des raisons de simplicité de son expression, et qui prend la forme suivante :

$$\boldsymbol{M\dot{v} + C(v)v + D(v)v + g(\eta) = \tau} \tag{1.6}$$

$\boldsymbol{M = M_{RB} + M_A}$: Matrice 6x6 contenant les termes de masse et d'inertie.

$\boldsymbol{C(v) = C_{RB}(v) + C_A(v)}$: Matrice 6x6 des termes de forces de Coriolis et de centrifuges

$\boldsymbol{D(v)}$: Matrice contenant des termes de forces d'amortissements.

$\boldsymbol{g(\eta)}$: Matrice englobe les termes des forces hydrostatiques.

$\boldsymbol{\tau}$: Forces et couples des propulseurs.

$\boldsymbol{v}$: Torseur cinématique du ballon dirigeable.

Nous avons, pour l'inertie coefficients de la matrice:

$$I = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{yx} & I_{yy} & -I_{yz} \\ -I_{zx} & -I_{zy} & I_{zz} \end{bmatrix} = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \tag{1.7}$$

La matrice d'inertie du système, c'est une matrice symétrique qui s'exprime ainsi :

$$M_{RB} = \begin{bmatrix} m[I]_{3\times3} & [0]_{3\times3} \\ [0]_{3\times3} & I \end{bmatrix} \tag{1.8}$$

m : masse du ballon.

La matrice d'inertie ajoutée dont les termes sont constants est :

$$M_A = \begin{bmatrix} a_{11} & 0 & 0 & 0 & 0 & 0 \\ 0 & a_{22} & 0 & 0 & 0 & 0 \\ 0 & 0 & a_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & a_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & a_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & a_{66} \end{bmatrix} \tag{1.9}$$

La matrice des forces de Coriolis et des forces centrifuges qui s'appliquent au ballon dirigeable. C'est matrice 6x6 antisymétrique et d'expression :

$$C_{RB} = \begin{bmatrix} 0 & 0 & 0 & 0 & mv_z & -mv_y \\ 0 & 0 & 0 & -mv_z & 0 & mv_x \\ 0 & 0 & 0 & mv_y & mv_x & 0 \\ 0 & mv_z & -mv_y & 0 & I_{zz}w_z & -I_{yy}w_y \\ -mv_z & 0 & mv_x & -I_{zz}w_z & 0 & I_{xx}w_x \\ mv_y & -mv_x & 0 & I_{yy}w_y & -I_{xx}w_x & 0 \end{bmatrix} \quad (1.10)$$

La matrice de forces de Coriolis et de forces centrifuges hydrodynamique et peut se mettre sous la forme :

$$C_A = \begin{bmatrix} 0 & 0 & 0 & 0 & a_{33}v_z & -a_{22}v_y \\ 0 & 0 & 0 & -a_{33}v_z & 0 & a_{11}v_x \\ 0 & 0 & 0 & a_{22}v_y & -a_{11}v_x & 0 \\ 0 & a_{33}v_z & -a_{22}v_y & 0 & a_{66}w_z & -a_{55}w_y \\ -a_{33}v_z & 0 & a_{11}v_x & -a_{66}w_z & 0 & a_{44}w_x \\ a_{22}v_y & -a_{11}v_X & 0 & a_{55}w_y & -a_{44}w_x & 0 \end{bmatrix} \quad (1.11)$$

La matrice d'amortissement est donnée par

$$D = diag\{D_{v_x} + D_{v_xv_x}|v_x|, D_{v_y} + D_{v_yv_y}|v_y|, D_{v_z} + D_{v_zv_z}|v_z|, D_{w_x} + D_{w_xw_x}|w_x|, D_{w_y} + D_{w_yw_y}|w_y|, D_{w_z} + D_{w_zw_z}|w_z|\} \quad (1.12)$$

Les coefficients d'amortissements sont déterminés expérimentalement.

Les efforts hydrostatiques, poussée d'Archimède et force de pesanteur est donnée par :

$$g(\eta) = \begin{bmatrix} (W-B)sin\theta \\ -(W-B)cos\theta sin\emptyset \\ -(W-B)cos\theta cos\emptyset \\ -(y_GW - y_BB)cos\theta cos\emptyset + (z_GW - z_BB)cos\theta sin\emptyset \\ (z_GW - z_BB)sin\theta + (x_GW - x_BB)cos\theta cos\emptyset \\ -(x_GW - x_BB)cos\theta sin\emptyset - (y_GW - y_BB)sin\theta \end{bmatrix} \quad (1.13)$$

Avec $W = \|F_G\|$ et $B = \|F_B\|$

la force de pesanteur est exprimée par $F_G = mg$ ou m est la masse du solide

la poussée d'Archimède est égale à la force opposée au poids que subit le volume du fluide :

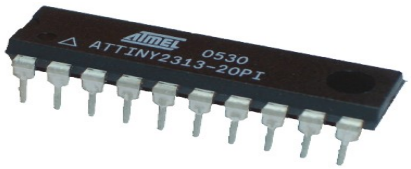
$$\boldsymbol{F_B = \rho V_f g} \quad (1.14)$$

ρ : La densité du fluide.

V_f : Le volume du fluide déplacé.

g : L'accélération gravitationnelle.

1.2 : Le microcontrôleur attiny2313 : [17] [18]



1.2.1 Introduction :

Les microcontrôleurs sont des ordinateurs à usage spécialisé. Ils exécutent des taches spécifiques d'une façon excellente. Il y a un certain nombre d'autres caractéristiques communes qui définissent les microcontrôleurs. Si un ordinateur comporte la majorité de ces caractéristiques, alors on peut l'appeler « microcontrôleur » :

- Les microcontrôleurs sont embarqués à l'intérieur d'autre dispositif, de sorte qu'ils puissent commander des dispositifs ou des actions du produit.
- Les microcontrôleurs sont consacrés à une tache et exécutent un programme spécifique. Le programme est stocké dans la mémoire morte (ROM).
- Les microcontrôleurs sont souvent des dispositifs de basse (faible) puissance.
- Les microcontrôleurs a un dispositif d'entrée spécifique et souvent (mais pas toujours) à une petite voyant LED ou afficheur LCD pour sortie.
- Le microcontrôleur est souvent petit et à prix réduit. Les composants sont choisis pour réduire au minimum la taille et pour être aussi peu couteux que possible.

Il existe plusieurs familles de microcontrôleurs, on cite :

- **Atmel :** c'est le plus grand fabricant des microcontrôleurs en Europe, leur produits principaux sont : famille 8051, AT91SAM, et **AVR.**
- **MOTOROLA Semi-conductor :** c'est un principal fabricant de microcontrôleurs, leur produit principaux sont :8HC05/08/11/12/16, 68300,MPC, M*Core, Flash.
- **Texas Instrument :** Malgré, bien connu par leur DSP, TI fait également un microcontrôleurs de très base puissance appelé le MSP430.
- **Microship :** C'est une société Américaine,elle a mis au point dans les années 90 un microcontrôleur CMOS : le PIC (**P**eriphiral **I**nterface **C**ontrôler).

1.2.2 Le kit de formation NA001 :

Le NA001 est un kit de formation construit autour du microcontrôleur attiny2313, ces caractéristiques principales sont :

- Microcontrôleur ATTINY2313 d'ATMEL

- Compilateur : GCC-gnu ou CV-avr ou IAR
- Facile à programmer et à corriger
- Facile à connecter par interface au PC ou lap top en utilisant la port parallèle LPT
- Facile d'extension et de connections à d'autres systèmes
- 8 lampes (LEDs) programmables pour la vérification
- 2 boutons poussoir externes programmables
- Dimension compacte : 6cm*9cm
- Facile à utilisée

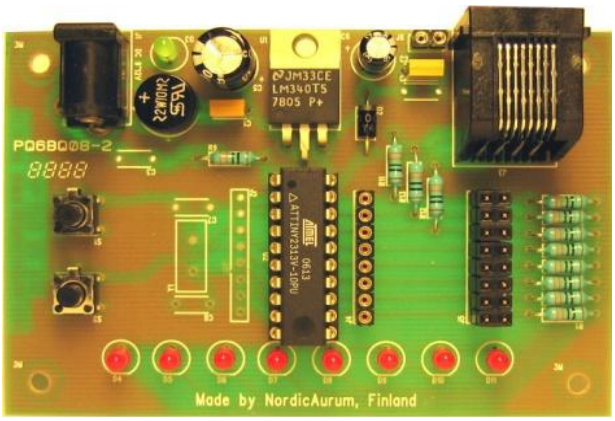


Figure 1.3 : le kit de formation NA001.

1.2.3 Présentation de attiny2313 :

Les AVR sont des microcontrôleurs (µC) 8bits RISC (Reduced Instruction Set Computer) ce qui signifie « calculateur à jeu réduit d'instructions ».les µC d'ATMEL sont très performants (architecture de Harvard), dont la famille est très étendue.

Attiny2313 est un microcontrôleur CMOS de 8-bits de basse puissance basé sur l'architecture AVR RISC.

1.2.3.1 Caractéristiques de attiny2313 :

Il fonctionne à 10-20 Mhz maximum. Il possède :

- 120 instructions (composant RISC),
- 2Ko de mémoire Flash pour le programme,
- 128 octets de RAM,
- 128 octets de d'Eprom,
- 1 compteurs/ timers de 8 bits .
- 1 compteurs/ timers de 16 bits.
- sources d'interruption externes et internes.

- 18 entrées/sorties configurables individuellement.
- Mode SLEEP.

1.2.3.2 Architecture interne :

La figure suivante présente l'architecture interne de attiny2313 :

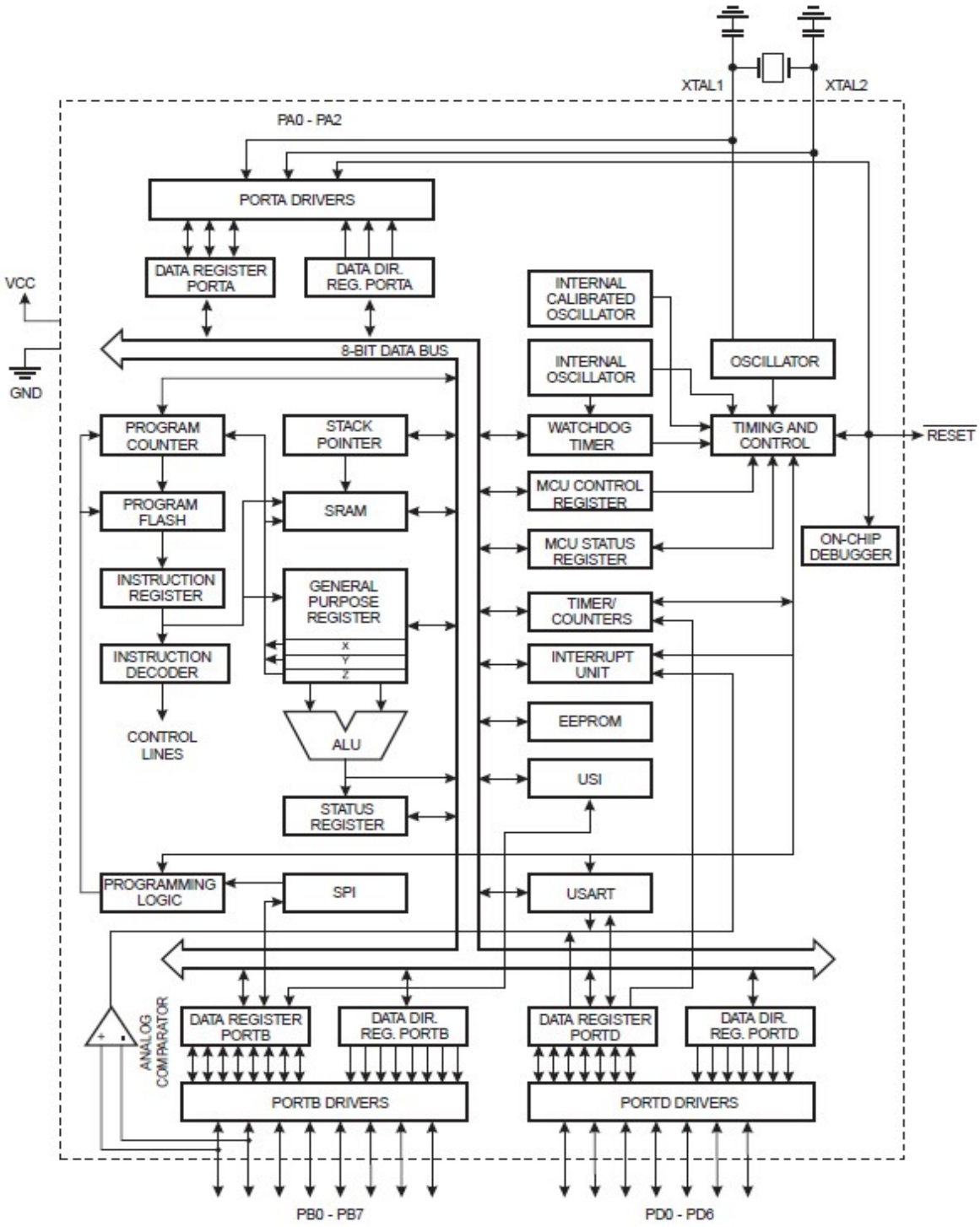


Figure 1.4 : Architecture interne de attiny2313.

1.2.3.3 Brochage de attiny2313 :

Ce microcontrôleur se présente sous la forme d'un boîtier PDIP/SOIC, il est schématisé dans la figure :

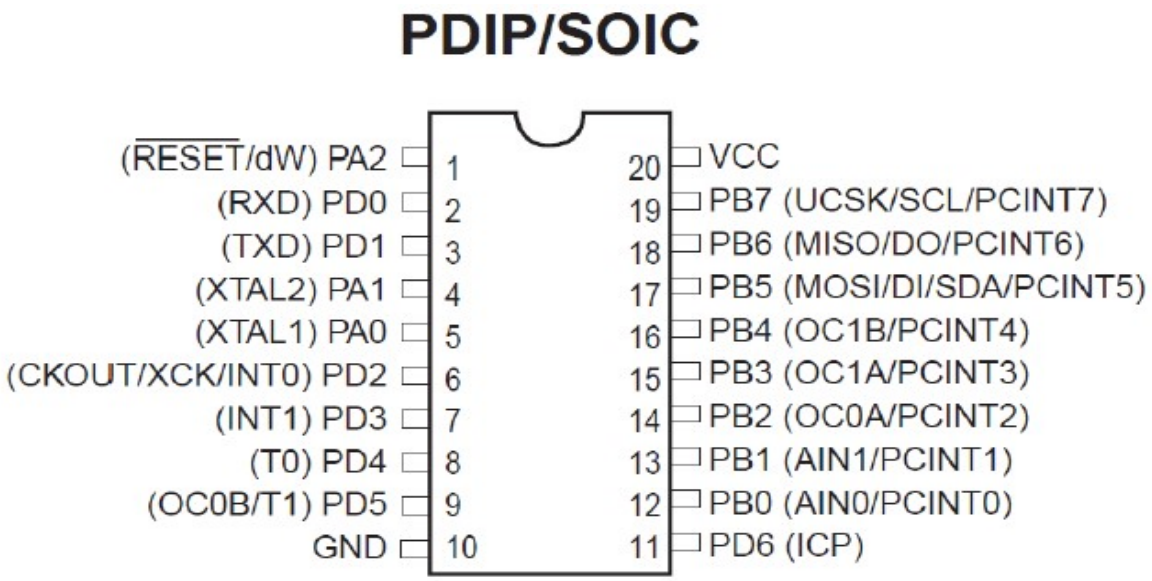


Figure 1.5 : brochage du microcontrôleur attiny2313.

L'attiny2313 possède 3 groupes de porte E/S, porte A, porte B et porte D.

- **Porte A (PA2…PA0)** a seulement 3 bornes bidirectionnelles.
- **Porte B (PB7…PB0)** a 8 bornes bidirectionnelles.
- **Porte D (PD6…PD0)** a 7 bornes bidirectionnelles.

V_{cc} et **GND** sont les pattes d'alimentation. Cette dernière doit être comprise entre 1.8 et 5.5 V.

Toutes les bornes du microcontrôleur Attinny2313 ont des résistances internes de Pull-up.

