

**Non classifié**

---

# **Le système «hardware in the loop»**

*Rapport Préliminaire No. 2*

Myriam Manai

Département de génie électrique et de génie informatique  
Université Laval  
Pavillon Adrien-Pouliot  
Québec, Québec G1K 7P4

L'entrepreneur est entièrement responsable de la validité scientifique ou technique de ce rapport de contrat, et le contenu de ce rapport n'est pas nécessairement approuvé ni entériné par R et D pour la défense Canada.

PWGSC Contrat No. W7701-3-32063  
Autorité Scientifique : Eric Gagnon

**R & D pour la Défense Canada Valcartier**

Rapport de contracteur  
RDDC Valcartier CR 2004-316  
Décembre 2004

**Non classifié**

---



**Non classifié**

---

## Résumé

---

Dans le cadre de ce projet, un UAV (véhicule aérien sans pilote) a été acheté de même qu'un autopilote pour le contrôler, basé sur des PID à gains variables. L'objectif principal du projet est d'améliorer les performances de l'autopilote. Le projet vise donc à remplacer quelques uns de ces PID par des commandes non-linéaires, en se basant sur un modèle du UAV essentiellement physique. La première phase du projet consiste à modéliser et à identifier le UAV. Pour ce faire, un modèle non-linéaire à 6 degrés de liberté disponible dans une librairie spécialisée de Simulink, la librairie Aerosym d'Unmanned Dynamics, est utilisé. Afin d'étudier le comportement de l'avion face aux algorithmes non-linéaires envisagés, ces algorithmes seront implantés dans l'autopilote et celui-ci sera connecté à un ordinateur simulant l'avion à l'aide du modèle. Ainsi, un système «hardware in the loop» a été développé et implanté pour fin de tests dans les étapes ultérieures du projet. L'implantation du système est décrite, plus particulièrement il est expliqué comment les signaux de commande transmis par l'autopilote sont convertis sous la forme des entrées du modèle et comment les sorties du modèle sont converties sous la forme des informations qui seraient normalement transmises par les capteurs présents sur le UAV.

## Abstract

---

For this project, a UAV (unmanned aerial vehicle) was purchased with an autopilot to control it. The autopilot is composed of PID with gains scheduling. The main purpose of this project is to improve the autopilot performances. Therefore, this project aims to replace some of the PID with nonlinear control, based essentially on a physical model.

**Non classifié**

---

The first phase of the project consists in modeling and identifying the UAV. To do so, a nonlinear six-degree-of-freedom aircraft model available in a specialized Simulink library, the Aerosym library from Unmanned Dynamics, is used. To study the behavior of the aircraft when controlled by some nonlinear algorithms, these algorithms will be implemented in an autopilot connected to a computer running the aircraft model. So, a hardware-in-the-loop system has been developed to make some tests in the future phases of this project. This system is described here. More precisely it is explained how the autopilot command signals are converted under the form of the model inputs and how the data coming from the aircraft model are converted under the same form as the data transmitted by the sensors on the aircraft.

# Sommaire

---

L'objectif du projet est de modifier l'autopilote d'une plate forme aérienne existante pour le rendre plus performant. L'autopilote utilisé est le MP2028 et la plate forme aérienne est un UAV (véhicule aérien sans pilote) acheté de MicroPilot.

Le UAV de MicroPilot est le modèle XtraEasy2 conçu par Hangar 9, d'une étendue d'ailes d'environ 1.75 m et d'une longueur d'environ 1.40 m, pesant environ 3 Kg. Les actionneurs qu'il comporte sont une gouverne de direction, une gouverne de profondeur, des ailerons et l'étrangleur d'un moteur à hélice.

Le MP2028 utilise 11 régulateurs PID à gains variables pour assurer le contrôle du UAV. Il comporte des capteurs permettant d'obtenir la pression absolue, la vitesse de l'avion par rapport à l'air, les accélérations de l'avion selon trois de ses axes de même que la vitesse angulaire de l'avion en tangage, en lacet et en roulis. Également, le MP2028 est muni d'un système de positionnement global (GPS) lui permettant de connaître sa position géodésique de même que sa vitesse nord-sud, est-ouest et sa vitesse en altitude. Il comporte aussi trois sorties PWM (à modulation de largeur d'impulsion), chacune d'entre-elles est multiplexée et permet de contrôler 4 servos. Seulement l'une d'entre-elle est utilisée pour contrôler les actionneurs. Cette sortie PWM peut être utilisée sous différentes configurations, afin de contrôler les actionneurs de la façon désirée.

Le projet vise à remplacer certains des régulateurs PID du MP2028 par des commandes non-linéaires basées sur des modèles en grande partie physiques. La méthodologie suit l'approche classique utilisée dans le domaine de la commande automatique et divise donc le projet en trois phases. La phase 1 est la modélisation et l'identification des paramètres du système. La phase 2 est la conception et la comparaison d'algorithmes de contrôle non-linéaires et la phase 3 englobe les essais expérimentaux sur le UAV pour valider les simulations.

Ce rapport décrit les travaux effectués quant à l'élaboration d'un système «hardware in the loop», c'est-à-dire un système mettant en relation l'autopilote et un ordinateur simulant l'avion.

Afin de modéliser le comportement du UAV, un modèle Simulink non-linéaire à 6 degrés de liberté appartenant à la librairie AeroSym, soit une librairie créée par Unmanned Dynamics, est utilisé. Les travaux en cours consistent à connecter l'autopilote à un ordinateur (système «hardware in

the loop») qui doit simuler à l'aide du modèle de l'avion, le comportement de l'avion face aux diverses commandes transmises par l'autopilote. Ainsi, le modèle doit fournir à l'autopilote l'information équivalente à celle des capteurs du UAV. Le modèle doit également être en mesure de recevoir les commandes normalement transmises par l'autopilote aux servos de l'avion. Le développement d'un tel système est en cours de réalisation sur un autopilote MP2000, soit la version antérieure au MP2028. Les travaux sont très avancés et la conversion des données du simulateur sous la forme des informations qui seraient transmises par chacun des capteurs est détaillée dans ce rapport, de même que la conversion des commandes transmises par l'autopilote sous la forme des entrées du modèle.

# Executive

---

The purpose of this project is to modify the autopilot of an existing aerial platform to improve its performances. The autopilot used is the MP2028 and the aerial platform is a UAV (unmanned aerial vehicle) purchased from MicroPilot.