1.2.3.4 Circuit équivalent de la borne d'E/S :

Chaque borne d'entrée/sortie est équipée d'une résistance programmable Pull up et deux diodes D1 et D2 de protection contre Vcc et la masse comme représentée sur la figure :

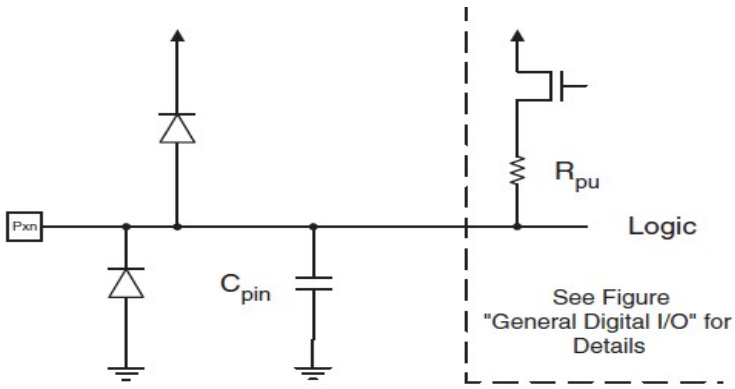


Figure1.6 : circuit équivalent de la borne d'E/S

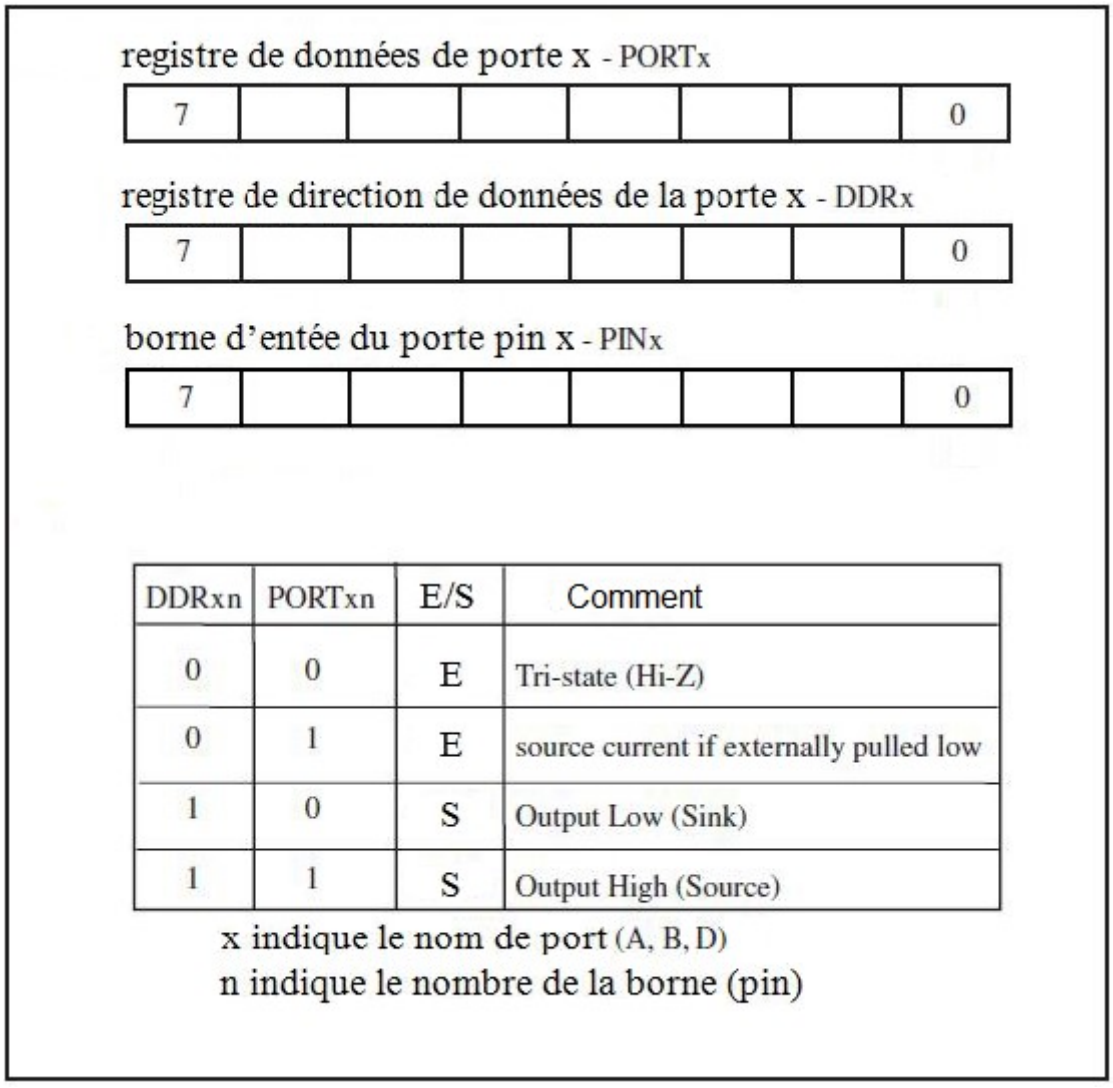
1.2.3.5 Configuration des ports d'entres/sortie :

Chaque porte peut être programmée en tant qu'entrée, sortie ou en tant que fonction spécial RX, TX, INT0, etc.

Pour programmer les bornes (pins), chaque borne est associée à 3 registres :

- Le registre de direction de données de la porte x (R/w) : **DDRx**
- Le registre de données de porte x (R/w) : **PORTx**
- La borne d'entée du porte pin x (R) : **PINx**

x indique le nom de port et il peut être A, B, ou D (exemple : DDRA, PORTA, PINA).



DDRxn	PORTxn	E/S	Comment
0	0	E	Tri-state (Hi-Z)
0	1	E	source current if externally pulled low
1	0	S	Output Low (Sink)
1	1	S	Output High (Source)

x indique le nom de port (A, B, D)
n indique le nombre de la borne (pin)

Figure 1.7 : Configuration des registres associés aux ports.

Conclusion :

Au début de ce chapitre nous nous sommes intéressés à la présentation et la modélisation du ballon dirigeable.

La partie modélisation est sans doute la partie la plus difficile dans l'élaboration d'une loi de commande pour le dirigeable. Elle fait principalement appel à des notions de mécanique des fluides.

Nous avons présenté les principales équations établissant le comportement du ballon dirigeable dans l'espace. De là nous avons tiré un modèle cinématique et un modèle dynamique pour le ballon dirigeable.

En utilisant ces deux modèles obtenus, nous établirons dans les chapitres suivants des techniques de commande qui nous permettront d'asservir la trajectoire du dirigeable.

Dans une deuxième partie, nous avons présenté le microcontrôleur Attiny2313 et le kit de formation NA001, qu'il sera utilisé pour implémenter les techniques de la régulation.

2.1 Linéarisation des équations dynamiques : [1] [10]

2.1.1 Introduction :

Dans le chapitre précédent, une description générale de la modélisation d'un dirigeable autonome a été donnée. Nous avons pu constater que ce modèle est non linéaire et complexe. De plus, le calcul des paramètres aérodynamiques est souvent fait à partir d'équations empiriques.

De ce fait, nous proposons pour avoir un modèle simplifié du dirigeable, de découpler en plan longitudinal et latéral, à fin d'obtenir une modélisation dans chaque plan. Cette modélisation est basée sur la linéarisation générale et sur des hypothèses simplificatrices. [1]

2.1.2 Découplage : [1] [11]

Du fait de la symétrie de forme du dirigeable par rapport à son plan vertical, on sent intuitivement qu'une évolution dans un plan vertical ne peut induire aucune force aérodynamique dans le plan horizontal, ni de couple autour de l'axe longitudinal Ox (le centre de gravité étant dans le plan de symétrie). Il en résulte que les mouvements dans le plan vertical ne conduisent à aucun couplage ni dans le plan latéral ni autour de l'axe longitudinal.

Par contre, les mouvements dans le plan latéral (xOy) conduisent à des forces de couplage dans le plan longitudinal (xOz) et autour de l'axe Ox.

Les caractéristiques et les modes de dirigeable qui ont été étudiés fournissent des informations précieuses sur le comportement des dirigeables. Dans cette section, le modèle présenté précédemment est linéarisé. L'influence des différentes forces sur les modes du dirigeable est étudiée pour obtenir une meilleure compréhension du comportement du dirigeable et les ajustements sont faits pour régler le comportement latéral du modèle.

- **Linéarisation des équations de mouvement:** [9]

A fin d'obtenir un modèle linéarisé du dirigeable, il est nécessaire de définir une valeur d'équilibre soit pour la vitesse ou la position :

Elles sont définies comme suite :

$$\eta_0(t) = [x_0(t)\ y_0(t)\ z_0(t)\ \emptyset_0(t)\ \ \theta_0(t)\ \Psi_0(t)]$$

$$v_0(t) = [v_{x0}(t)\ v_{y0}(t)\ v_{z0}(t)\ w_{x0}(t)\ \ w_{y0}(t)\ w_{z0}(t)] \quad (2.1)$$

L'erreur autour de ces valeurs :

$$\Delta\eta(t) = \eta(t) - \eta_0(t) \qquad \Delta v(t) = v(t) - v_0(t) \quad (2.2)$$

Donc : on peut linéariser l'équation de la dynamique du dirigeable (1.6) :

$$M\Delta\dot{v} + \frac{\partial C(v)v}{\partial}\Big|_{v_0}\Delta v + \frac{\partial D(v)v}{\partial}\Big|_{v_0}\Delta v + \frac{\partial g(\eta)}{\partial}\Big|_{\eta_0}\Delta\eta = \tau \quad (2.3)$$

Et pour l'équation cinématique nous pouvons appliquer les erreurs $\Delta\eta$ et Δv, et on néglige les termes du seconde ordre :

$\dot{\eta}_0 + \Delta\dot{\eta} = J\,(\eta_0 + \Delta\eta)(v_0 + \Delta v)$

$\Delta\dot{\eta} = J\,(v_0 + \Delta\eta)(v_0 + \Delta v) - \dot{\eta}_0$

$\Delta\dot{\eta} = J\,(\eta_0 + \Delta\eta)(v_0 + \Delta v) - J(\eta_0)v_0$ (2.4)

$\Delta\dot{\eta} = J\,(\eta_0 + \Delta\eta)v_0 + J\,(\eta_0 + \Delta\eta)\Delta v - J(\eta_0)v_0$

$\Delta\dot{\eta} \simeq J\,(\eta_0 + \Delta\eta)\Delta v + (\,J\,(\eta_0 + \Delta\eta) - J(\eta_0)\,)v_0$

On pose $x_1 = \Delta v$, $x_2 = \Delta\eta$ et on a $\tau = Bu$

On obtient le modèle suivant :

$$\begin{cases} M\,x_1 + C(\dot{t})x_1 + G(t)x_2 = Bu \\ \dot{x}_2 = J(t)x_1 + J^*(t)x_2 \end{cases} \quad (2.5)$$

Avec : $C(t) = \frac{\partial C(v)}{\partial v}\Big|_{v_0(t)}$ $D(t) = \frac{\partial D(v)}{\partial v}\Big|_{v_0(t)}$ $G(t) = \frac{\partial g(\eta)}{\partial \eta}\Big|_{\eta_0(t)}$

$$J(t) = J(\eta_0(t)) \quad ; \quad J^*(t) = (J\big(\eta_0(t) + \Delta\eta(t)\big) - J(\eta_0(t))$$

Enfin, on calcule les différentes matrices, la représentation d'état du modèle sera :

$$\begin{bmatrix} M\dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -(C(t) + D(t) & -G(t) \\ J(t) & J^*(t) \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\begin{bmatrix} B \\ 0 \end{bmatrix}u \quad (2.6)$$

Par conséquent :

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} M^{-1}(C(t) + D(t)) & M^{-1}G(t) \\ J(t) & J^*(t) \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} M^{-1}B \\ 0 \end{bmatrix}u \quad (2.7)$$

$$\Leftrightarrow \quad \dot{x} = Ax + B'u$$

2.1.3 Stabilisation des modèles du dirigeable : [1] [15] [16]

La stabilisation d'un système linéaire est caractérisé par des valeurs propres (pôles) de la matrice dynamique A :

$$\det(pI - A) = 0 \quad (2.8)$$

L'analyse de stabilité a eu comme conséquence une distribution des pôles des modèles longitudinaux et latéraux linéarisés.

La figure suivante montre l'évolution des pôles des modèles linéarisés : [17]

Figure 2.1 : caractéristiques des endroits de pôles de la dynamique longitudinal et latéral.

2.1.3.1 Stabilisation du modèle longitudinal :

Le mode de montée (*surge mode*) : Il est décrit par un véritable pôle d'une grande constante de temps et est associée à la vitesse d'avancement. L'influence de l'augmentation de la vitesse dans ce mode est à peine perceptible, l'autre pôle réel est associé à la vitesse de descente.

Le mode de poussée (*heave mode*) : Comme la vitesse augmente, le mode devient plus rapide et se développe en un affaissement de hauteur. Ce mode est décrit par le plus rapide des deux pôles réels.

Le mode pendule longitudinal (*longitudinal pendulum mode*) : Il correspond à la paire de pôles complexes qui est associée à l'angle de tangage et la vitesse de tangage.

Dans l'état stationnaire l'amortissement est nul et la propriété d'oscillation du pendule dans ce mode devient évident. Comme la vitesse augmente la fréquence diminue.

Tous les modes sont stables (à la marge pour *air-stationnaire*) sur l'enveloppe de vol.

2.1.3.2 Stabilisation du modèle latéral :

Le pair de pôles qui caractérise le mode d'oscillation roulis (*Roll Oscillation mode*) sont liées à la vitesse de roulis et l'angle de roulis. Ce mode oscillatoire de roulis est équivalent à un mouvement du pendule légèrement amorti. Avec l'augmentation de la vitesse, la stabilité générale s'améliore, puisque le taux d'amortissement de mode oscillatoire et les valeurs propres devenues plus négative. Toutefois, contrairement à ce qui se passe dans le cas longitudinal, la fréquence du mode oscillatoire reste pratiquement inchangée tout au long de la plage de vitesse et que le facteur

d'amortissement subit une augmentation, qui à son tour semble être directement proportionnelle à la variation de vitesse.

Se référant aux deux pôles réels, le mode lent, habituellement appelé mode de descente glissé (*sideslip subsidence mode*), est associé à la vitesse de glissement.

Le mode rapide est le mode de descent lacet (*yaw subsidence*) liés à la vitesse de lacet, il présente une constante de temps qui diminue avec la vitesse.

Comme dans le cas longitudinal, tous les modes sont stables sur l'enveloppe de vol.

2.1.4 Modèles rapprochés : [10] [14]

2.1.4.1 Modèle longitudinal :

Dans le cas longitudinal, le vecteur d'état est : $x = [v_z\ w_y\ \theta\ z\ ]^T$ et le vecteur d'entrée est donnée par $u = [\delta_e\ T_r\ ]^T$ où toutes les variables représentent les variations autour de la valeur d'équilibre. Par conséquent, l'équation dynamique longitudinale est donnée par:

$$\begin{bmatrix} \dot{v}_z \\ \dot{w}_y \\ \dot{\theta} \\ \dot{z} \end{bmatrix} = A_{long} \begin{bmatrix} v_z \\ w_y \\ \theta \\ z \end{bmatrix} + B_{long} \begin{bmatrix} \delta_e \\ T_r \end{bmatrix} \tag{2.9}$$

z , v_z sont l'altitude et la vitesse linéaire.

w_y , θ sont la vitesse et l'angle de tangage respectivement.

δ_e et T_r sont les commandes d'altitude et de poussée respectivement.

On prend en considération seulement la commande d'altitude δ_e.

$$\begin{bmatrix} \dot{v}_z \\ \dot{w}_y \\ \dot{\theta} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & a_3 & 0 \\ a_4 & a_5 & a_6 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & a_7 & 0 \end{bmatrix} \begin{bmatrix} v_z \\ w_y \\ \theta \\ z \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ 0 \\ 0 \end{bmatrix} \delta_e \tag{2.10}$$

a_1, a_4 sont les forces et les moments liés à la vitesse de descente.

a_2, a_5 sont les forces et les moments liés à la vitesse de tangage.

a_1, a_1 sont les forces et les moments liés à l'angle de tangage.

a_7 est la vitesse de déplacement donnée pour le ballon dirigeable.

b_1, b_2 sont les forces et les moments liés à la commande.

2.1.4.2 Modèle latéral :

Dans le cas latéral, le vecteur d'état est : $x = [v_y\ w_z\ \psi\ ]^T$ et le vecteur d'entrée est donné par $u = [\delta_e\ T_r\ ]^T$ où toutes les variables représentent les variations autour de la valeur d'équilibre. Par conséquent, l'équation dynamique latéral est donnée par:

$$\begin{bmatrix} \dot{v}_y \\ \dot{w}_z \\ \dot{\psi} \end{bmatrix} = A_{lat} \begin{bmatrix} v_y \\ w_z \\ \psi \end{bmatrix} + B_{lat} \begin{bmatrix} \delta_r \\ T_r \end{bmatrix} \tag{2.11}$$

v_y est la vitesse de glissement (vitesse linéaire).

w_z est la vitesse de descente (vitesse angulaire).

ψ est l'angle de lacet.

δ_r est la commande de direction.

$$\begin{bmatrix} \dot{v}_y \\ \dot{w}_z \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} a_8 & a_9 & 0 \\ a_{10} & a_{11} & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} v_y \\ w_z \\ \psi \end{bmatrix} + \begin{bmatrix} b_3 \\ b_4 \\ 0 \end{bmatrix} \delta_r \tag{2.12}$$

a_8, a_{10} sont les forces et les moments liés à la vitesse de glissement.

a_9, a_{11} sont les forces et les moments liés à la vitesse de lacet.

b_3, b_4 sont les forces et les moments liés à la commande.

2.2 Contrôle d'un ballon dirigeable autonome:

2.2.1 Introduction :

L'objectif de cette partie est de présenter et synthétiser des lois de commande afin de commander le système étudié. Ce dernier est découplé précédemment en deux sous-systèmes : longitudinal et latéral.

Avant l'application d'une technique de commande à un système complexe il faut la choisir en s'appuyant sur les critères suivant : robustesse et réalisabilité de la commande, temps de calcul et organisation du régulateur. Nous abordons d'abord une commande linéaire(PID) et une commande non linéaire (commande à structure variable).

2.2.2 Technique de commande linéaire :

[13]

Ces techniques ont l'avantage d'être simple à mettre en œuvre. Leur principe de fonctionnement consiste à exprimer la commande du système en fonction des erreurs des états que l'on souhaite commander.

Le régulateur PID assure une correction égale à la somme de l'action proportionnelle P, de l'action intégrale I et de l'action dérivée D.

- Le terme proportionnel P assure la rapidité.
- Le terme intégral I annule l'erreur statique.
- Le terme dérivé D améliore la stabilité.

Soit :

$$U(t) = k_p\, e(t) + k_i \int_0^t e(t)dt + k_d \frac{de(t)}{dt} \qquad (2.13)$$

Ou : e (t) est l'erreur de régulation.

U (t) est le signal de commande délivré par le régulateur.

On en tire la fonction de transfert du correcteur PID :

$$U(s) = Kp(1 + \frac{1}{T_i\, s} + s\, T_d) \qquad (2.14)$$

- **Schéma de régulation PD :**

La méthodologie de conception qu'on va utiliser dans cette section a été proposée par M. JALVING [1], la technique est composée de deux lois de commande, un PD pour commander l'angle $\boldsymbol{\theta}$ et un autre Proportionnelle (P), afin de commander z (voir figure suivante).

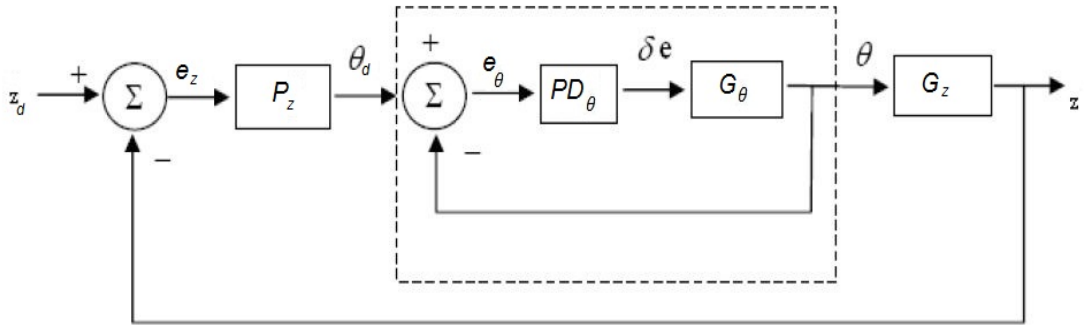


Figure 2.2 : Diagramme de la commande PD en mode longitudinale [1].

Avec :

z_d et θ_d : les consignes.

$e_z\ et\ e_\theta$: les erreurs de z et θ.

$P_z\ et\ PD_\theta$: les régulateurs P et PD pour z et θ.

$G_z\ et\ G_\theta$: les fonctions de transfert en z et θ, ces derniers sont obtenus par l'application de la transformation de Laplace au modèle (2.10) .

La commande est exprimée par l'expression suivante :

$$\delta_L = G_1(z - z_d) + G_2\theta + \ G_3 q \tag{2.15}$$

Les constantes *G2* et *G*3, sont calculées pour avoir une dynamique de boucle fermée désirée. *G*1 est choisi pour garantir que la boucle d'angle $\boldsymbol{\theta}$ soit plus rapide que la boucle d'altitude $\boldsymbol{z}$.

- **Fonction de transfert en Tangage θ :** elle fait la relation entre l'angle θ et l'entrée de la commande δe : on a

$$G_\theta = C_\theta^T(s\,I - {A_{long}}^{-1})B_{long} \tag{2.16}$$

$$G_z = C_z^T(s\,I - {A_{long}}^{-1})B_{long} \tag{2.17}$$

$C_\theta^T\ et\ C_z^T$ sont les coefficients des vecteurs mesures :

$$C_\theta^T = [0\ 1\ 0] \quad , C_z^T = [0\ 0\ 1] \tag{2.18}$$

Les matrices sont définies d'après l'équation (2.10)

$$\begin{bmatrix} \dot{w}_y \\ \dot{\theta} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} a_5 & a_6 & 0 \\ 1 & 0 & 0 \\ 0 & u_0 & 0 \end{bmatrix} \begin{bmatrix} w_y \\ \theta \\ z \end{bmatrix} + \begin{bmatrix} b_2 \\ 0 \\ 0 \end{bmatrix} \delta_e \tag{2.19}$$

Les fonctions de transfert sont :

$$G_\theta = \frac{b_1}{s^2 - a_5 s - a_6} \qquad \text{et} \qquad G_z = \frac{u_0}{s}\frac{b_1}{s^2 - a_5 s - a_6} = \frac{u_0}{s} G_\theta \tag{2.20}$$

$$\text{D'où} \qquad PD_\theta = K_p^\theta + K_d^\theta s \qquad \text{et} \quad P_z = K_p^z \tag{2.21}$$

Les fonctions de transfert des boucles fermées sont :

$$BF_\theta = \frac{(K_p^\theta + K_d^\theta s)b_2}{s^2 + (b_2 K_d^\theta - a_5)s + b_2 K_p^\theta - a_6} \tag{2.22}$$

Et

$$BF_z = \frac{u_0 K_p^z}{s + u_0 K_p^z} \tag{2.23}$$

Par la comparaison des coefficients du dénominateur de l'équation (2.22) avec le polynôme ${s^2 + 2\zeta w_n\, s + {w_n}^2}$ on obtient :

$$K_p^\theta = \frac{{w_n}^2 + a_6}{b_2}$$

$$K_d^\theta = \frac{2\zeta w_n + a_5}{b_2} \tag{2.24}$$

Si on prend la constante du temps égale à 6 secondes pour la boucle d'altitude z, on aura : [1]

$$K_p^z = \frac{1}{6u_0} \tag{2.25}$$

Dans une deuxième section, on va concevoir une loi PD afin obtenir une direction désirée, elle prévoit une marge de phase additionnelle et augmente ainsi la robustesse.

La loi de contrôle ne contient pas l'action intégrale. La raison principale d'omettre l'action intégrale est que le relais de gouvernail de direction à une marche arrêt non linéaire qui causerait des cycles de vibration si l'action intégrale est ajoutée.[1]

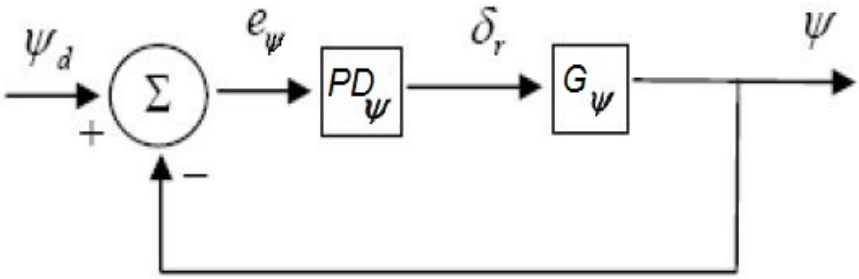


Figure 2.3 : Diagramme de la commande PD en mode latérale.

ψ_d: la consignes.

e_ψ : l'erreurs de ψ.

PD_ψ : le régulateur PD pour ψ.

Les sous états de directions sont : $x_{direction} = [w_z \ \psi]^T$

Les matrices sont définies d'après l'équation (2.12) :

$$\begin{bmatrix} \dot{w}_z \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} a_{11} & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} w_z \\ \psi \end{bmatrix} + \begin{bmatrix} b_4 \\ 0 \end{bmatrix} \delta_r \tag{2.26}$$

La fonction de transfert est :

$G_\psi = C_\psi^T (S\,I - A_{lat}) B_{lat}$ avec : $C_\psi^T = [0\ 1]$

Donc : $G_\psi = \frac{b_4}{s(s - a_{11})}$

La fonction de transfert en boucle fermée est :

$$BF_\psi = \frac{\left(K_p^\psi + K_d^\psi s\right) b_4}{s^2 + \left(b_4 K_d^\psi - a_{11}\right) s + b_4 K_p^\psi} \tag{2.27}$$

Par la comparaison des coefficients du dénominateur de l'équation (2.27) avec le polynôme $s^2 + 2\zeta w_n\, s + {w_n}^2$ on obtient :

$$K_p^{\psi} = \frac{{w_n}^2}{b_4}$$

(2.28)

$$K_d^{\psi} = \frac{2\zeta w_n + a_{11}}{b_4}$$

La loi de commande du contrôleur PD est : $PD_{\psi} = K_p^{\psi} + K_d^{\psi}\, s$

2.2.3 Technique de commande non linéaire : [7] [11] [13]

Les lois de commande classique tel que PID donnent de bons résultats dans le cas des systèmes linéaires à paramètres constants. Pour des systèmes non linéaires ou ayant des paramètres non constants, ces lois de commande classique peuvent être insuffisantes car elles sont non robustes. Les lois de commande dite à structure variable constituent une bonne solution à ces problèmes. La commande à structure variable (CSV) est par nature une commande non linéaire.

La caractéristique principale des systèmes à structure variable est que leur loi de commande se modifie d'une manière discontinue.

Les commutations de la commande s'effectuent en fonction des variables d'état, utilisées pour créer une "variété" ou "surface" dite de glissement dont le but est de forcer la dynamique du système de correspondre avec celle définie par l'équation de la surface. Quand l'état est maintenu sur cette surface, le système est dit en régime glissant. Ainsi tant que les conditions de glissement sont assurées, la dynamique du système reste insensible aux variations des paramètres du processus, aux erreurs de modélisation (dans une gamme qui reste plus large par rapport à celle des approches classiques de l'automatique), et à certaines perturbations.

Ce mode de glissement est souvent qualifié d'idéal du fait qu'il requiert pour exister, une fréquence de commutation infiniment grande. De fait, tout système de commande comprend des imperfections telles que retards, hystérésis, qui imposent une fréquence de commutation finie. La trajectoire d'état oscille alors dans un voisinage de la surface de glissement, phénomène appelé **chattering** ou **broutement**.

Ce type de systèmes a en général deux modes de fonctionnement :

- Le mode non glissant, (reaching mode) ou encore mode d'accès.
- Le mode glissant, (sliding mode).

- **Le mode glissant :**

L'étude du régime glissant d'un système de commande à structure variable sous-entend la définition et l'étude de problèmes particuliers tels que :

- Choix de la surface de glissement.
- Existence et unicité des solutions en régime glissant (commande équivalente).
- Invariance du régime glissant vis à vis d'incertitudes paramétriques et/ou perturbations.

Soit $\tilde{x} = x - x_d$ l'erreur de suivi d'une variable d'état$x$.

Nous pouvons définir la variation de surface $s(t)$ dans l'espace d'état par une équation scalaire $\boldsymbol{\sigma}[\tilde{\boldsymbol{x}}(\boldsymbol{t})] = \mathbf{0}$. La surface de glissement est definie de telle sorte que l'erreur $\tilde{x}$ converge vers zéro.

Une surface de glissement est définie pour les systèmes à une seule entrée et une seule sortie par:

$$s = \dot{\tilde{x}} + \lambda\tilde{x} = 0 \tag{2.29}$$

Où $\lambda > 0$ est la bande passante de pilotage. Sur la surface de glissement, nous avons:

$$\dot{\tilde{x}} + \lambda\tilde{x} = 0 \quad \Rightarrow \quad \tilde{x}(t) = \exp\left[-\lambda(t - t_0)\tilde{x}(t_0)\right] \tag{2.30}$$

La dynamique de l'exponentielle (2.30) assure que l'erreur $\tilde{x}$ converge vers zéro en temps fini pour tout initial $\tilde{x}(t_0)$ .Cette définition de la surface de glissement est bonne pour les systèmes ayant une seule sortie et une seule entrée.

Pour un système multi-état couplé, une autre définition de la surface de glissement est utilisée, basée sur des variables d'état, plutôt que des erreurs de sortie:

$$\sigma(\tilde{X}) = S^T\tilde{X} = 0 \tag{2.31}$$

Où $\tilde{X} = X - X_d$ est un vecteur des erreurs de variables d'état souhaités, et S est un vecteur de coefficients connus pour être conçu sur le modèle linéaire du système. Les coefficients dans le vecteur S déterminent complètement la surface de glissement. Cette définition de la surface de glissement assure la convergence de l'état d'erreur vers zéro. La discussion qui suit et la conception SMC sont basées sur cette dernière définition.

Les modèles linéarisés décrivant le mouvement en mode Longitudinale et Latéral-Directionnel du Ballon Dirigeable peuvent être mis sous la forme : [1]

$$\dot{X} = AX + Bu \tag{2.32}$$

L'objectif de contrôle en vient maintenant à trouver une loi sur le contrôle de conduire $\sigma(\tilde{X})$ vers zéro basé sur le modèle (2.32).

Par la définition de la fonction de Lyapunov :

$$v(\sigma) = \frac{1}{2}[\sigma]^2 \tag{2.33}$$

Nous pouvons garantir que la surface de glissement $\sigma(\tilde{X}) = 0$ est atteint en un temps fini par l'équation (2.34) pour s'assurer que : $\dot{V} = \sigma\dot{\sigma} \leq 0$

$$\sigma\dot{\sigma} = -\eta^2|\sigma| \quad \text{ou} \quad \dot{\sigma} = -\eta^2 sgn(\sigma) \tag{2.34}$$

Où η^2 est un paramètre ajustable. D'après l'équation. (2.31), nous avons:

$$\begin{aligned}\dot{\sigma}(\tilde{X}) &= S^T\dot{\tilde{X}} \\ &= S^T(\dot{X} - \dot{X}_d) \\ &= S^T(AX + Bu) - S^T\dot{X}_d \end{aligned} \tag{2.35}$$

Soit (2.34) égale à (2.35), nous avons:

$$S^T(AX + Bu) - S^T\dot{X}_d = -\eta^2 sgn(\sigma) \tag{2.36}$$

Par conséquent, nous pouvons justifier la fonction de (2.34) en choisissant une entrée de commande comme suit:

$$\begin{aligned}u &= u_{eq} + u_{dis} \\ &= (S^TB)^{-1}(-S^TAX) + (S^TB)^{-1}(S^T\dot{X}_d - \eta^2 sgn(\sigma))\end{aligned} \tag{2.37}$$

La loi de commande u de (2.37) se compose de deux parties. La première partie u_{eq} est une loi de contrôle à rétroaction linéaire basé sur le modèle nominal de l'équation. (2.32) avec un vecteur de gain $K^T = (S^TB)^{-1}B^TA$ calculé à partir de placement de pôles.

Le deuxième partie u_{dis} est un asservissement non linéaire avec le terme de commutation $\eta^2 sgn(\sigma)$ son signe entre plus et moins, selon le côté du plan de glissement de l'erreur.

La dernière partie est pour la conduite et le maintien du système sur la surface de glissement $\sigma(\tilde{X}) = 0$. Si η est choisit assez grand, u_{dis} peut fournir robustesse nécessaire aux effets des perturbations instantanées et de la dynamique non modélisé.

Cependant, la fonction discontinue $sign$ à la loi de commande peut conduire à un contrôle bavardage. Ce bavardage implique une haute activité de contrôle, et en outre, il peut exciter la dynamique à haute fréquence négligé dans la modélisation.

Par conséquent, il est souhaitable de lisser l'activité de contrôle près de la surface de glissement. Dans la pratique, nous pouvons modifier la loi de commande par mode de glissement en remplaçant la fonction *sign* avec une fonction continue de $tanh$:

$$u = u_{eq} + u_{dis}$$

$$= K^T X + (S^T B)^{-1}(S^T \dot{X}_d - \eta^2 tanh(\sigma/\emptyset)) \quad (2.38)$$

Où $\emptyset$ est l'épaisseur de la couche limite de la surface de glissement.

D'après $K^T = (S^T B)^{-1} B^T A$, nous avons:

$$S^T(A - K^T B) = 0 \qquad A - K^T B = A_c \quad (2.39)$$

Par conséquent, S^T est un annulateur gauche de A_c, ou S est un vecteur propre droit de A_c^T correspondant à une valeur propre nulle.

Avec ce choix de S, on peut déterminer complètement la surface de glissement $S^T \widetilde{X} = 0$ et la loi de contrôle par mode de glissement (2.38).

En résumé, la procédure de conception de la commande en mode glissant est constituée de deux étapes: d'abord, de déterminer la loi d'asservissement u_{eq} sur la base du modèle nominal, deuxièmement, la conception d'une loi de contrôle u_{dis} pour compenser les écarts de performances idéales en raison des incertitudes.

2.3 Détermination des lois de commande :

Nous présentons les équations du mouvement du ballon dirigeable, les valeurs numériques des paramètres du ballon dirigeable sont montrées dans l'Annexe B.

2.3.1 Mode longitudinal :

D'après (2.10), nous présentant les équations du mouvement en mode longitudinal :

$$\begin{bmatrix} \dot{v}_z \\ \dot{w}_y \\ \dot{\theta} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} -0.39 & 3.937 & 0.015 & 0 \\ 0.108 & -0.620 & -0.137 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -9.5 & 0 \end{bmatrix} \begin{bmatrix} v_z \\ w_y \\ \theta \\ z \end{bmatrix} + \begin{bmatrix} -0.242 \\ -0.430 \\ 0 \\ 0 \end{bmatrix} \delta_e$$

2.3.1.1 Commande linéaire :

La commande est exprimée par l'expression (2.15).

D'après les équations (2.24) (2.25) :

$K_p = -0.74$ et $K_d = -1.78$ pour la régulation de l'angle θ .

$K_p = -0.14$ pour la régulation de l'altitude z .

Avec $\zeta = 0.8$ $\tau = 1$ $w_n = 0.25$

2.3.1.2 Commande non linéaire : [1]

en choisissant les pôles p=[0 , -0.5 , -0.6 , -0.7]

le vecteur K est calculée en utilisant matlab : k= place (A, B,p)

k=[-0.2219 -1.7369 -0.8322 0]

la matrice A_c est donnée par :

$$A_c = A - Bk = \begin{bmatrix} -0.4175 & 3.6846 & -0.1032 & 0 \\ 0.0026 & -1.3825 & -0.5023 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -9.5 & 0 \end{bmatrix}$$

La surface de glissement est obtenu en résolvant l'équation (2.31) : $A_C^T S = 0$.

Alors S = [1 -8.5186 -15.4615 0.4396]

Où $S = 1v_z - 8.5186\, w_y - 15.4615\ \theta + 0.4396\ (z - z_d)$

On choisit ϕ = 0.4, la loi de commande final est :

$\delta_e = k * [v_z\ \ w_y\ \ \theta\ \ z]^T + \tanh(\frac{S}{\phi})$

$$\delta_e = -0.2219\, v_z - 1.7369\, w_y - 0.8322\, \theta + \text{sign}(\frac{S}{0.4})$$

2.3.2 Mode latéral :

D'après (2.12), nous présentant les équations du mouvement en mode latéral :

$$\begin{bmatrix} \dot{v}_y \\ \dot{w}_z \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} -0.416 & -3.985 & 0 \\ -0.131 & -0.584 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} v_y \\ w_z \\ \psi \end{bmatrix} + \begin{bmatrix} 1.455 \\ -0.495 \\ 0 \end{bmatrix} \delta_r \qquad (2.40)$$

2.3.2.1 Commande linéaire :

D'après les équations (2.24) (2.25) :

$K_p = -0.1263$ et $K_d = 0.3717$ pour la régulation de l'angle ψ .