The purchased UAV is the XtraEasy2 developed by Hangar 9, with a wingspan of about 1.75 m, a length of about 1.40 m and weighting about 3 Kg. The actuators are a rudder, an elevator, two ailerons and propeller motor throttle. The MP2028 uses 11 PID regulators with gains scheduling to ensure the control of the UAV. It is equipped with sensors allowing the measurement of the absolute pressure, the UAV airspeed, its accelerations in three axes and its pitch, roll and yaw angular rates. Also, the MP2028 has a GPS to determine the UAV geodesic position and its nord, east and down velocities. It also comprises three PWM (pulse-width modulation) outputs, each of them allowing to control four servos. Only one is used to control the actuators. This PWM output can be used in different configurations, to control the actuators in the desired way.

This project aims to replace some MP2028 PID regulators by nonlinear control based in most part on a physical model. The methodology follows the classical approach used in process control and divides the project in three phases. Phase 1 is the identification of the system parameters to obtain a model. Phase 2 is the conception and comparison of the nonlinear control algorithms and Phase 3 consists in experimental tests on the UAV to validate the simulations.

This report describes the work done about the elaboration of a hardware-in-the-loop system. This system connects an autopilot with a computer simulating the aircraft.

In order to model the UAV, a nonlinear six-degree-of-freedom Simulink model from the Aerosym library, a library created by Unmanned Dynamics, is used. Presently, the autopilot is almost connected to a computer running the aircraft model (hardware-in-the-loop system) that will simulate the aircraft behavior in function of diverse commands sent by the autopilot. Here the model has to provide the same information as the sensors did within the vehicle. Also, the model has to be able to receive the commands transmitted by the autopilot, sent normally to the servos controlling the UAV actuators. One hardware-in-the-loop system using a MP2000 autopilot, the previous

version of the MP2028, is almost finished. This report shows the system development, i.e. the model output data conversion under the same form as the information provided by the vehicle sensors and the autopilot command conversion under the form of the model inputs.



# Table des matières

<b>1</b>	<b>Introduction</b>	<b>15</b>
<b>2</b>	<b>Le système expérimental : autopilote - UAV</b>	<b>18</b>
<b>3</b>	<b>Le système «hardware in the loop» : autopilote - simulateur du UAV</b>	<b>20</b>
3.1	La lecture des signaux de commande . . . . .	21
3.2	La reproduction des signaux des accéléromètres . . . . .	22
3.3	La reproduction des signaux des capteurs de vitesse angulaire	25
3.4	La reproduction des signaux du capteur de la vitesse par rapport à l'air . . . . .	28
3.5	La reproduction du signal du capteur de l'altitude . . . . .	37
3.6	La reproduction des paquets d'information du système de positionnement global (GPS) . . . . .	44
<b>4</b>	<b>Conclusion</b>	<b>49</b>

# Liste des tableaux

1	Description des symboles . . . . .	12
1	Description des symboles (suite) . . . . .	13
1	Description des symboles (suite) . . . . .	14
2.1	La configuration des servos du UAV . . . . .	18
2.2	Les capteurs utilisés par l'autopilote MP2000 . . . . .	19
3.1	Les paquets TSIP transmis par le GPS : Fréquence et contenu	46
3.2	Les paquets TSIP : Les types de données transmises . . . . .	46
3.3	Les paquets TSIP : Le paquet 4Ah . . . . .	47
3.4	Les paquets TSIP : Le paquet 56h . . . . .	47
3.5	Les paquets TSIP : Le paquet 41h . . . . .	47
3.6	Les paquets TSIP : Le paquet 46h . . . . .	48
3.7	Les paquets TSIP : Le paquet 4Bh . . . . .	48
3.8	Les paquets TSIP : Le paquet 6Dh . . . . .	48
3.9	Les paquets TSIP : Le paquet 82h . . . . .	48

# Table des figures

1.1	Le système expérimental : autopilote et UAV . . . . .	16
1.2	Le système «hardware in the loop» : autopilote et simulateur du UAV . . . . .	17
3.1	Le système «hardware in the loop» détaillé . . . . .	21
3.2	La reproduction des sorties générées par les accéléromètres (capteur ADXL202) . . . . .	24
3.3	La reproduction des sorties générées par les capteurs de vi- tesses angulaires (capteurs CG16D) . . . . .	27
3.4	La sortie en tension du capteur de la vitesse par rapport à l'air (capteur MPX10) selon ses spécifications . . . . .	29
3.5	Les étapes générales de la reproduction de la sortie générée par le capteur MPX10 . . . . .	29
3.6	La reproduction de la tension générée par le capteur MPX10 faite entièrement à l'intérieur de Simulink . . . . .	31
3.7	La vitesse par rapport à l'air en fonction de la pression dyna- mique . . . . .	32
3.8	La vitesse par rapport à l'air en fonction de la tension générée par le capteur MPX10 . . . . .	32
3.9	La reproduction de la tension générée par le capteur MPX10 avec ajout d'un circuit imprimé afin d'augmenter la résolution du signal . . . . .	35
3.10	Le circuit de l'interface pour le capteur MPX10 . . . . .	36
3.11	La sortie en tension du capteur d'altitude (capteur MPX4115A) selon ses spécifications . . . . .	37
3.12	Les étapes générales de la reproduction de la sortie générée par le capteur MPX4115A . . . . .	38
3.13	La reproduction de la tension générée par le capteur MPX4115A faite entièrement à l'intérieur de Simulink . . . . .	41

3.14	La reproduction de la tension générée par le capteur MPX4115A avec ajout d'un circuit afin d'augmenter la résolution du signal	42
3.15	Le circuit de l'interface pour le capteur MPX4115A . . . . .	43

TAB. 1: Description des symboles

<b>Symboles</b>	<b>Description</b>
<b>Le système expérimental : autopilote-UAV</b>	
$a_x$	Accélération de l'avion exprimée dans le référentiel du corps de l'avion (selon l'axe $X_b$ ) et causée par l'action des forces autres que la gravité
$a_y$	Accélération de l'avion dans le référentiel du corps de l'avion (selon l'axe $Y_b$ ) et causée par l'action des forces autres que la gravité
$Alt$	Altitude en mètres fournie par le modèle au-dessus d'un point de référence
$Lat$	Latitude géodésique
$Lon$	Longitude géodésique
$P$	Composante de la vitesse angulaire autour de l'axe $X_b$ (roulis)
$P_{abs}$	Pression atmosphérique selon l'altitude provenant d'une table de données dans le modèle AeroSym
$Q$	Composante de la vitesse angulaire autour de l'axe $Y_b$ (tanguage)
$R$	Composante de la vitesse angulaire autour de l'axe $Z_b$ (lacet)
$Th$	Poussée du moteur (valeur entre 0 et 1)
$V_a$	Amplitude de la vitesse de l'avion par rapport à l'air
$V_{nord}$	Composante de la vitesse vers le nord
$V_{est}$	Composante de la vitesse vers l'est
$V_{bas}$	Composante de la vitesse vers le bas
$\delta_e$	Angle de déflexion de la gouverne de profondeur
$\delta_r$	Angle de déflexion de la gouverne de direction
$\delta_a$	Angle de déflexion des ailerons
<b>Les signaux de commandes</b>	
$n_{mes}$	Nombre de coups d'horloge correspondant à la largeur d'impulsion lue
$F_t$	Fréquence du signal PWM à lire
$F_{clk}$	Fréquence de l'horloge de la carte NI6601

TAB. 1: Description des symboles (suite)

<b>L'accéléromètre</b>	
$A(g)$	Accélération de l'avion mesurée en unité de g (constante gravitationnelle)
$D_x$	Rapport de la largeur d'impulsion sur sa période («duty cycle») du signal provenant de l'accéléromètre correspondant à l'accélération selon l'axe $X_b$ de l'avion
$D_y$	Rapport de la largeur d'impulsion sur sa période («duty cycle») du signal provenant de l'accéléromètre et correspondant à l'accélération selon l'axe $Y_b$ de l'avion
$g$	Constante gravitationnelle
$g_{bx}'$	Composante de la gravité selon l'axe $X_b$ du référentiel du corps de l'avion
$g_{by}'$	Composante de la gravité selon l'axe $Y_b$ du référentiel du corps de l'avion
$X_b, Y_b, Z_b$	Axes du référentiel du corps de l'avion
<b>Les capteurs de vitesse angulaire</b>	
$m_{(CG16D)}$	Pente de la droite exprimant la tension de sortie du capteur CG16D en fonction de la vitesse angulaire mesurée
$Vo_{(CG16D)}$	Origine de la droite exprimant la tension de sortie du capteur CG16D en fonction de la vitesse angulaire mesurée
$Vout_{(CG16D)}$	Tension de sortie du capteur de vitesse angulaire CG16D
$w$	Vitesse angulaire de l'avion selon l'axe mesuré par le capteur CG16D (quelconque)
<b>Le capteur de vitesse par rapport à l'air</b>	
$F_{(MPX10)}$	Facteur multiplicatif utilisé dans la reproduction du signal du capteur MPX10
$m_{(MPX10)}$	Pente de la droite exprimant la tension de sortie du capteur MPX10 en fonction de la pression dynamique mesurée
$P_0$	Pression totale sur un corps
$P_S$	Pression statique de l'air

TAB. 1: Description des symboles (suite)

$V_{o(MPX10)}$	Origine de la droite exprimant la tension de sortie du capteur MPX10 en fonction de la pression dynamique mesurée
$V_{out(MPX10)}$	Tension de sortie du capteur de vitesse par rapport à l'air MPX10
$V_{ref(MPX10)}$	Tension de référence par rapport à laquelle le capteur MPX10 donne sa tension de sortie $V_{out(MPX10)}$
$\rho$	Masse volumique de l'air
<b>Le capteur de l'altitude</b>	
$h$	Altitude en pieds au-dessus d'un point de référence utilisée dans l'équation 3.12
$F_{(MPX4115A)}$	Facteur multiplicatif utilisé dans la reproduction du signal du capteur MPX4115A
$m_{(MPX4115A)}$	Pente de la droite exprimant la tension de sortie du capteur MPX4115A en fonction de la pression $P_S$ mesurée
$P_{s0}$	Pression atmosphérique au point de référence lors de l'évaluation de la pression par l'équation 3.12
$V_{o(MPX4115A)}$	Origine de la droite exprimant la tension de sortie du capteur MPX4115A en fonction de la pression $P_S$ mesurée
$V_{out(MPX4115A)}$	Tension de sortie du capteur d'altitude MPX4115A

# Chapitre 1

## Introduction

Les tests en vols sont effectués avec un autopilote MP2028. L'autopilote utilise 11 PID pour contrôler l'avion. Il reçoit de l'information quant aux mouvements de l'avion grâce à différents capteurs. Les entrées de l'autopilote sont donc les signaux fournis par ces capteurs. À partir de cette information, l'autopilote décide alors de l'action à prendre et transmet aux servos contrôlant les différents actionneurs de l'avion les signaux de contrôle appropriés. Le système expérimental est illustré à la figure 1.1 et est décrit dans la section 2 page 18.

Avant de tester les algorithmes de contrôle sur l'avion réel, ils seront testés sur un simulateur à l'aide d'un système «hardware in the loop». Pour ce faire, un ordinateur simulera en temps réel le comportement de l'avion à l'aide d'un modèle. Cet ordinateur sera connecté à l'autopilote (sans capteurs) via des cartes d'acquisition. Ainsi, les signaux de sortie du simulateur remplaceront ceux des capteurs réels de l'avion et seront transmis à l'autopilote. De même, les signaux de commande formulés par l'autopilote pour les actionneurs de l'avion seront transmis aux entrées de l'ordinateur qui simulera les actionneurs et leurs effets sur l'avion.

Le système «hardware in the loop» est illustré à la figure 1.2 et est décrit dans la section 3 page 20. Le modèle simule le comportement de l'avion. Il prend donc comme entrée les commandes aux actionneurs sous forme d'angle pour le gouvernail de direction, le gouvernail de profondeur et les ailerons et de fraction pour la poussée du moteur. Ainsi, l'autopilote transmet à une carte d'acquisition les signaux PWM destinés aux ailerons, au gouvernail de profondeur, au gouvernail de direction et au servo contrôlant la poussée du moteur de l'avion. Ces signaux PWM sont ensuite traités afin d'en déduire



les mouvements que chaque actionneur doit accomplir (angle ou fraction de poussée), afin d'être fournis au modèle qui simule alors la réaction de l'avion face à ces commandes. Le comportement de l'avion, soit sa réaction aux commandes, se traduit par plusieurs variables dans le modèle dont les vitesses angulaires de l'avion, ses accélérations linéaires, son altitude, sa vitesse par rapport à l'air et sa position dans un modèle géodésique. Ces variables sont les sorties du modèle. L'ordinateur doit pouvoir fournir à l'autopilote les données de ces variables, sous la même forme que celles des capteurs avec lesquels l'autopilote fonctionne généralement. Il s'agit donc de transformer les données de sortie du modèle de façon à retrouver la largeur d'impulsion (dans le cas d'un signal PWM) ou la tension (dans le cas d'un signal analogique) que génèrerait chacun des capteurs dans pareille situation.