2.3.2.2 Commande non linéaire : [1]

en choisissant les pôles p=[0 , -0.25 , -0.26]

le vecteur K est calculée en utilisant matlab : k= place (A, B,p)

k=[3.5 10.44 0]

la matrice A_c est donnée par :

$$A_c = A - Bk = \begin{bmatrix} -5.2386 & -19.1749 & 0 \\ 1.6015 & 4.5836 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

La surface de glissement est obtenue en résolvant l'équation : $A_C^T S = 0$.

Alors S = [1 3.4 3.35]

Où $S = 1v_y + 3.4\, w_z + 3.35\ (\psi - \psi_d)$

On choisit ϕ = 0.4, la loi de commande final est :

$\delta_e = k * [v_y \ \ w_z \ \ \psi]^T + 2.5\tanh\left(\frac{S}{\phi}\right)$

$$\delta_e = 3.5v_y + 10.44\, w_z + 2.5\ \text{sign}\left(\frac{S}{0.4}\right)$$

2.4 Conclusion :

Dans ce chapitre, nous avons vu le découplage du modèle mathématique du dirigeable en deux modes : longitudinal et latéral. Pour chaque mode, on a étudié la linéarisation et la stabilisation du modèle du dirigeable.

Nous avons montré les notions de la technique de commande classique PID, qui est une technique de commande linéaire qui permet une convergence acceptable vers la consigne.

Nous avons vu la commande par mode de glissement, qui est une commande non linéaire robuste. Elle consiste à amener le système à une surface dite de glissement, sur laquelle les objectifs de commande sont réalisables, et le maintenir sur cette surface. Nous avons vu aussi le plus grand problème de cette commande, qui est le phénomène de chattering, et les solutions proposées pour l'éliminer.

Les résultats de simulation de ces lois sont présentés dans les chapitres suivants.

3.1 Introduction :

Après la détermination et la simulation des lois de commande, on va les implémenter dans notre microcontrôleur. Pour tester nos programmes, on utilisera le MatLab pour créer un environnement équivalent au système étudié et on connecte la carte à microcontrôleur avec le PC en utilisant une communication série par la liaison RS232.
Le travail à faire est résumé par l'organigramme suivant :

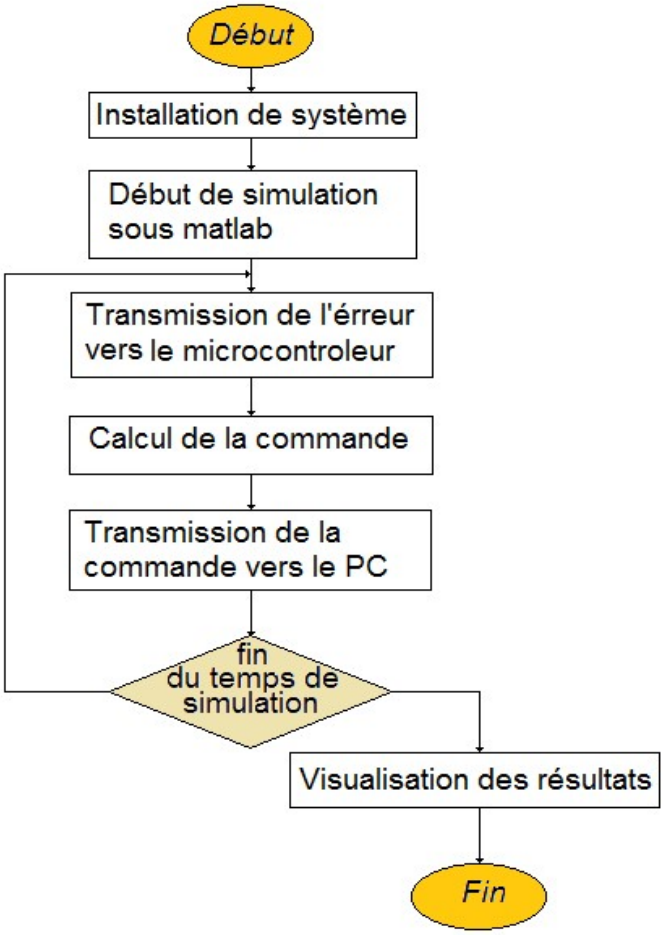


Figure 3.1 : diagramme de déroulement de travail.

3.2 La liaison RS-232: [19]

RS-232 est une norme qui standardise un port de communication de type série.

Il est disponible sur presque tous les PC même s'il a tendance à être remplacée pas la porte USB. Malgré cela, il reste encore très utilisé dans l'industrie notamment, grâce à sa robustesse et a sa simplicité.

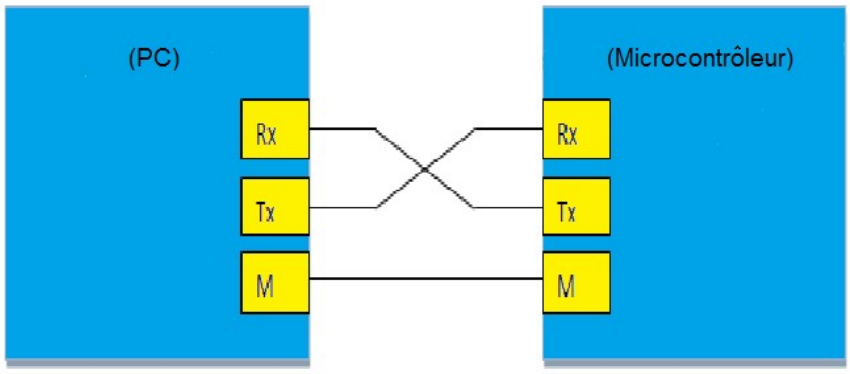


Figure 3.2 : Principe d'une liaison série (RS232).

Sur la liaison série, les bits d'information (1 ou 0) arrivent successivement, à des intervalles réguliers.

L'octet à transmettre est envoyé bit par bit (en commençant par le bit de poids faible) par l'émetteur sur la ligne Tx du PC, vers la ligne Rx du microcontrôleur.

Ce type de liaison est en fait une liaison asynchrone car il n y a pas d'horloge commune entre l'émetteur et le récepteur.

Du coup, pour que la communication puisse fonctionner correctement, des bits supplémentaires sont indispensables :

- Bit de début, placé au début de la trame.
- Bit de fin, placé à la fin de la trame.

La liaison filaire de type RS-232 est utilisée pour plusieurs raisons :

- Simplicité d'utilisation au niveau logiciel : standard très répandu.
- Cout modique : un circuit intégré et un câble.
- Mise en œuvre matérielle aisée : un simple composant logique.

Pour réaliser la liaison série entre le PC et le kit utilisé, nous avons utilisé un MAX232 de MAXIM (voir l'annexe).

Il sert d'interface électrique entre une liaison série TTL (0-5V) et une liaison série RS232 (+12 -12V) et ce avec une simple alimentation 5V.

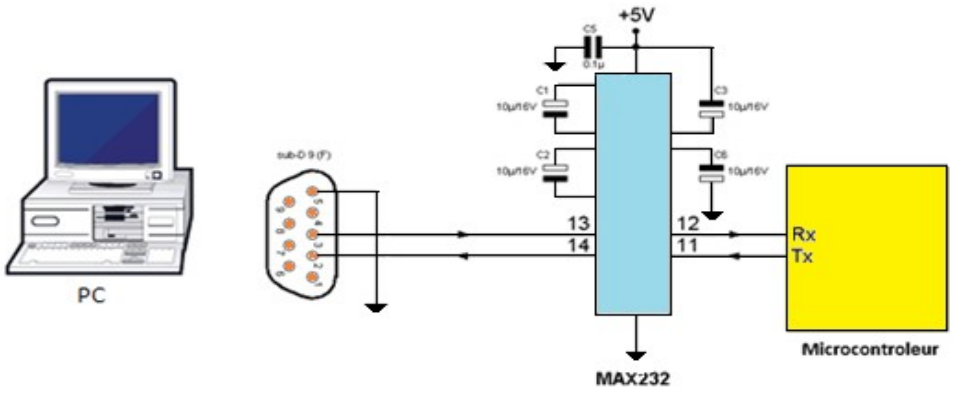


Figure 3.3 : interfaçage électrique de liaison RS232.

Ce montage est réalisé en utilisant le kit de formation NA001, une carte d'interface qui contienne le max232 et le câble de connexion série.

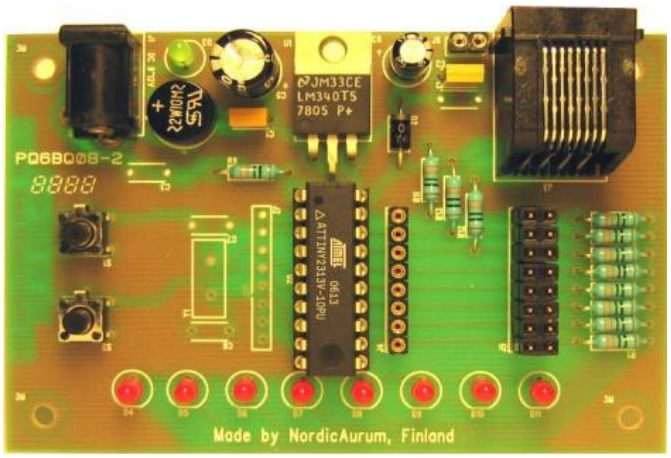


Figure 3. 4 : le kit de formation NA001.

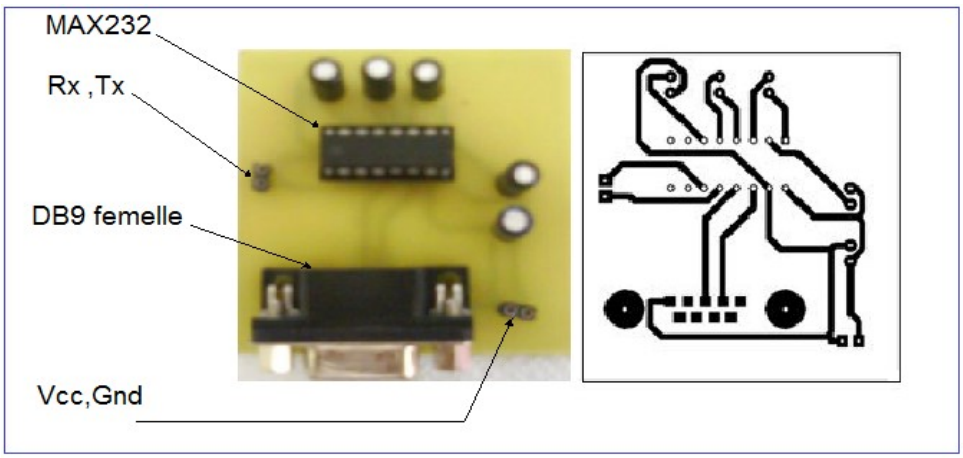


Figure 3.5 : Photo de l'implémentation du module de communication Max232.

3.2.1 Configuration de la communication série :

3.2.1.1 Le microcontrôleur : [19]

Pour le microcontrôleur, c'est le module USART (Universal Synchronous Asynchronous Receiver Transmitter) qui permet d'envoyer et de recevoir des données en mode série, soit de façon synchrone, soit asynchrone.

Le module USART de l'attiny2313 est accessible par le biais des broches (RXD) PD0 et (TXD) PD1. Sachant qu'une liaison série synchrone nécessite une ligne dédicacée à l'horloge, il ne reste donc qu'une seule ligne pour transmettre les données. On en déduit que le microcontrôleur ne pourra émettre et recevoir en même temps en utilisant l'USART en mode synchrone ; On parlera donc de liaison « half-duplex ».

Par contre, le mode asynchrone n'a pas besoin de ligne d'horloge, il nous restera alors 2 lignes pour communiquer, chacune étant dédicacée à un sens de transfert. Nous pourrons donc envoyer et recevoir des données en même temps ; On parlera de liaison « full-duplex ».

Une liaison série est une ligne où les bits d'information (1 ou 0) arrivent successivement, soit à intervalles réguliers (transmission synchrone), soit à des intervalles aléatoires, en groupe (transmission asynchrone). La liaison RS232 est une liaison série asynchrone.

L'octet à transmettre est envoyé bit par bit (poids faible en premier) par l'émetteur sur la ligne Tx, vers le récepteur (ligne Rx) qui le reconstitue. La vitesse de transmission de l'émetteur doit être identique à la vitesse d'acquisition du récepteur. Ces vitesses sont exprimées en BAUDS (1 baud correspond à 1 bit / seconde, dans notre cas). Il existe différentes vitesses normalisées: 9600, 4800….Etc.

La communication peut se faire dans les deux sens (duplex), soit émission d'abord, puis réception ensuite (half-duplex), soit émission et réception simultanées (full-duplex).

La transmission étant du type asynchrone (pas d'horloge commune entre l'émetteur et le récepteur), des bits supplémentaires sont indispensables au fonctionnement: bit de début de mot (START Bit), bit(s) de fin de mot (STOP Bit).

D'autre part, l'utilisation éventuelle d'un bit de parité, permet la détection d'erreurs dans la transmission.

La figure 3.6 présente le bloc du module USART dans le microcontrôleur attiny2313 :

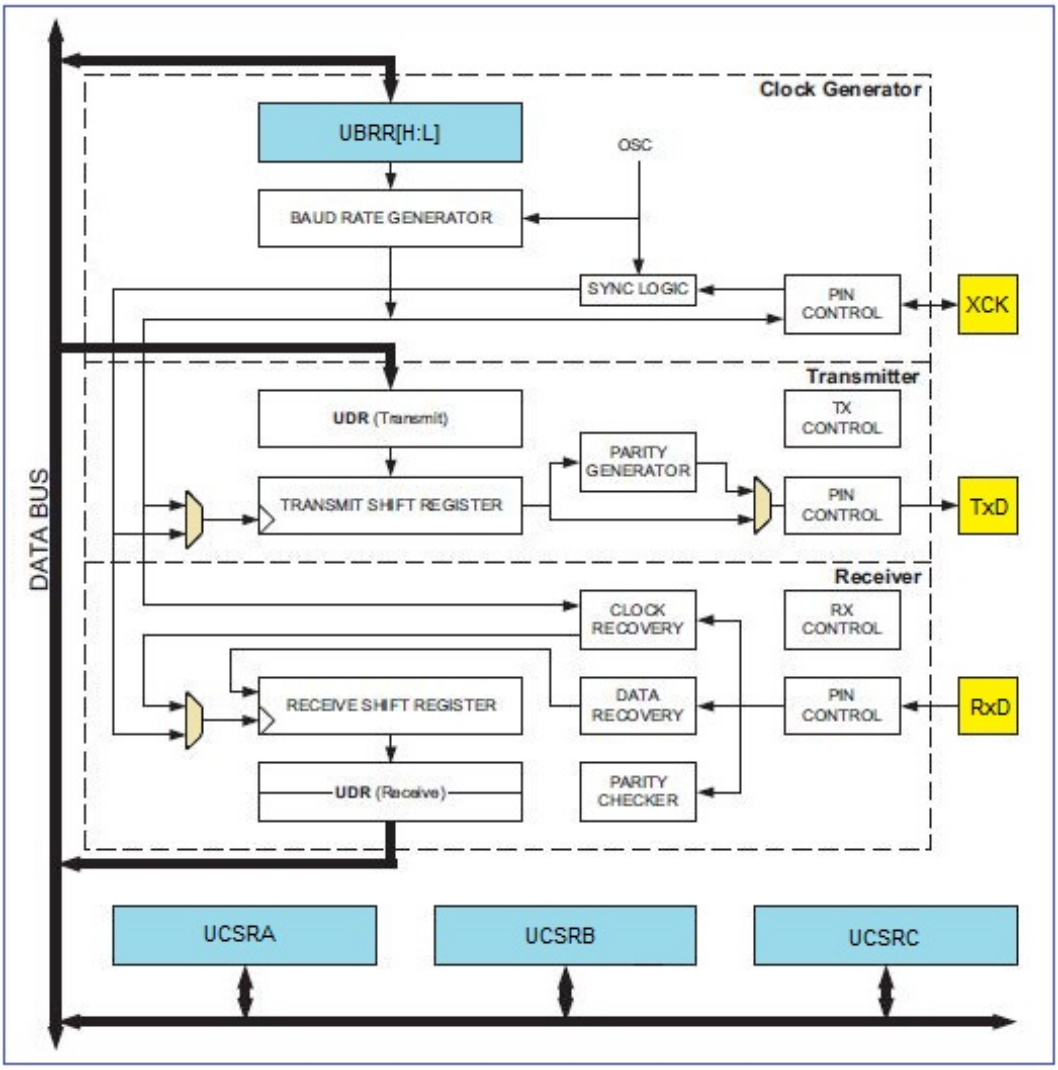


Figure 3.6 : Diagramme du bloc USART.

- **Initialisation de l'USART :**

L'initialisation de ce module consiste à régler le baud rate, le format d'une trame et l'activation du réception/transmission (tous dépend de l'application).

- Pour choisir le baud rate, on doit configurer un registre nommé UBRRL.
- Pour activer la réception/transmission (Rx/Tx), on doit configurer le registre UCSRB.

Des autres registres sont disponibles, leurs bits sont utilisés comme des flags pour savoir l'état de transmission (réception/ transmission) et ils sont utilisés dans des autres configurations, pour des autres applications.