Le montage «hardware in the loop» est élaboré à partir de l'autopilote MP2000, soit la version antérieure au MP2028. Cependant, il y a peu de différence entre les deux autopilotes. Les deux autopilotes ont la même configuration pour les servos, les mêmes PID et les mêmes capteurs, à l'exception des capteurs de vitesse angulaire.

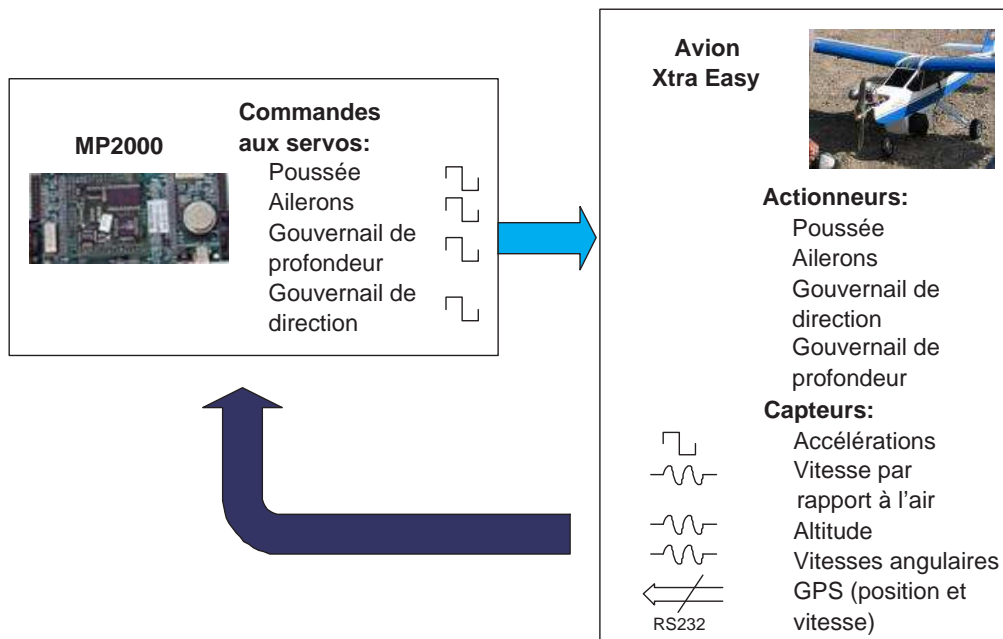


FIG. 1.1 – Système expérimental : autopilote et UAV

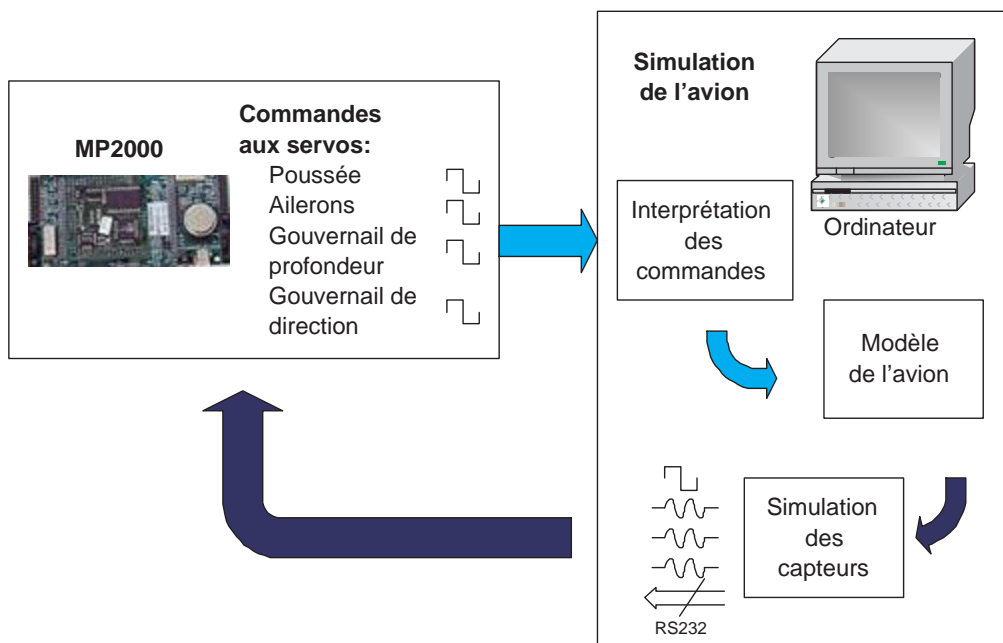


FIG. 1.2 – Système «hardware in the loop» : autopilote et simulateur du UAV

## Chapitre 2

# Le système expérimental : autopilote - UAV

Les sorties de l'autopilote sont normalement destinées aux actionneurs. L'avion comporte quatre actionneurs (il n'a pas de volets «flaps»). Chaque actionneur est contrôlé par un servo connecté à l'autopilote. La configuration des signaux destinés aux servos sur l'autopilote est en mode normal, décrit par les colonnes 1 et 2 du tableau 2.1 (voir la documentation technique de l'autopilote pour connaître les autres configurations possibles [2]). Les relations entre chaque servo, l'actionneur qu'il commande et l'entrée du modèle utilisé par le simulateur qui lui est associée sont présentées dans le tableau 2.1. Ainsi, l'autopilote transmet aux quatre servos de l'autopilote un signal à modulation de largeur d'impulsion (PWM).

TAB. 2.1 – La configuration des servos du UAV

<b>Servos (connecteurs sur l'autopilote)</b>	<b>Signaux</b>	<b>Entrées correspondantes du simulateur</b>
Servo 1 (S1)	Déflexion des ailerons	$\delta_a$
Servo 2 (S2)	Déflexion de la gouverne de profondeur	$\delta_e$
Servo 3 (S3)	Déflexion de la gouverne de direction	$\delta_r$
Servo 4 (S4)	Poussée	$Th$

Les entrées de l'autopilote sont normalement excitées par les signaux pro-

venant des capteurs. Les capteurs présents sur le UAV de même que les sorties correspondantes du modèle utilisées par le simulateur sont présentés dans le tableau 2.2.

L'avion est souvent représenté dans un référentiel qui lui est fixe, dont l'origine se situe au centre de gravité de celui-ci. Il s'agit du référentiel du corps de l'avion. L'axe  $X_b$  de ce référentiel pointe vers le nez de l'avion, l'axe  $Y_b$  vers la pointe de l'aile droite de l'avion et l'axe  $Z_b$  est perpendiculaire aux deux autres axes et pointe vers le bas. Les capteurs permettent entre autres de mesurer les accélérations selon les axes  $X_b$  et  $Y_b$  de même que les vitesses angulaires autour de chacun des axes de ce référentiel.