3.2.1.2 Le PC :

La porte série du PC peut être configurée par MatLab en utilisant le programme suivant :

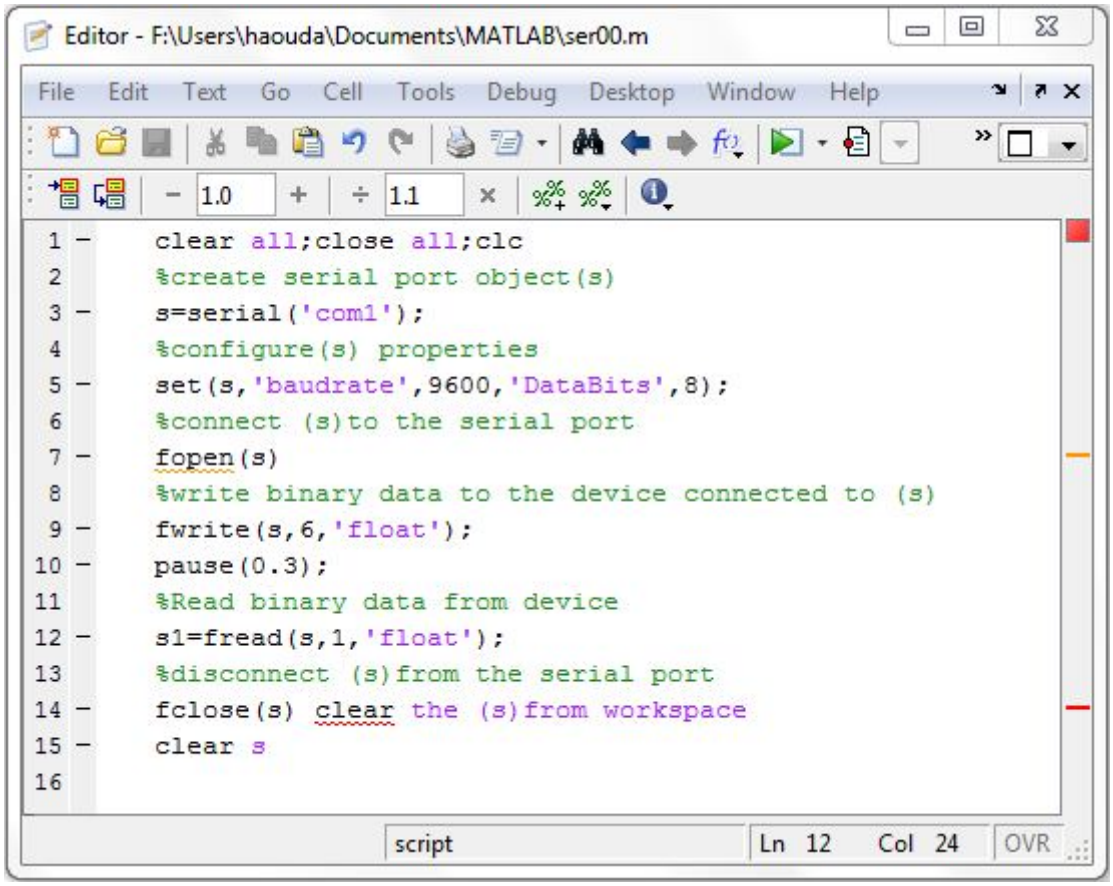


Figure 3.7 : Configuration de la porte série avec MatLab.

3.3 Programmation du microcontrôleur :

Le microcontrôleur utilisé dans cette application est programmé en C. L'utilisation du langage C permet d'alléger considérablement le développement d'une application.

3.3.1 Etapes de développement du programme :

Nous avons procédé à subdiviser notre travail en trois étapes :

- Etape1 : l'activité de programmation est un jeu de construction dans laquelle, il suffit d'enchaîner des instructions élémentaires pour parvenir à résoudre notre problème. Dans notre cas nous avons utilisé le logiciel de compilation « Code Vision AVR IDE».

- Etape2 : Après l'obtention d'un programme compilé, le besoin de simuler son bon déroulement devient une nécessite puisqu'il nous permet d'avoir une idée claire sur le côté matériel, de plus nous pouvons visualiser le comportement du microcontrôleur avec ses périphériques. Dans notre cas nous avons opté pour le logiciel **« ISIS»** qui nous permettra de chargé facilement le programme compilé dans le microcontrôleur.
- Etape3 : Dans la phase terminale, une fois le fichier source compilé et simulé, il va falloir le transférer dans la mémoire du microcontrôleur. Pour cela il faut connecter le kit NA001 avec le PC par le câble de programmation, le logiciel de transfert utilisé est le CVR.

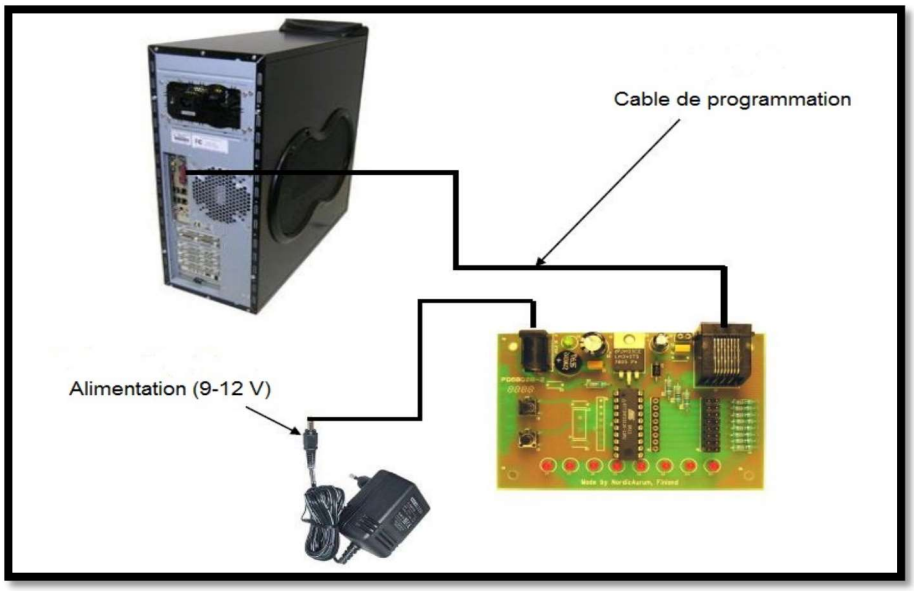


Figure 3.8 : installation du système

La figure 3.9 montre la connexion entre le câble de programmation et le microcontrôleur.

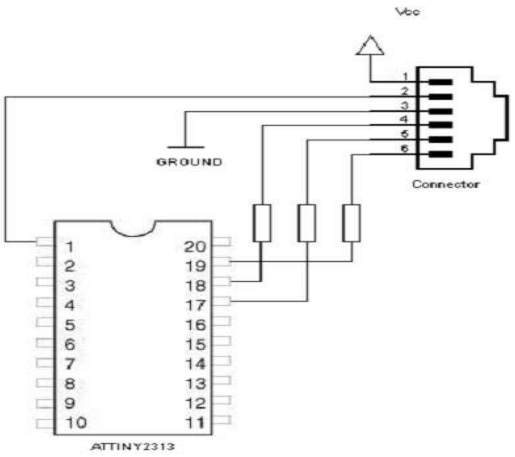


Figure 3.9 : Programmation de l'ATTINY2313.

Le câble est relié à une fiche en parallèle sur l'ordinateur et d'une fiche RJ45 sur la carte d'essai.
Le pin 2 est utilisé comme le bouton RESET pour le microcontrôleur. Quand un programme est chargé dans le microcontrôleur, les registres doivent être mis à leur valeur initiale.
Broche 4, 5 et 6 sont les broches de programmation.

3.3.2 Ecriture du programme :

3.3.2.1 Présentation du logiciel de programmation :

Le logiciel de programmation que nous utilisons est CodeVision AVR. Il s'agit d'un environnement de développement intégré, c'est un compilateur en langage C. Il est largement utilisé par les entreprises et les universités pour la programmation de microcontrôleur, comme l'ATTINY2313. Dans cette partie, nous présenterons la configuration du logiciel.

3.3.2.2 Interface :

L'interface de CVR est l'image ci-dessous :

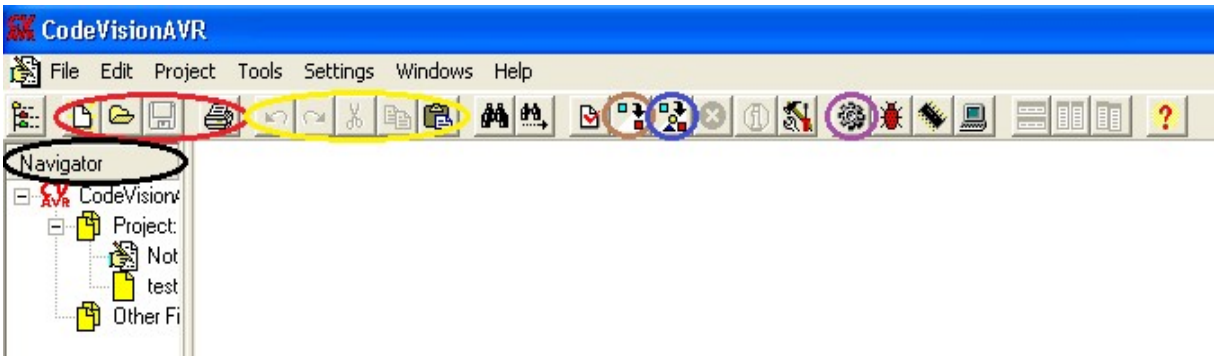


Figure 3.10: Principaux outils de CodeVisionAVR.

Le cercle noir désigne le navigateur, il permet de naviguer entre les différents fichiers du projet. Le cercle rouge regroupe les quatre fonctions de base d'un logiciel : créer un nouveau fichier ou projet, ouvrir un fichier existant, enregistrer et imprimer. Le cercle jaune réunit cinq autres importantes possibilités : annuler, restaurer, couper, copier et coller.
Les cercles vert, bleu et violet sont les plus intéressants. Le premier icône est la compilation. Elle permet de compiler le code et de vérifier les éventuelles erreurs et « warning 2».
L'icône désigné en bleu compile dans un premier temps, puis implante le programme dans le composant. Enfin, le cercle violet, l'engrenage, ouvre une fenêtre de paramétrage et génère automatiquement un code en fonction des paramètres. Ce code est ensuite à compléter.

3.3.2.3 Configuration :

Dans cette partie, nous allons aborder la configuration du logiciel CodeVisionAVR afin de pouvoir programmer correctement le microcontrôleur.

Toutes les étapes suivantes sont importantes et doivent être exécutées rigoureusement.
Tout d'abord, il faut créer un nouveau projet, en cliquant sur l'icône « Create new file » à droite de la barre cerclée de rouge dans la partie précédente. La fenêtre suivante apparaît :

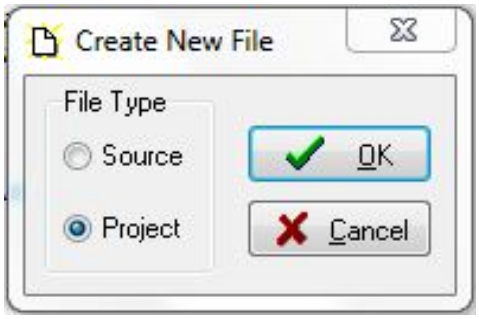


Figure 3.11 : Création d'un nouveau projet.

Après un appui sur la touche « OK » pour valider, le programme ouvre une fenêtre de confirmation. Il faut également valider en cliquant sur « Yes ».

Ensuite, la fenêtre de paramétrage du composant s'affiche. Il faut remplir les champs comme suit :

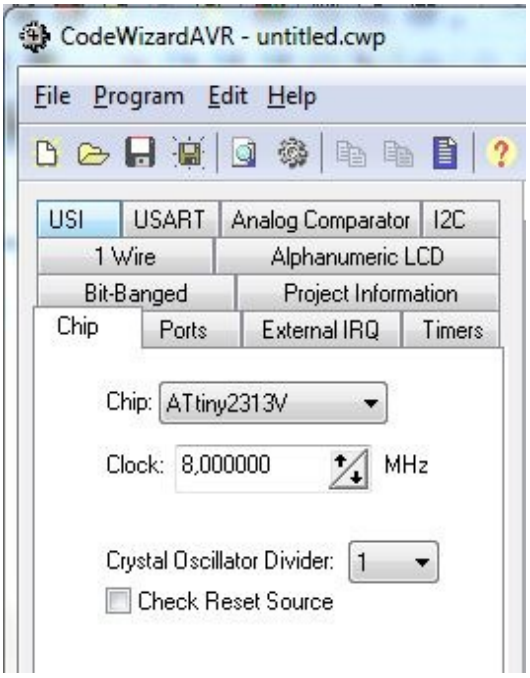


Figure 3.12: Fenêtre de paramétrage.

Puis en passant dans l'onglet « Ports », on peut programmer les quatre ports A, B et C selon qu'ils sont en entrée ou en sortie.

Par défaut, tous les ports sont en entrée, tous les bits sont alors indiqués en « In » pour les « Data Direction » et les « Pullup/Output Value » sont à 1.

Il suffit de cliquer sur « In » pour que les bits passent en « Out » et à 0.

Puis en passant dans l'onglet « USART », la configuration sera comme suite :

Activer la reception / transmission, choisir le baud rate ,choisir les parametres de communication et le mode.

Cette configuration est présentée dans la figure suivante :

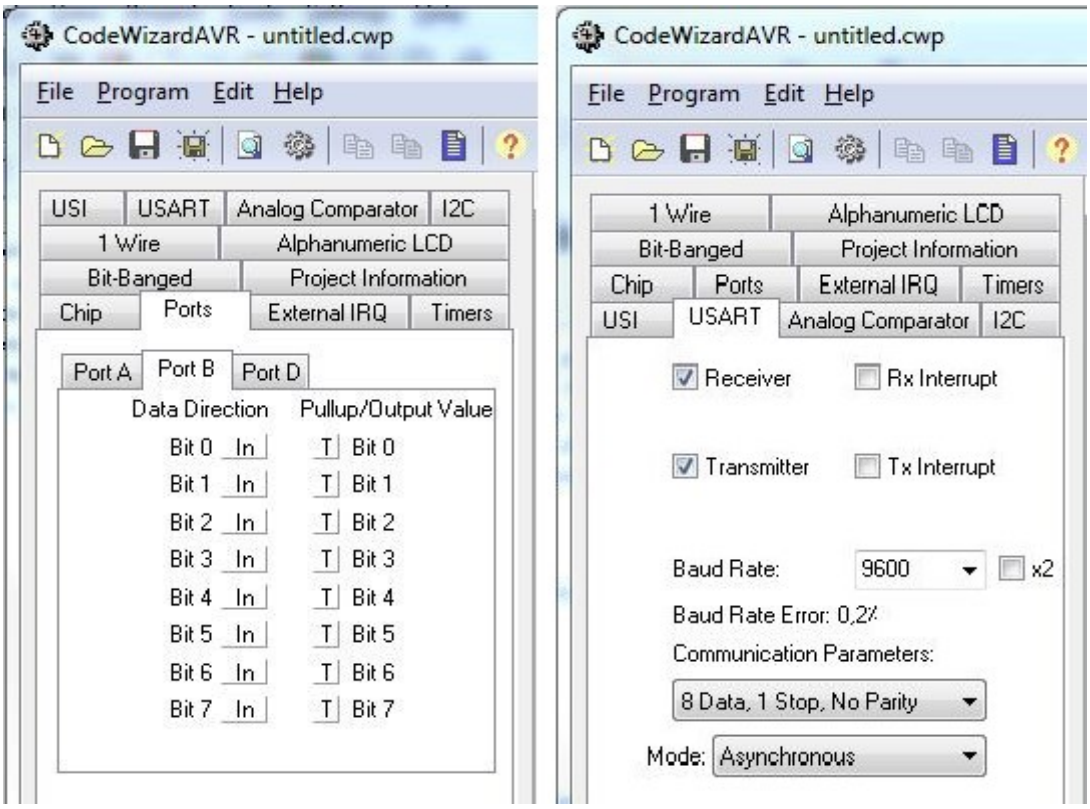


Figure 3.13: Configuration des ports d'entrées / sorties et de l'USART.

Le logiciel est maintenant bien paramétré. Pour retrouver cette configuration dans le programme, afin qu'elle prenne effet, le logiciel génère du code qui est ensuite à compléter. C'est la commande « Generate » qui donne ce code dans une nouvelle fenêtre. Le programme demande la sauvegarde du logiciel. Enfin la fenêtre de configuration est fermée.

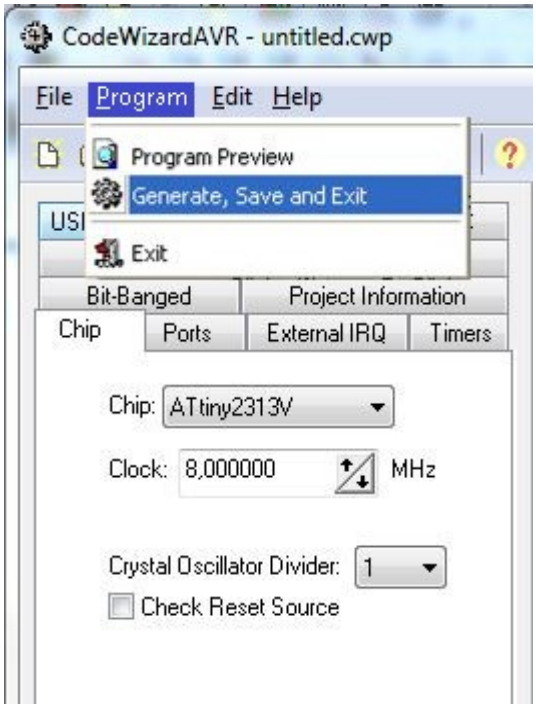


Figure 3.14: Génération du code.

Il reste maintenant deux éléments à paramétrer.

La fenêtre « Programmer Settings », s'obtient en suivant le chemin « Prommager Settings ». Par défaut, le premier champ est « AtmelSTK500/AVRISP »; il faut alors sélectionner «Dontronics DT006».

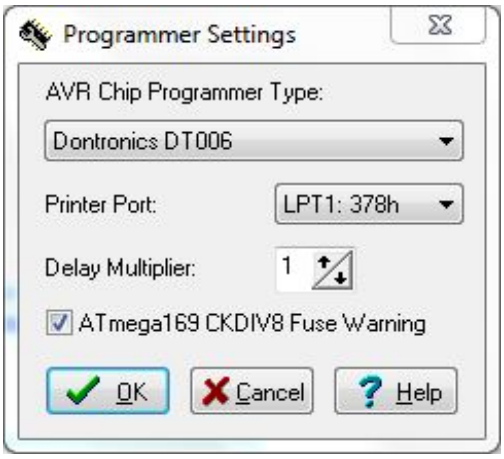


Figure 3.15: La fenêtre "programmer settings"

Enfin, il reste la configuration du projet à faire : il faut suivre le chemin Tools/Chip programmer. Pour transférer le fichier source compilé dans la mémoire du microcontrôleur, il suffit de cliquer sur le bouton « Programme All ».

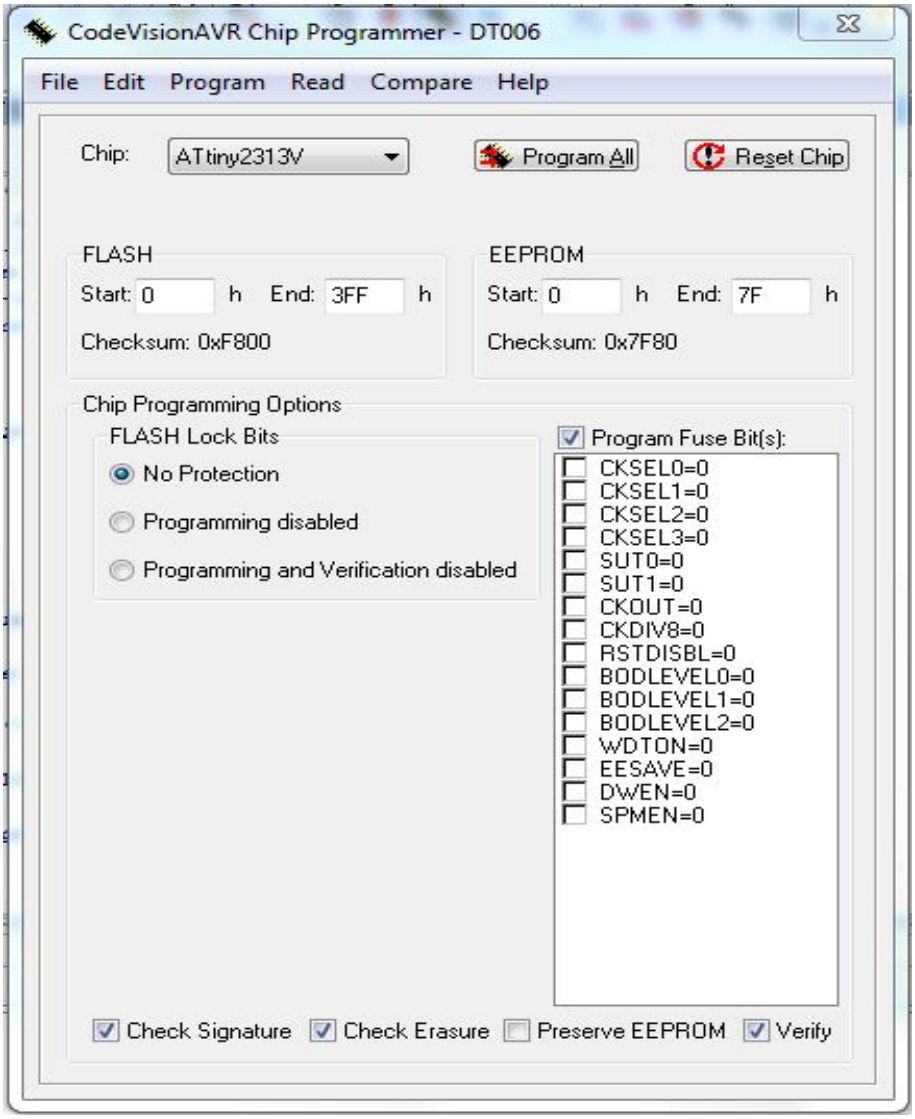


Figure 3.16 : La fenêtre Chip Programmer

3.4 Organisation du programme :

Avant de procéder à l'élaboration du programme contenu dans le microcontrôleur, il est préférable d'établir au préalable un algorithme qui définisse son principe de fonctionnement.

L'organigramme suivant décrit sommairement l'organisation du programme. Il représente l'initialisation de l'USART et décrit la boucle principale réalisant la correction.
L'asservissement se fait en trois phases : la réception de l'erreur, le calcul de la commande et l'émission du résultat sur Tx.

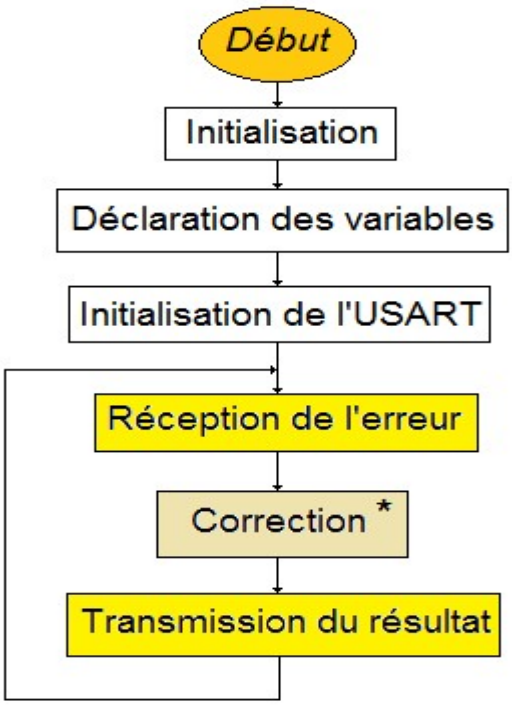


Figure 3.17 : Organisation du programme

3.4.1 Initialisation :

Dans cette partie du programme, on a déclaré les bibliothèques, les variables globales et les fonctions à utiliser, cette partie incluse aussi la déclaration des entrées/sorties et la configuration de la fréquence utilisée, l'horloge…

3.4.2 Déclaration des variables :

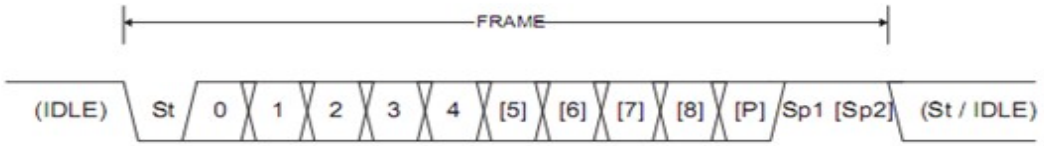
Les variables locales, les constants et leurs types sont déclarés dans cette partie au début du programme.

3.4.3 Initialisation de l'USART :

Il faut tout d'abord autoriser la Réception et la Transmission par l'USART. Ceci s'obtient en mettant à 1 les Bits RXEN (Reception Enable) et TXEN (Transmission Enable) du Registre UCSRB (USART Control and Status Register B).

UCSRB	7	6	5	4	3	2	1	0
	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W
Initial Value	0	0	0	0	0	0	0	0

Il faut ensuite déterminer le format d'échange des données (Frame Format).



IDLE Etat de repos. Toujours haut.

St Start bit. Toujours bas.

0 à 8 Bits de donnée. Les éléments entre [ ] sont facultatifs. Les autres sont obligatoires.

P Bit de Parité. Facultatif. Parité paire ou impaire.

Sp Stop bit. Le deuxième est facultatif. Toujours haut.

Le format le plus courant se présente comme ceci: 1 Start Bit, 8 Data Bits, no Parity Bit, 1 Stop Bit.

Le format est déterminé par les bits UCSZ2:0, UPM1:0 et USBS des registres UCSRB et UCSRC.

Voici le contenu de ces registres:

UCSRB

	7	6	5	4	3	2	1	0
	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W
Initial Value	0	0	0	0	0	0	0	0

UCSRC

	7	6	5	4	3	2	1	0
	URSEL	UMSEL	UPM1	UPM0	USBS	UCSZ1	UCSZ0	UCPOL
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	1	0	0	0	0	1	1	0

Le nombre de Data Bits est donné par la combinaison des bits UCSZ selon le tableau suivant:

UCSZ2	UCSZ1	UCSZ0	Character Size
0	0	0	5-bit
0	0	1	6-bit
0	1	0	7-bit
0	1	1	8-bit
1	0	0	Reserved
1	0	1	Reserved
1	1	0	Reserved
1	1	1	9-bit

Tableau3.1 : configuration des bits UCSZ

Pour un format de 8 Data Bits, on voit que les valeurs de UCSZ correspondent aux valeurs par défaut de ces Bits. Il n'est donc pas indispensable d'initialiser ces Bits.

Le mode de parité est déterminé par les Bits UPM1:0 selon le tableau ci-dessous:

UPM1	UPM0	Parity Mode
0	0	Disabled
0	1	Reserved
1	0	Enabled, Even Parity
1	1	Enabled, Odd Parity

Tableau 3.2 : configuration des bits UPM.

On voit à nouveau que pour une absence de contrôle de parité, les bits UPM ont la valeur par défaut. Leur initialisation n'est donc pas indispensable.

Enfin, la sélection du nombre de Stop Bit se fait par USBS (Usart Stop Bit Select) selon le tableau ci-dessous :

USBS	Stop Bit(s)
0	1-bit
1	2-bit

Tableau 3.3 : configuration du bit USBS.

Pour 1 Stop Bit, la valeur est à nouveau la valeur par défaut.
Il reste à fixer le mode synchrone ou asynchrone selon le tableau suivant :

UMSEL	Mode
0	Asynchronous Operation
1	Synchronous Operation

Tableau 3.4 : configuration du bit UMSEL.

Ici encore le Bit pour transmission asynchrone a sa valeur par défaut.
Si on veut quand même initialiser l'USART, voici la procédure :

```
void USART_Init( unsigned int baud )
{
  /* Set baud rate */
  UBRRH = (unsigned char)(baud>>8);
  UBRRL = (unsigned char)baud;
  /* Enable receiver and transmitter */
  UCSRB = (1<<RXEN)|(1<<TXEN);
  /* Set frame format: 8data, 2stop bit */
  UCSRC = (1<<USBS)|(3<<UCSZ0);
}
```

3.4.4 Réception de l'erreur :

Lorsqu'une donnée est reçue, le bit RXC du registre UCSRA est mis à 1. le bit RXC du registre UCSRA est mise à 1.

UCSRA

	7	6	5	4	3	2	1	0
	RXC	TXC	UDRE	FE	DOR	PE	U2X	MPCM
Read/Write	R	R/W	R	R	R	R	R/W	R/W
Initial Value	1	0	0	0	0	0	0	0

Voici un exemple de routine de réception d'une donnée :

```
unsigned char USART_Receive( void )
{
  /* Wait for data to be received */
  while ( !(UCSRA & (1<<RXC)) )
       ;
  /* Get and return received data from buffer */
  return UDR;
}
```

Le problème qui se pose c'est que les variables à recevoir sont de type « float » qui sont codé sur 32 bits, pour cela on a utilisé un pointeur de type « char » qui utilise 8 bits et on a effectué une concaténation de 4 valeurs successives de ce pointeur pour avoir la variable reçue de type « float ».

Cela prend un temps de calcul considérable mais ne pose pas de problème car la nature des ballons dirigeable exige une régulation lente.

3.4.5 Transmission des résultats :

Avant l'envoi d'une donnée, il faut attendre que le Registre UDR (Usart Data Register) soit vide.

Lorsqu'il est vide, le bit UDRE (Usart Data Register Empty) de registre UCSRA mise à 1.

UCSRA	7	6	5	4	3	2	1	0
	RXC	TXC	UDRE	FE	DOR	PE	U2X	MPCM
Read/Write	R	R/W	R	R	R	R	R/W	R/W
Initial Value	0	0	1	0	0	0	0	0

Voici un exemple de routine d'envoi d'une donnée:

```
void USART_Transmit( unsigned char data )
{
  /* Wait for empty transmit buffer */
  while ( !( UCSRA & (1<<UDRE)) )
       ;
  /* Put data into buffer, sends the data */
  UDR = data;
}
```

Pour le cas d'une information codée sur 32 bits, on utilise un pointeur pour transmettre cette information par morceaux de 8 bits.

3.4.6 Correction :

3.4.6.1 Commande linéaire :

La réalisation d'un asservissement PD numérique se résume à l'application de la formule :

$$U(k) = Kp\ e(k) + Kd[\ e(k) - e(k-1)] \qquad (3.1)$$

Pour plus de détail de cette formule, voir l'annexe D.

Le choix de la période d'échantillonnage T_e dépend du temps d'exécution d'une itération du programme du microcontrôleur.

L'organigramme suivant illustre le traitement de la correction PD :

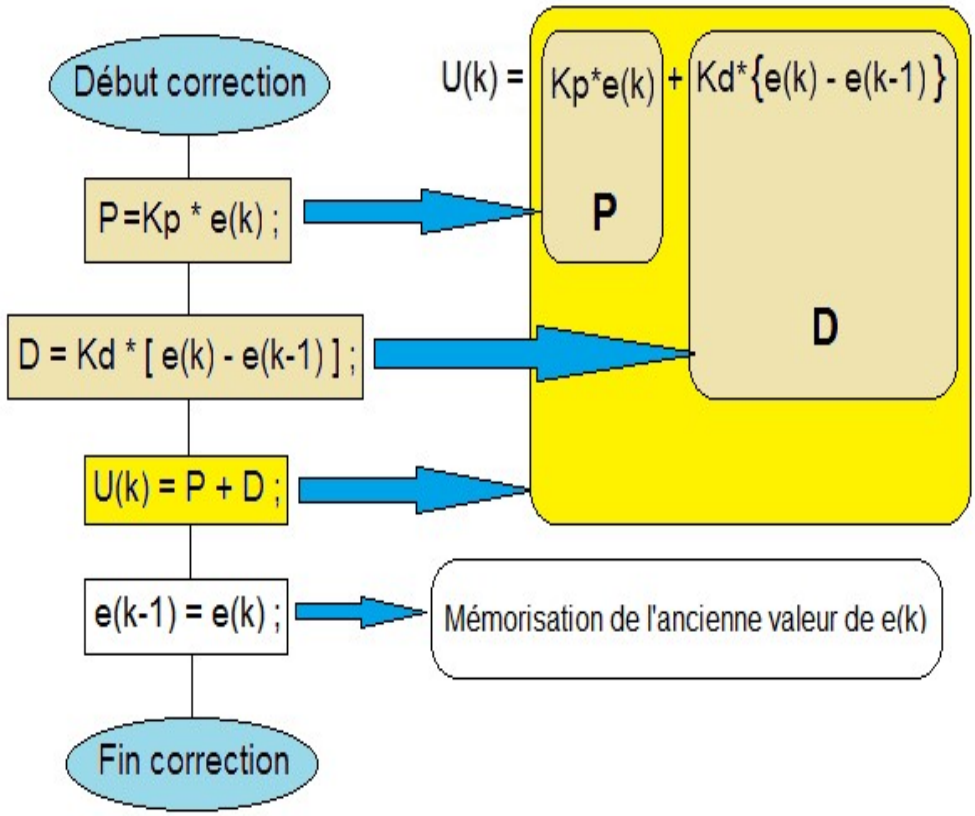


Figure 3.18 : L'organigramme de la correction PD.

3.4.6.2 Commande non linéaire :

Pour construire la loi de commande à implémenter, on a utilisés les équations de chapitre deux (2.38) et (2.39) :

$$u = u_{eq} + u_{dis} = K^T X + (S^T B)^{-1}(S^T \dot{X}_d - \eta^2 tanh(\sigma/\emptyset))$$

Et

$$S^T(A - K^T B) = 0 \quad ; \quad A - K^T B = A_c$$

La fonction « $tanh$ » peut être utilisée, elle se trouve dans la bibliothèque < math.h > du compilateur C.

L'organigramme suivant montre les différentes étapes de la correction par mode glissant :

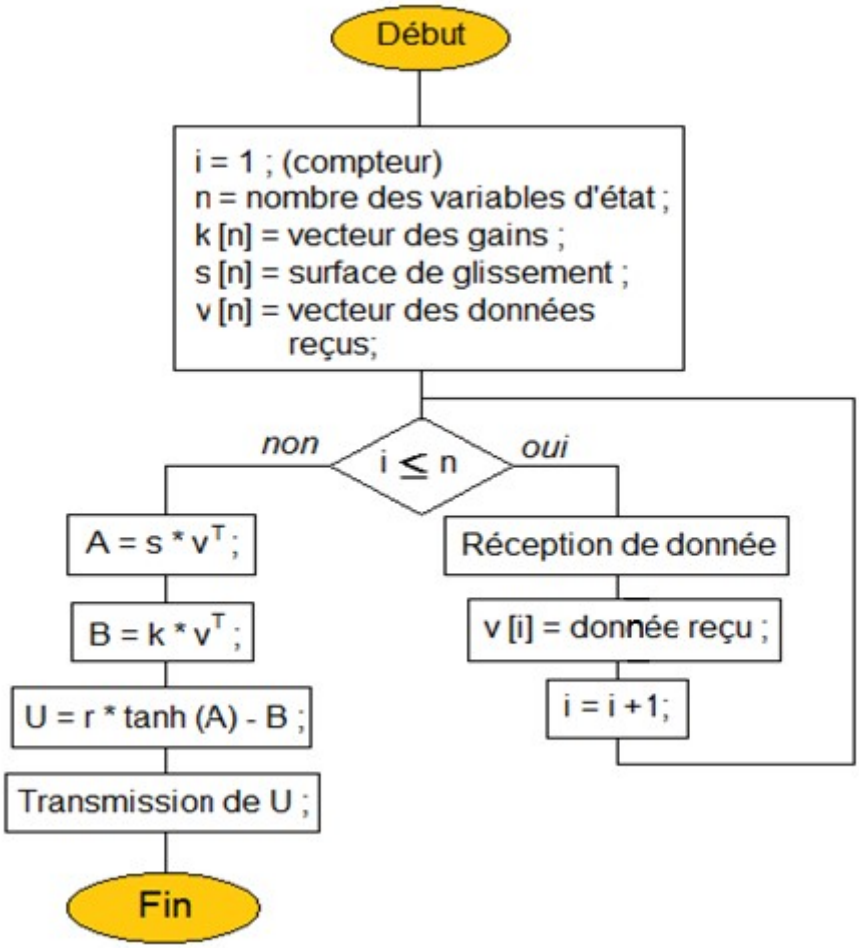


Figure 3.19 : L'organigramme de la correction par mode glissant.