TAB. 2.2 – Les capteurs utilisés par l'autopilote MP2000

Capteurs	Signaux	Sorties du simulateur correspondantes
Analog Devices ADXL202	Accélérations en x et en y	$a_x + g_{bx}'$ et $a_y + g_{by}'$
Motorola MPX10	Vitesse par rapport à l'air	$V_a$
Motorola MPX4115A	Pression absolue	$P_{abs}$
Nec/Tokin CG-16D	Vitesse angulaire (roulis)	$P$
Nec/Tokin CG-16D	Vitesse angulaire (tangage)	$Q$
Nec/Tokin CG-16D	Vitesse angulaire (lacet)	$R$
Lassen TM SKII GPS	Position géodésique et vitesse linéaire dans le référentiel NED (nord-est-bas)	$(Lat, Lon, Alt_{HAE})$ et $(V_{nord}, V_{est}, V_{bas})$

# Chapitre 3

## Le système «hardware in the loop» : autopilote - simulateur du UAV

Pour créer le simulateur, la librairie XPC Target et les fonctionnalités de Real Time Workshop de Simulink sont utilisées. Le modèle de l'avion est compilé et téléchargé dans un ordinateur qui exécute le modèle en temps réel. Trois cartes d'acquisitions sont installées dans l'ordinateur et sont utilisées pour communiquer avec l'autopilote.

Le système «hardware in the loop» est présenté plus en détail à la figure 3.1. Deux cartes NI6601 permettent de lire et d'écrire à partir de l'ordinateur des signaux numériques (entre autres des signaux à modulation de largeur d'impulsion). Elles sont utilisées pour reproduire les signaux des accéléromètres et pour lire les signaux provenant de l'autopilote et destinés normalement aux servos de l'avion. La carte NI6703 permet de lire et d'écrire à partir de l'ordinateur des signaux analogiques. Elle permet de reproduire les signaux des capteurs de vitesses angulaires, d'altitude et de vitesse par rapport à l'air. Ces trois types de capteurs fournissent tous les trois une tension proportionnelle à leur lecture.

Afin de reproduire les signaux et tensions des capteurs, les sorties du modèle sont d'abord traitées (toujours à l'intérieur de Matlab/Simulink). Dans le cas des accéléromètres, les largeurs d'impulsion des signaux PWM correspondantes aux accélérations linéaires que les accéléromètres fourniraient sont calculées. Ces signaux PWM peuvent alors être transmis à l'autopilote par les cartes d'acquisition sans plus de traitements. Dans le cas des

capteurs analogiques, soit les capteurs d'altitude, de vitesse par rapport à l'air et des vitesses angulaires, ils fournissent une tension en fonction de ce qu'ils mesurent. La carte d'acquisition NI6703 peut transmettre une tension à l'intérieur de  $\pm 10V$  et a 16 bits de résolution, ce qui représente une variation de tension minimale possible de 0,3mV. Bien que sur de grandes plages de données cette résolution soit suffisante pour représenter le comportement des capteurs analogiques, dans certains cas la variation de la tension fournie par certains capteurs est plus faible.

Ainsi, pour ce qui est des sorties analogiques (pression absolue et vitesse par rapport à l'air), des cartes électroniques ont été élaborées afin d'optimiser l'usage des cartes d'acquisition, bien qu'il soit possible dans une certaine mesure de faire l'implantation du système sans circuit d'amélioration.

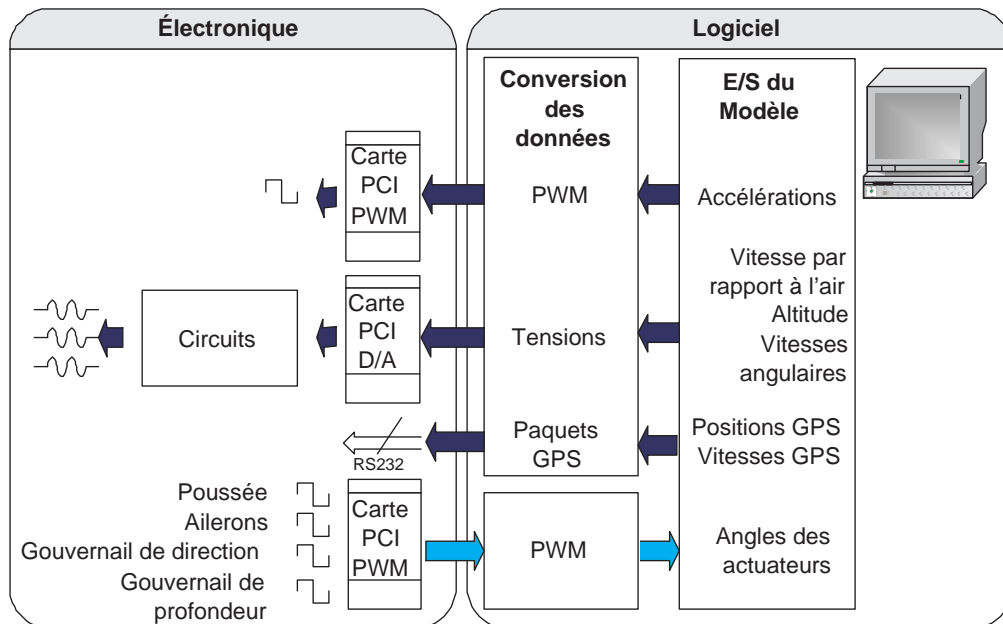


FIG. 3.1 – Le système «hardware in the loop» détaillé

### 3.1 La lecture des signaux de commande

L'autopilote peut accueillir jusqu'à trois cartes de servos et transmet trois signaux multiplexés, un signal pour chacune des cartes de servos. Aussi, une

carte contient huit servos. Dans le présent projet, une seule carte de servos est utilisée et donc, un seul signal multiplexé est utilisé. Ce signal contient les commandes pour les huit servos disponibles. Quatre servos sont utilisés pour les actionneurs du UAV. La table 2.1 indique le servo utilisé pour chaque actionneur).

La période d'un signal multiplexé (fournissant les commandes pour huit servos) a été mesurée à environ 23.4ms. La période d'un signal destiné à un servo est d'environ 2.92ms.

La carte d'acquisition NI6601 permet de lire le nombre de coups d'horloge  $n_{mes}$  correspondants à la largeur de l'impulsion du signal lue. Ainsi, pour un signal destiné à un servo, le rapport de la largeur d'impulsion sur la période du signal est obtenue par l'équation suivante :

$$D_w = n_{mes} \times \frac{F_t}{F_{clk}} \quad (3.1)$$

où  $D_w$  est le rapport de la largeur d'impulsion d'une commande,  $F_t$  est la fréquence du signal PWM à lire ( $F_t \approx 342.4Hz$ ) et  $F_{clk}$  est la fréquence de l'horloge, soit de 20MHz.

La position du servo selon le rapport  $D_w$  lu est estimée par une relation linéaire. La valeur maximal du rapport  $D_w$  est associé à la position maximale du servo et la valeur minimale du rapport  $D_w$ , à la position minimale du servo. Le largeur d'impulsion varie de 1ms à 2ms selon les spécifications fournies par MicroPilot.

## 3.2 La reproduction des signaux des accéléromètres

Les deux accéléromètres de l'ADXL202 indiquent les accélérations selon les directions  $X_b$  et  $Y_b$  dans le référentiel du corps de l'avion. À noter que ces accéléromètres rendent compte également de l'accélération gravitationnelle. L'ADXL202 renvoie donc deux signaux PWM, soit un signal pour chaque direction, dont le rapport de la largeur d'impulsion sur la période («duty cycle»)  $D$  est proportionnel à l'accélération. Par exemple, le signal de sortie en x est de 0.5 lorsqu'aucune accélération n'est présente et augmente de  $\pm 12.5\%$  pour une accélération de 1 g (g étant la constante de temps gravitationnelle), selon le sens de l'accélération par rapport à l'axe du capteur. L'accélération en unité de g (soit  $\approx 9.81m/s^2$ )  $A(g)$  de chacune des sorties

$x$  et  $y$  est donnée en fonction du rapport  $D$  de la largeur d'impulsion sur la période de la façon suivante [1] :

$$A(g) = \frac{D - 0.5}{0.125} \quad (3.2)$$

Pour reproduire le comportement d'un capteur, le simulateur doit transmettre la valeur du rapport  $D$  la largeur d'impulsion sur la période. La reproduction de ce signal se fait à l'intérieur de Simulink, tel qu'indiqué à la figure 3.2. Le modèle calcule les accélérations et fournit  $a_x$  et  $a_y$ , soit les accélérations de l'avion dues aux forces à l'exception de la gravité. Pour tenir compte de la gravité, les accélérations de l'avion  $g_{bx}'$  et  $g_{by}'$  dues à cette force sont calculées également. Toutes ces accélérations sont calculées dans le référentiel du corps de l'avion et sont en  $m/s^2$ . Pour chacun des axes, les accélérations sont additionnées afin d'obtenir l'accélération totale de l'avion à l'intérieur de chaque axe. Cette valeur, qui est équivalente à la mesure faite par le capteur selon son axe, est traitée dans Matlab/Simulink afin d'en déduire le rapport  $D$  de la largeur d'impulsion sur la période à transmettre à la carte d'acquisition numérique. Pour ce faire, les équations 3.3 et 3.4 sont utilisées :

$$D_x = 0.5 + (0.125) \frac{a_x + g_{bx}'}{g} \quad (3.3)$$

$$D_y = 0.5 + (0.125) \frac{a_y + g_{by}'}{g} \quad (3.4)$$

À noter que dans ces équations, pour convertir l'accélération totale en unité de  $g$ , celle-ci est divisée par la constante gravitationnelle.

Les valeurs  $D_x$  et  $D_y$  sont ensuite transmises à la carte NI6601 afin de créer les deux signaux PWM correspondants. Ces signaux sont alors directement transmis à l'autopilote.



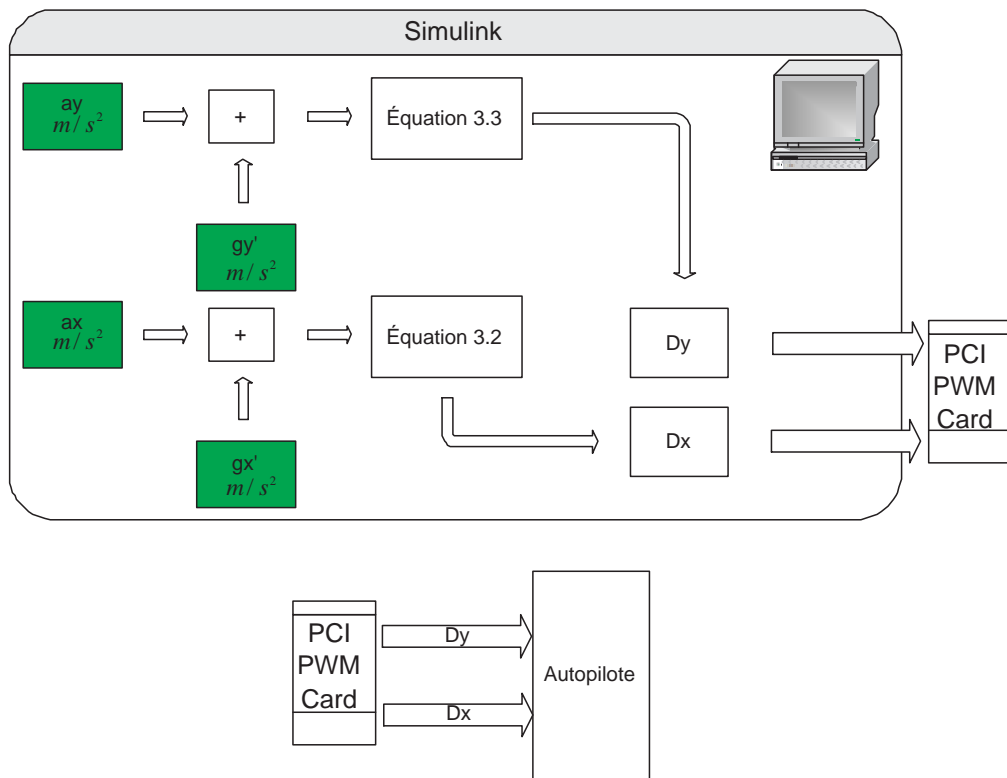


FIG. 3.2 – La reproduction des signaux d'accélération

### 3.3 La reproduction des signaux des capteurs de vitesse angulaire

Afin de mesurer les vitesses angulaires de l'avion selon les trois axes du référentiel du corps de l'avion  $X_b$ ,  $Y_b$  et  $Z_b$ , trois capteurs («rate gyro») sont présents sur l'autopilote et orientés selon chacun de ces trois axes. Sur l'autopilote MP2000, les capteurs utilisés sont des CG-16D. La tension transmise  $V_{out(CG16D)}$  par ce capteur est proportionnelle à la vitesse angulaire  $w$  en deg/s autour de l'axe mesuré et s'exprime selon une droite de pente  $m_{(CG16D)}$  avec une tension à l'origine  $V_{o(CG16D)}$  (pour plus de détails voir les spécifications du capteur [6]).