3.5 Liaison avec MatLab :

Pour lier le microcontrôleur avec le SIMULINK de MatLab on a utilisé deux blocs :

- **To Instrument** : pour envoyer les données vers le microcontrôleur.
- **Query Instrument** : pour recevoir les données envoyées par le microcontrôleur.

Ces blocs sont dans la boite à outils : **Instrument Control Toolbox**.

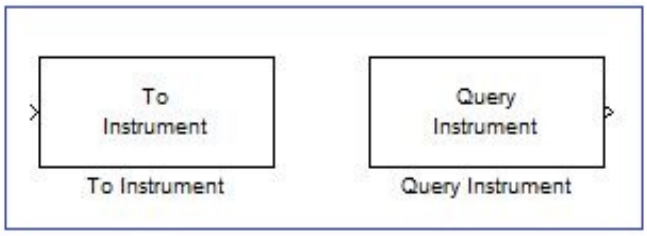


Figure 3. 20 : les blocs **To Instrument** et **Query Instrument**.

3.6 Conclusion :

Nous venons de présenter dans ce chapitre les différents étapes suives pour l'implémentation des lois de commande étudiées.

Pour la tache de communication, on a cité les différentes configurations nécessaires pour lier le PC (MatLab) avec le microcontrôleur.

Nous devons noter que les périphériques et les composants dont le kit de formation NA001 dispose nous ont beaucoup facilité la tâche d'implémentation et de réalisation des algorithmes de commande développés.

4.1 Introduction :

Dans ce chapitre, nous présentons les simulations numériques confirmant la convergence du dirigeable vers le point cible au moyen des commandes développées. Les simulations ont été effectuées en utilisant MATLAB/Simulink.

Dans une deuxième partie, nous présentons les résultats des lois de commande implémentés dans le microcontrôleur.

- **Détermination de trajectoire du Ballon Dirigeable :**

Nous avons fixé la vitesse linéaire à 9.5 m/s. Pour explorer l'exécution des lois de commandes, nous avons simulé un vol autonome à quatre phases de vol et maintien stationnaire à une altitude désirée pendant une période prédéterminée.

On a :

$$\begin{cases} \dot{x} = u\cos\Psi - V_y \sin\Psi \\ \dot{y} = u\cos\Psi - V_y \sin\Psi \end{cases}$$

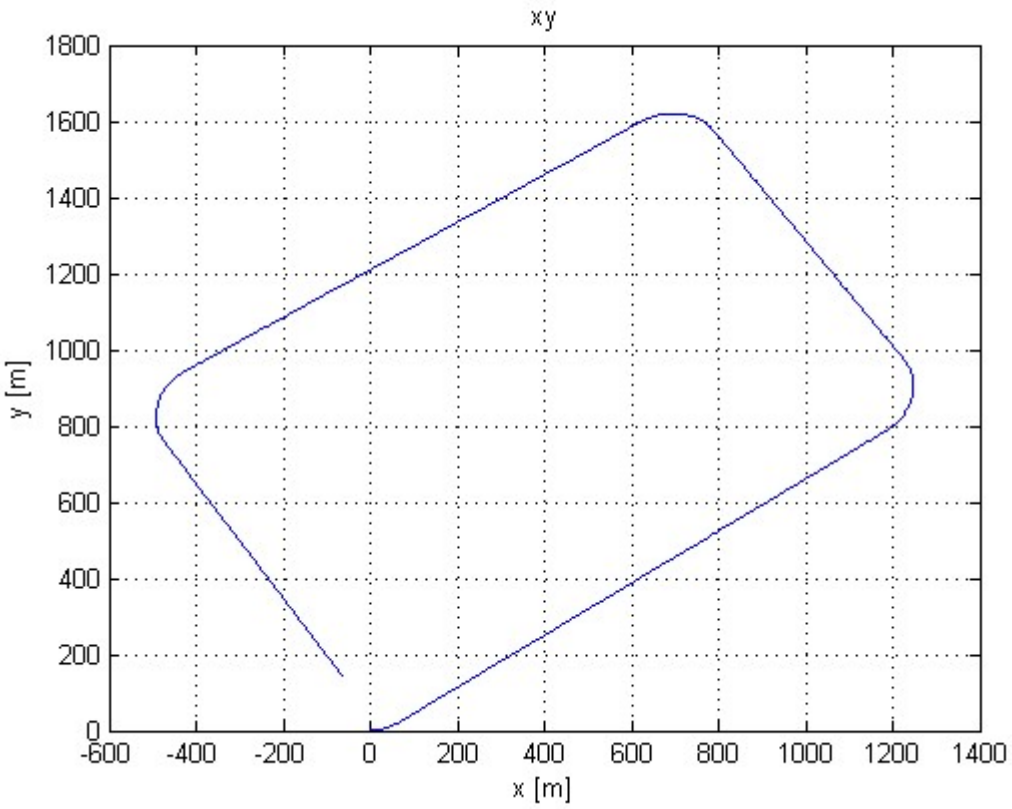


Figure 4.1 : Trajectoire du ballon dirigeable pour 04 phases du vol.

4.2 Blocs de simulations :

4.2.1 Mode longitudinal :

- **Commande linéaire :**

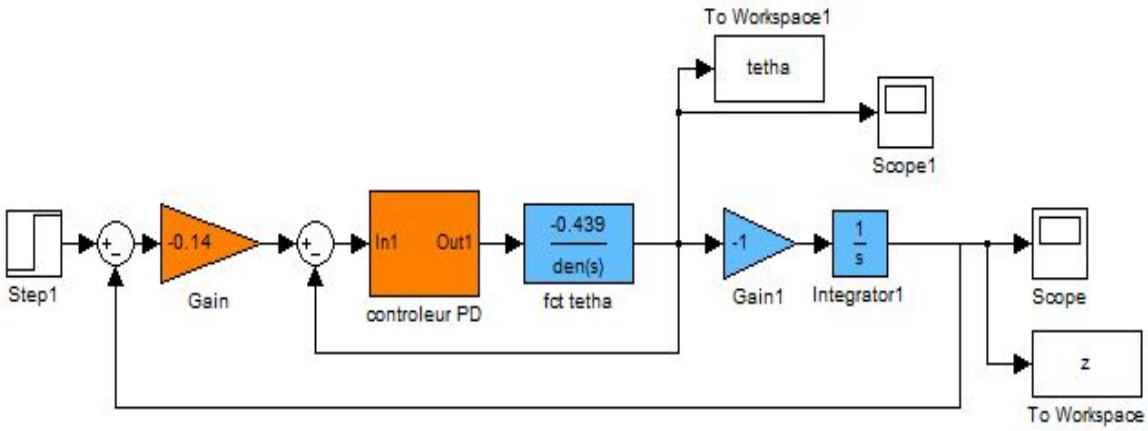


Figure 4.2 : la commande PD sous Simulink.

- **Commande par mode de glissement :**

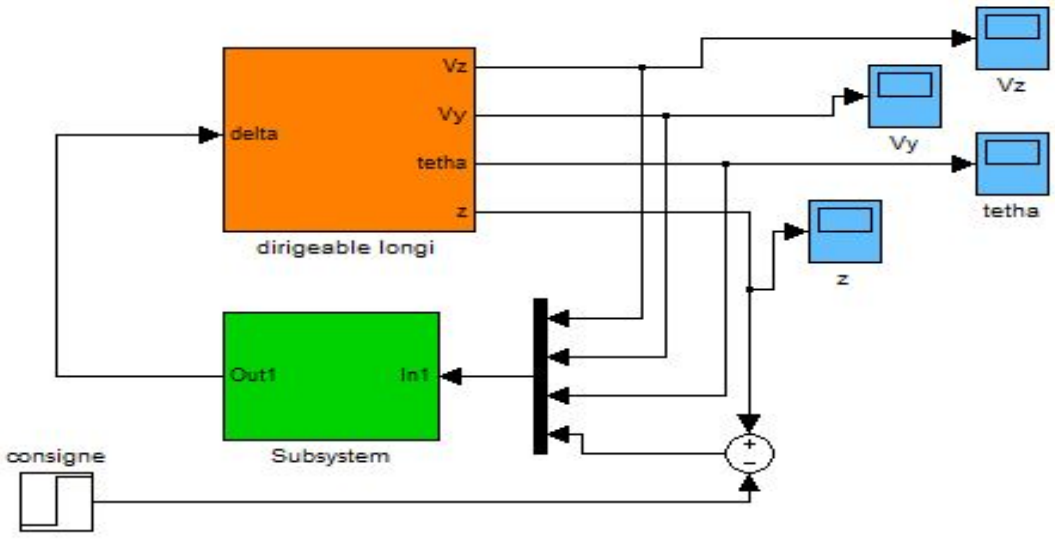


Figure 4.3 : la commande par mode glissant sous Simulink.

- **Résultats :**

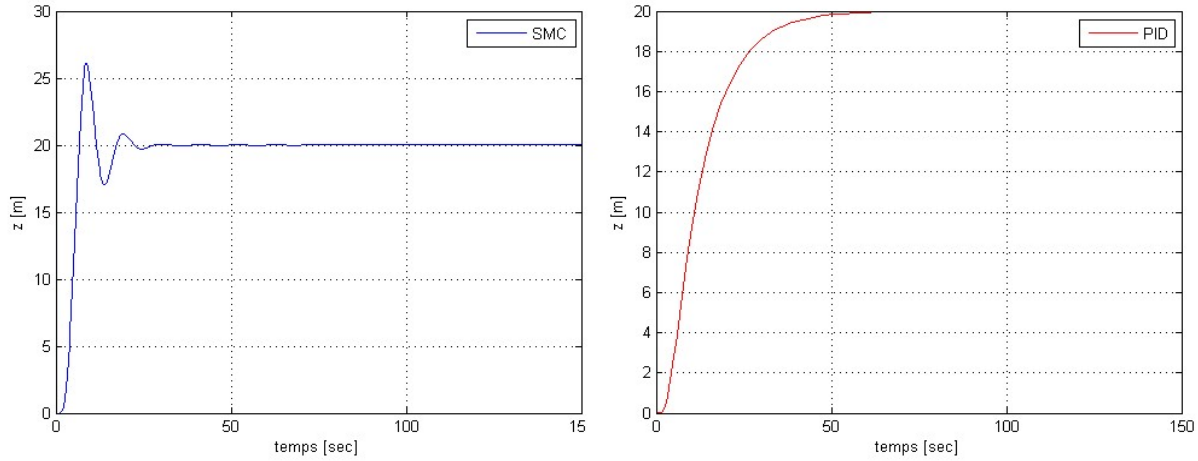


Figure 4.4 : L'altitude z devant les lois PID et SMC.

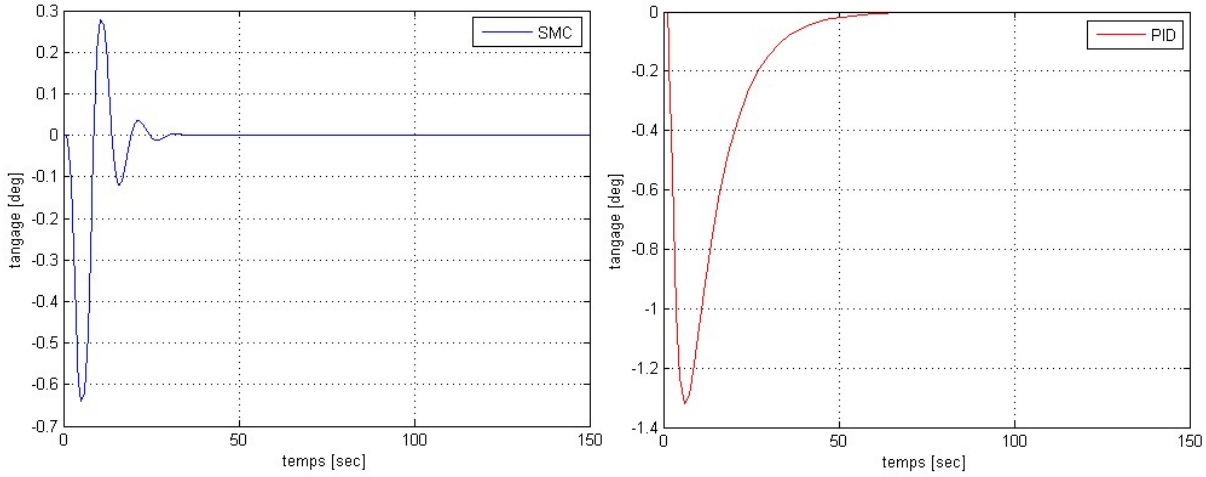


Figure 4.5 : Tangage θ devant les lois PID et SMC.

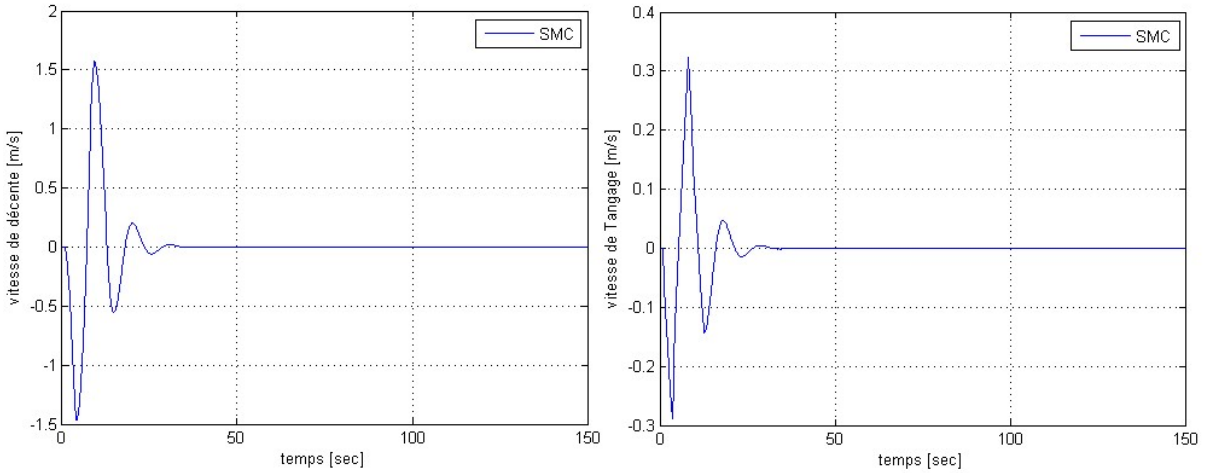


Figure 4.6 : Vitesse de déscente V_z et vitesse de Tangage w_y devant la loi SMC.

4.2.2 Mode latéral :

- **Commande linéaire :**

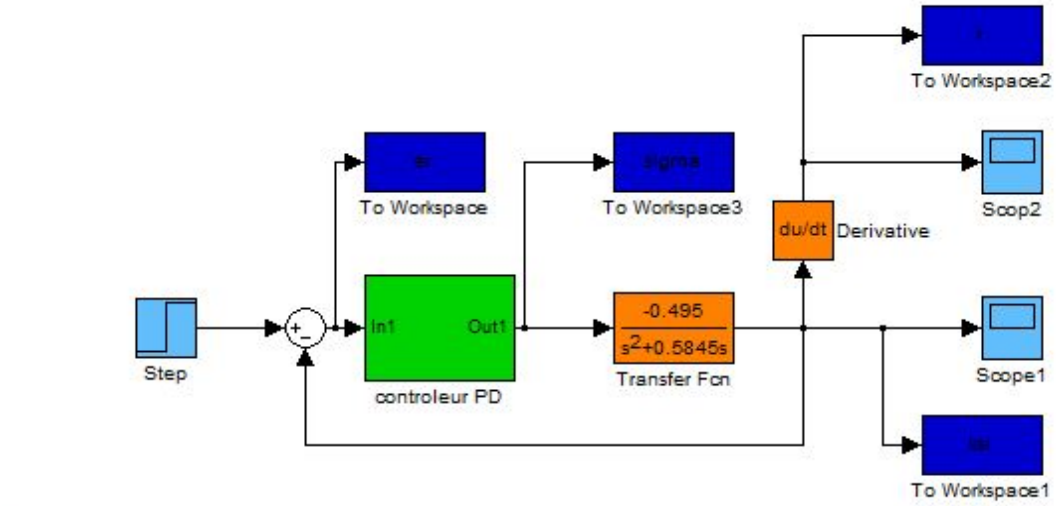


Figure 4.7 : la commande PD sous Simulink.