$$V_{out(CG16D)} = m_{(CG16D)} \times (w) + V_{o(CG16D)} \quad (3.5)$$

La reproduction des tensions des capteurs de vitesses angulaires est représentée à la figure 3.3. À noter que dans cette figure, les indices des variables dénotant qu'elles s'appliquent pour le capteur CG-16D (ex :  $V_{out(CG16D)}$ ) ne sont pas indiqués pour alléger la présentation. Le modèle fournit les vitesses angulaires en roulis P (autour de l'axe  $X_b$ ), en tangage Q (autour de l'axe  $Y_b$ ) et en lacet (autour de l'axe  $Z_b$ ). Afin de reproduire les tensions fournies par chacun des trois capteurs CG-16D, la tension à générer pour chacune des vitesses angulaires  $P$ ,  $Q$ , et  $R$  est calculée à partir de l'équation 3.5. Cette équation est implantée entièrement dans Simulink pour chacune des vitesses angulaires  $P$ ,  $Q$ , et  $R$ . Pour chacune des vitesses angulaires, la tension ainsi calculée est transmise directement à l'autopilote via une carte d'acquisition. La tension de sortie du capteur CG16D est donnée par rapport à une tension de référence  $V_{refCG16D}$ , également transmise à l'autopilote. La résolution de ce capteur tel que mentionnée dans les spécifications est de 1.1mV/(deg/s), ce qui semble suffisant par rapport à la résolution de la carte, soit 0.3mV.

Les vitesses angulaires et les angles en tangage et en roulis déduits par l'autopilote ne proviennent pas uniquement des signaux fournis par les capteurs de vitesses angulaires. En effet, bien que la vitesse angulaire et l'angle en lacet soient déduits uniquement à partir du capteur de vitesse angulaire pour cet axe, cela n'est pas le cas pour les vitesses angulaires et les angles pour les deux autres axes du référentiel du corps de l'avion. Pour ces deux axes, les vitesses angulaires et les angles semblent déduits de l'addition de deux signaux traités. Lors d'une rotation angulaire en roulis par exemple, le capteur de la vitesse angulaire en roulis détecte la rotation et l'accéléromètre y détecte une

variation dans l'accélération (due à la composante  $y$  de l'accélération gravitationnelle dans le référentiel du corps de l'avion qui varie lors de la rotation). En utilisant le système «hardware in the loop», si un signal indiquant aucune accélération est transmis à l'entrée de l'accéléromètre  $y$  de l'autopilote et qu'un signal indiquant une rotation angulaire d'une durée limitée (impulsion) est transmis à l'entrée du capteur de la vitesse angulaire en roulis de l'autopilote, une fois la rotation terminée, l'angle en roulis calculé par l'autopilote revient rapidement à sa valeur de départ, alors qu'il devrait rester constant, à la valeur de l'angle obtenu après la rotation. Ainsi, la vitesse angulaire en roulis une fois intégrée semble filtrée par un filtre passe-bas. Toujours en utilisant le système «hardware in the loop», si un signal de rotation angulaire indiquant aucune rotation est transmis à l'entrée du capteur de la vitesse angulaire en roulis de l'autopilote et qu'un signal indiquant une accélération en  $y$  correspondant à une rotation en roulis donnée est transmise à l'autopilote, l'angle varie lentement pour atteindre sa valeur finale à la fin de la rotation et reste constant. Il semble que l'angle en roulis en régime permanent soit déduit de l'accélération en  $y$  lue par l'autopilote et que le signal soit filtré par un filtre passe-haut.

Ainsi, lors d'une réelle rotation en roulis, le signal transmis par le capteur de vitesse angulaire en roulis serait utilisé pour mesurer la période transitoire dans le calcul de l'angle en roulis, alors que le signal provenant de l'accélération en  $y$  serait utilisé pour en déduire la valeur en régime permanent. Les mêmes observations et déductions ont été faites pour la vitesse angulaire et l'angle en tangage, provenant des signaux fournis par le capteur de la vitesse angulaire en tangage et de l'accéléromètre en  $x$ .

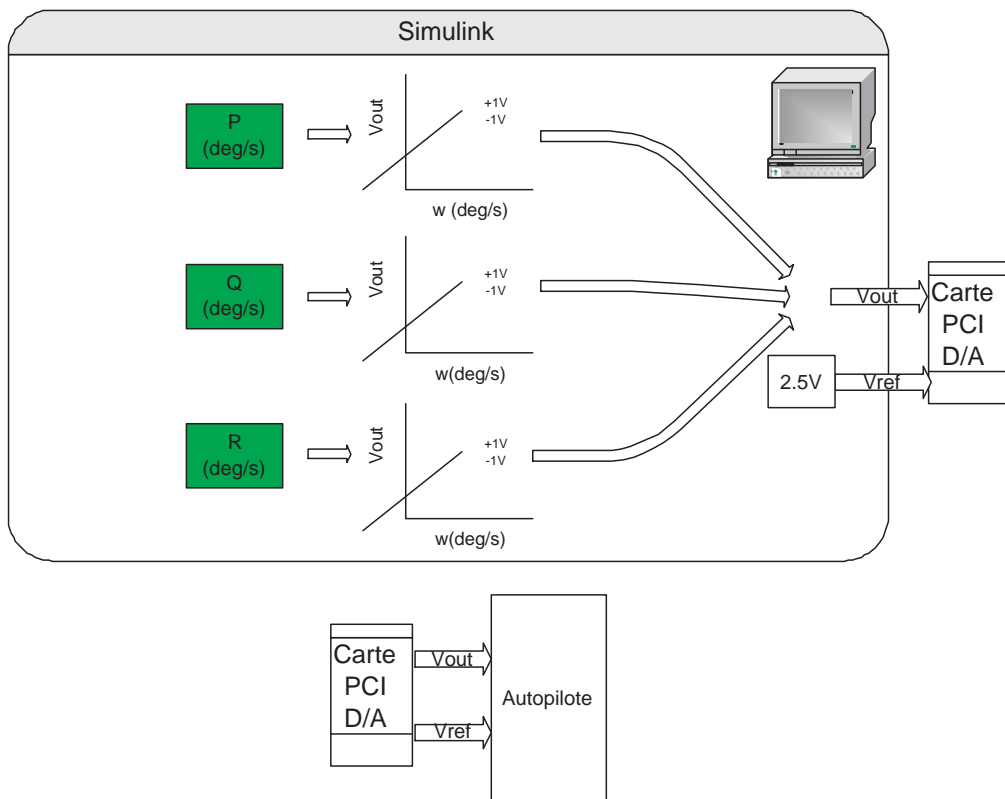


FIG. 3.3 – La reproduction de la tension générée par les trois capteurs de vitesses angulaires