- **Commande par mode de glissement :**

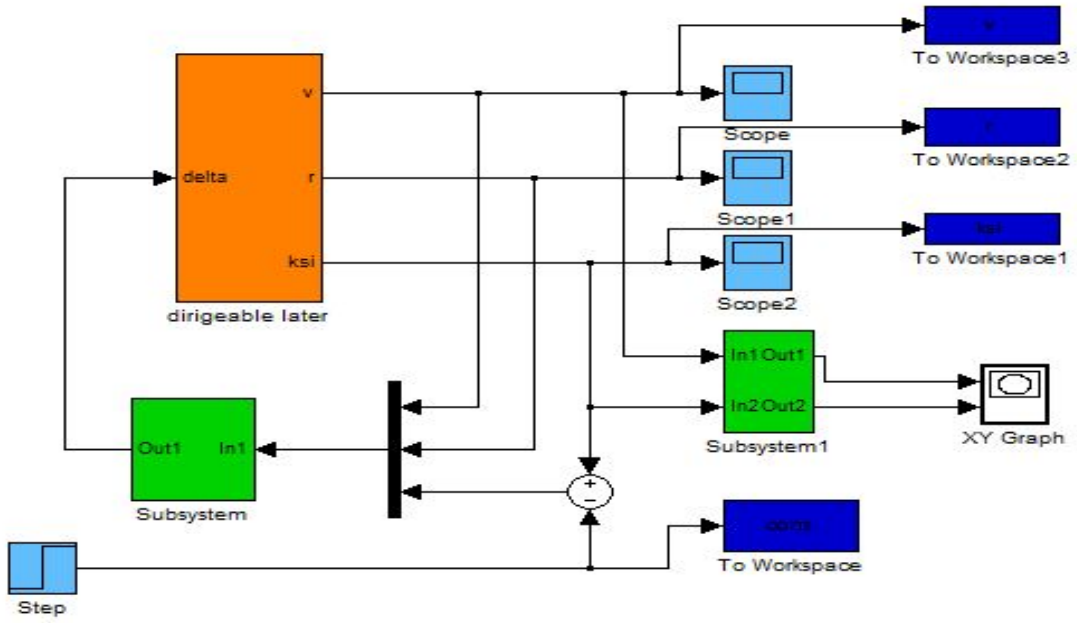


Figure 4.8 : la commande par mode glissant sous Simulink.

➢ **Résultats :**

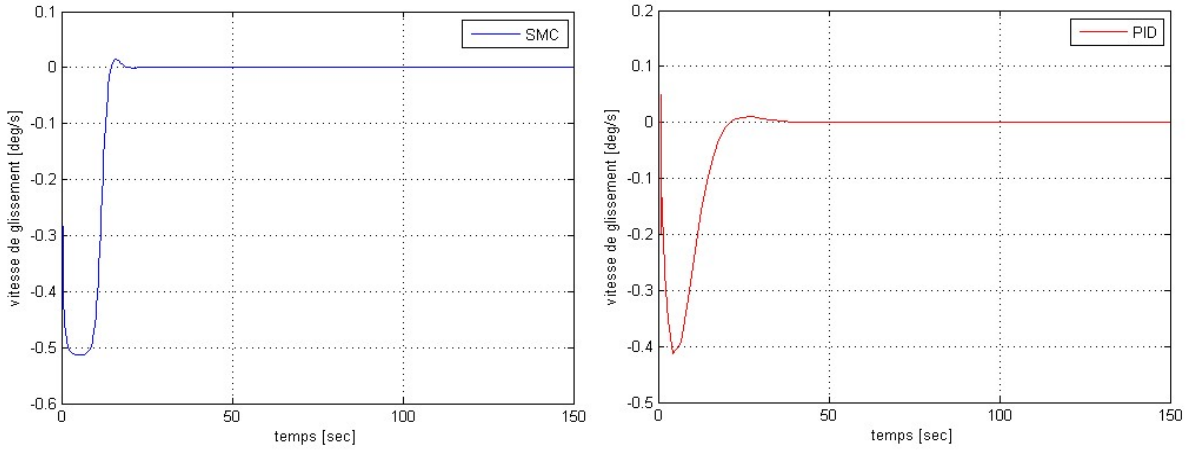


Figure 4.9 : Vitesse de Glissement V_y devant les lois PD et SMC.

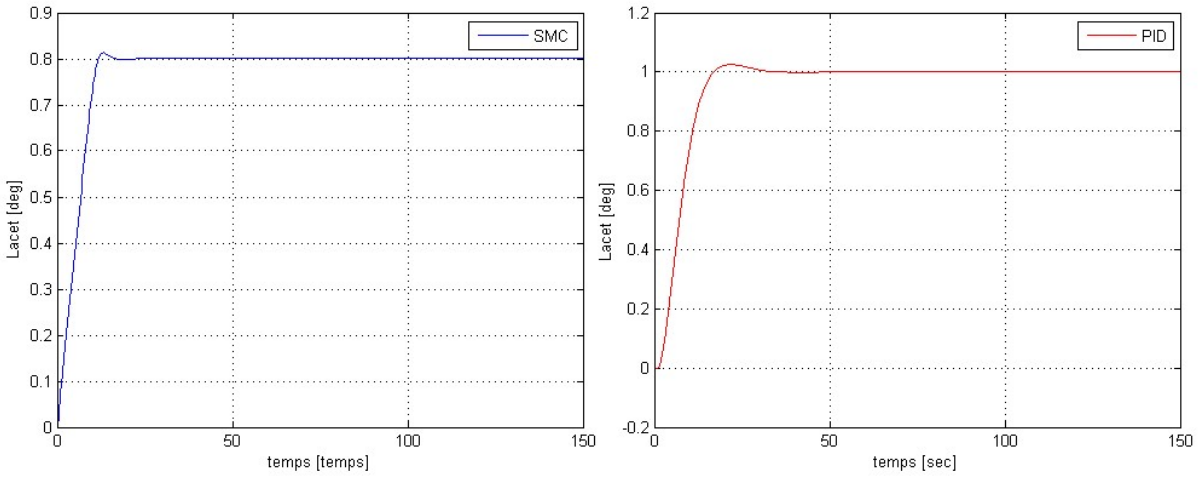


Figure 4.10 : Lacet Ψ devant les lois PD et SMC.

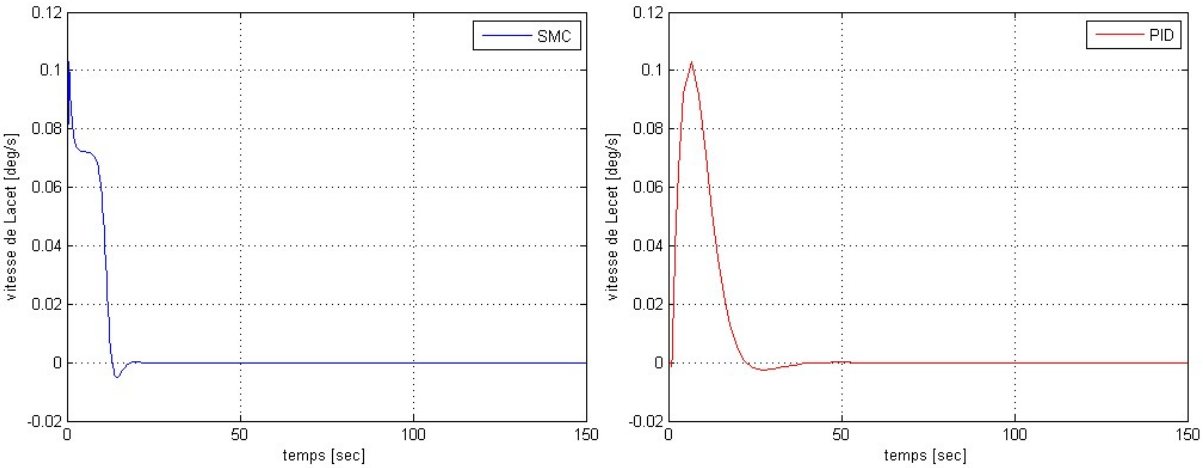


Figure 4.11 : Vitesse de Lacet w_z devant les lois PID et SMC.

4.3 Discussion :

Dans ce chapitre, nous avons présenté les résultats de simulation des lois de commande en régime glissant classique et la commande PID. Ces résultats nous ont permis de montrer que le comportement du dirigeable est correct et acceptable, contrôlé par ces lois de commande.

On remarque que le contrôleur de mode glissant (SMC) à une robustesse et une précision tout à fait meilleure que le contrôleur PID. Ce dernier est lent et peu robuste.

A noter que le contrôleur de mode glissant (SMC) provoque des dépassements et des oscillations indésirables.

D'après la figure 4.4, on remarque aussi que la commande du mouvement longitudinale (Altitude), assure bien la stabilisation du dirigeable autour de l'altitude désirée z_d=20 m dans moins de 30 secondes dans le cas du contrôle par mode glissant et dans 50 secondes dans le cas du contrôle PID.

4.4 Résultats d'implémentation :

4.4.1 Bloc de simulation :

Pour connecter le Simulink du MatLab avec le microcontrôleur, on a utilisé le schéma suivant :

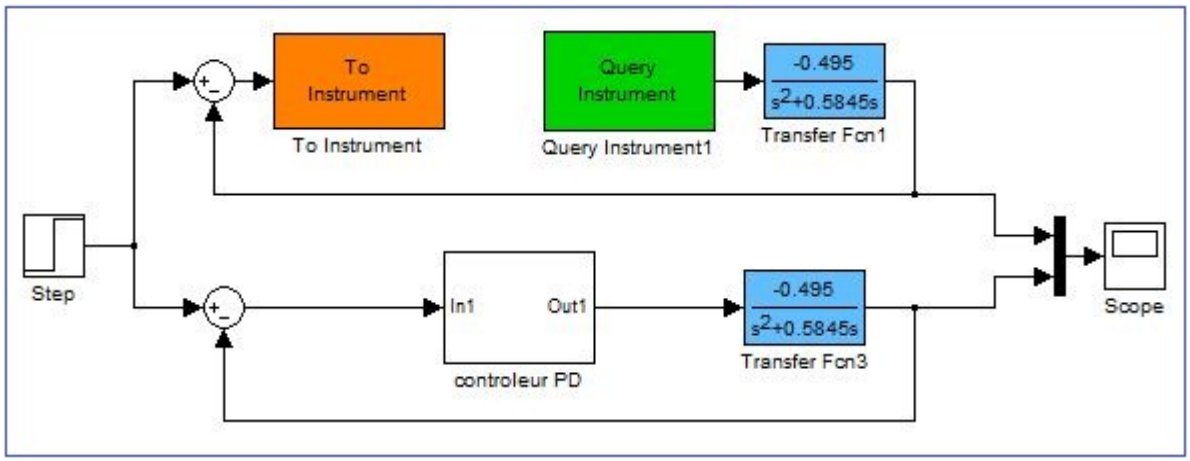


Figure 4.12 : connexion de Simulink avec le microcontrôleur

4.4.2 Problèmes rencontrés:

Pendant l'implémentation et le test des programmes on a rencontré des problèmes :

- Le microcontrôleur a une mémoire très limitée (2 Ko de mémoire de programme qui est utilisé pour stocker du code exécutable et les constantes; 128B de la mémoire de données qui est utilisée pour stocker le modèle et les variables du programme).
- Le microcontrôleur supporte les entiers 16 bits et les opérations en virgule fixe. La multiplication et la division des nombres entiers est très rapide, mais ça pose des problèmes de débordement potentiel avec multiplication et erreur d'arrondi avec la division.
- Il ne supporte pas directement les nombres à virgule flottante, mais peuvent faire face avec eux en utilisant des solutions de certains programmes.
- Les calculs en virgule flottante peuvent aussi être soutenue par des bibliothèques de langage C, mais seulement après avoir engagé des frais généraux importants dans les deux mémoires et temps d'exécution.
- Problème de la configuration des blocs « Query instrument » et « To instrument ».
- La synchronisation du matlab avec le périphérique.

4.4.3 Solutions proposés :

- Pour le problème des nombres à virgule flottante, on a utilisé les pointeurs pour la manipulation des données de type « float ».
- Pour le MatLab on n'a pas trouvé la bonne façon de configurer les blocs de communication avec simulink.

4.5 Conclusion :

A la conclusion de ce chapitre, nous pouvons faire état de beaucoup de points positifs mais aussi négatifs.

Pour commencer, faisons état de ce qui a été, et de ce qui n'a pas été réalisé :

- Nous avons pu faire simuler le vol du ballon dirigeable avec un contrôle par mode de glissement et un contrôle PID.
- On a pu mettre en place différentes matières telles que l'électronique, l'informatique, et l'automatique ont été appliquées pour la réalisation de notre projet.

Parmi les travaux que nous n'avons pas réalisés, nous pouvons compter :

- la vérification de techniques implémentées dans le microcontrôleur par simulink.
- Implémenter les algorithmes sans dépasser l'espace mémoire disponible dans le microcontrôleur.
- Etudier des autres techniques de commande.

Conclusion et perspectives :

Les travaux présentés dans ce mémoire ont porté sur la réalisation d'un système de commande d'un ballon dirigeable à l'aide des composants électroniques programmables basés sur une étude de la modélisation d'un ballon dirigeable, en utilisant un microcontrôleur de la famille AVR c'est l'ATtiny2313V.

Dans un premier temps, on a présenté une synthèse de la modélisation des dirigeables autonomes. Nous avons pour cela établi deux modèles : cinématique et dynamique fondés sur l'hypothèse d'un corps rigide. Cette hypothèse permet de les modéliser de manière simplifiée par la méthode de Newton Euler. Ceci a l'avantage de faciliter la mise en œuvre des algorithmes de contrôle, de stabilisation ou de génération de trajectoires qui leurs sont dédiés. Nous avons découplé le modèle linéarisé dans le plan horizontal et vertical du dirigeable, ainsi que les hypothèses prises en compte pour établir ces modèles.

Puis, nous avons étudié et cité deux techniques de commande : commande par mode de glissement qui est une commande non linéaire robuste vis-à-vis des erreurs de modélisation, et perturbation extérieur. Et une commande PID qui certes non robuste mais fonctionnel
et nécessite une linéarisation du modèle étudié.
La détermination de ces lois de commande nous a servi de les implémenter dans le microcontrôleur ATtiny2313V.

Nous avons affronté des difficultés et des problèmes pendant le test des programmes implémentés. Il s'agit de problèmes de communication entre le microcontrôleur et le simulink par la liaison RS232.

Comme perspective, il serait intéressant d'utiliser d'autres techniques de commande telle que la commande par logique flou et combinaison entre plusieurs de ces méthodes par exemple le glissant flou.
Il est préférable de bien choisir le microcontrôleur (précision, vitesse d'exécution, caractéristiques).
Il est envisageable d'utiliser d'autres méthodes de test du programmes implémentés comme le logiciel LabView, ou bien d'utiliser un calculateur analogique.

Listes des figures et des tableaux

Intorduction génerale :

Chapitre 1 :

Chapitre 2 :

Chapitre 3 :

Chapitre 4 :

Bibliographie :

[1] Y.TAMI, *Commande par mode de glissement d'un ballon dirigeable autonome*, Thèse de magister, Université YAHIA FARES de Medea, 2010.

[2] A.MELBOUS, *Optimisation des paramètres d'un contrôleur neuro-flou par les algorithmes génétiques. Application a la commande du cap d'un ballon dirigeable par la vision*, Thèse de magister, Université SAAD DAHLEB de Blida, Juin 2006.

[3]A.MELBOUS, Y.TAMI, A.GESSOUM, *" UAV controller design and analysis using sliding mode"*, 3rd international conference on electrical engineer, OCT 31-NOV 02, Sousse, Tunisia, 2009.

[4] S.LACROIX, *" Ballons dirigeables autonomes"*, LAAS /CNRS, Toulouse, France.

[5]S.HIMA, Y.BESTAOUI, *" Planification de Trajectoires pour la Navigation des Dirigeables Autonomes"*, Laboratoire des Systèmes Complexes, CEMIF, Université d'Evry Val d'Essonne, France, 20 janvier 2004.

[6]A.MARTINI, *Modélisation et Commande de vol d'un hélicoptère drone soumis à une rafale de vent,* Thèse de doctorat, Université PAUL VERLAINE – Metz, France.

[7]F. FARAH, F.HACHID, *Synthèse et implémentation pratique de lois de commandes non linéaires : Application à un simulateur d'hélicoptère,* Thèse d'ingénieur, école nationale polytechnique, El-Harrach, Algérie, Juin 2005.

[8] BENNASEUR.S, *Modélisation et commande d'engins volants flexibles,* thèse de Doctorat, Université EVRY VAL d'Essonne, France, 2009.

[9] F. M. METELO, L. R. G. CAMPOS, *Vision Based Control of an Autonomous Blimp (Video Blimp),* Thèse de doctorat, Université Heriot-Watt.

[10] A.KORNIENKO, *System Identification Approach for Determining Flight Dynamical Characteristics of an Airship from Flight Data,*Thèse de doctorat, Université Stuttgart,2006.

[11] J-M.SPIEWAK, *Contribution à la coordination de flottille de véhicules sous-marins autonomes,* Thèse de doctorat, Université Montpellier II , septembre 2007.

[12] S.VAN DER ZWAAN, A. BERNARDINO, J. SANTOS-VICTOR, *" vision based station keeping and docking for floating vehicles"*, institute des systems robotiques, université Technique de Lisbon.

[13] M. TADJINE, O. STIHI, Modélisation, *commande et réalisation d'un robot mobile uni cycle*, Thèse d'ingénieur, école nationale polytechnique, El-Harrach, Algérie, Juin 2005.

[14] L.NI, *Fault Tolerant Control of Unmanned Underwater Vehicles*, Thèse de doctorat, Université Blacksburg, Virginia.

[15] X.W.CLAUDE, Y.HU,″ Modelling_and_Linear_Control_of_a_Buoyancy-Driven_Airship″, Hal, juillet 2009.

[16] A.MELBOUS, Y.TAMI, A.GESSOUM , M.HADJSADOK, *" UAV controller Design and Analysis using sliding Mode".* Laboratory of Electric System and Remote Control, Electronic Department University Saad Dahlab Blida , Algeria 2010.

[17] « *Une approche pratique aux microcontrôleurs en utilisation le Kit de formation NA001* », ATMEL, 2009.

[18] www.datasheetarchive.com/ATtiny2313V-10PU-datasheet.html

[19] P.Beugnet, R. Polonowski, M. Nik Khairail Afzan Nik, *Modélisation et interface de contrôle d'un quadri-rotor* , Rapport de projet, Ecole ingénieur électrique Amiens France,2008.