### 3.4 La reproduction des signaux du capteur de la vitesse par rapport à l'air

Selon l'équation de Bernoulli 3.6 [5], la pression totale  $P_0$  (en Pa) qu'un corps subit est la somme de la pression statique  $P_s$  (en Pa) due à l'air ambiant, et de la pression dynamique  $\rho \frac{V_a^2}{2}$  due à sa vitesse par rapport à l'air  $V_a$  en m/s ( $\rho$  étant la masse volumique de l'air en  $kg/m^3$ ).

$$P_0 - P_s = \rho \frac{V_a^2}{2} \quad (3.6)$$

Le capteur MPX10 permet à l'autopilote d'obtenir la vitesse par rapport à l'air. Ce capteur mesure en fait la pression dynamique, c'est-à-dire la différence entre la pression statique et la pression totale exercée par l'air sur le capteur lorsque l'avion est en cours de vol. Il renvoie alors une tension proportionnelle à la différence de pression  $P_0 - P_s$  mesurée. La tension générée (en V) par le capteur en fonction de la différence de pression  $P_0 - P_s$  (en KPa) est illustrée à la figure 3.4 (selon les spécifications du capteur [3]) et peut être représentée par l'équation 3.7. En effet, La tension  $V_{out(MPX10)}$  que le capteur renvoie s'exprime par une droite de pente  $m_{(MPX10)}$  avec une tension à l'origine  $V_{o(MPX10)}$ .

$$V_{out(MPX10)} = m_{(MPX10)} \times (P_0 - P_s) + V_{o(MPX10)} \quad (3.7)$$

La sortie du modèle de l'avion est en fait la vitesse par rapport à l'air  $V_a$ . Les étapes pour reproduire le comportement du capteur sont représentées à la figure 3.5. À noter que dans les figures de cette section, les indices des variables dénotant qu'elles s'appliquent pour le capteur MPX10 (ex :  $V_{out(MPX10)}$ ) ne sont pas indiqués pour alléger la présentation. Pour reproduire le comportement du capteur, le simulateur doit fournir la valeur de la différence de pression  $P_0 - P_s$ . Ainsi, à partir de  $V_a$ , la différence de pression  $P_0 - P_s$  est calculée avec l'équation de Bernoulli 3.6. Puis, la différence de pression  $P_0 - P_s$  (en Pa puisque l'équation est en unités SI) est convertie en KPa. La tension correspondante normalement fournie par le capteur est ensuite déterminée à partir de l'équation de la droite du capteur (équation 3.7).

Il est possible d'effectuer toute la transformation de la vitesse  $V_a$  vers la tension à transmettre à l'autopilote à l'intérieur de Matlab/Simulink en implantant simplement l'équation 3.7, tel qu'illustré à la figure 3.6. La tension

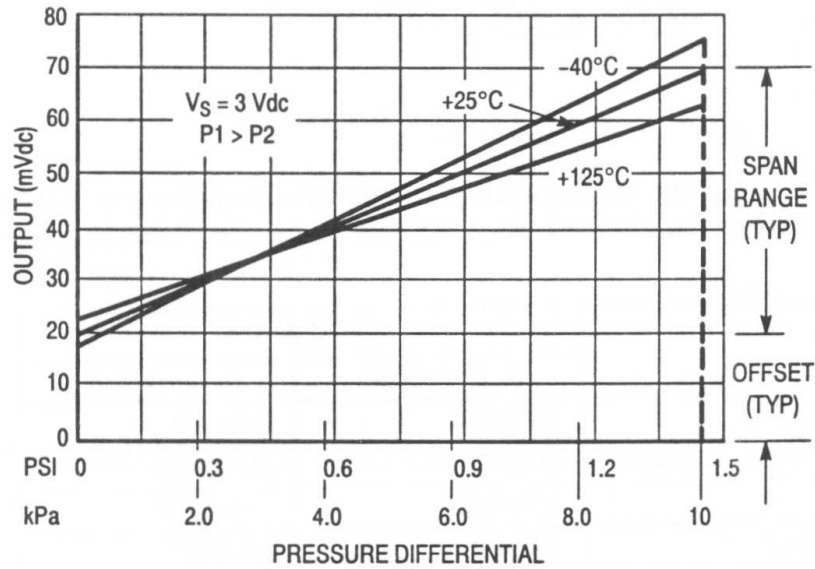


FIG. 3.4 – La sortie en tension du capteur MPX10 en fonction de la différence de pression mesurée selon ses spécifications[3]

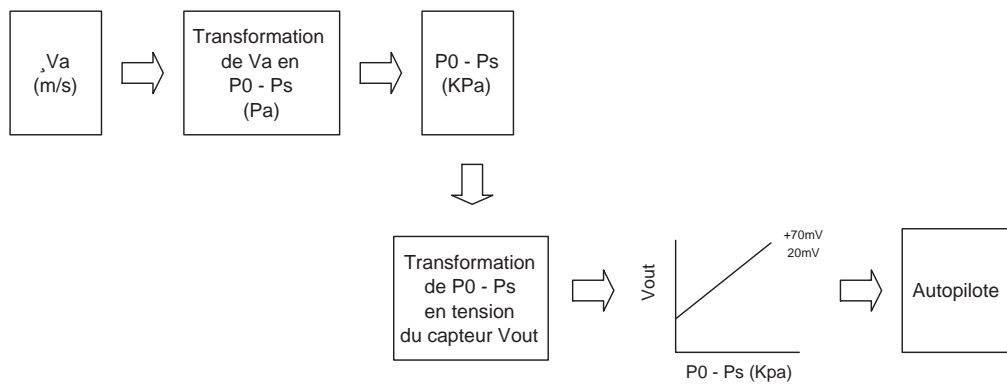


FIG. 3.5 – Les étapes de transformation de la vitesse par rapport à l'air  $V_a$  du modèle en la tension normalement générée par le capteur MPX10

$V_{out(MPX10)}$  obtenue est alors dirigée vers la sortie de la carte d'acquisition qui est directement connectée à l'autopilote. Le capteur donne en fait sa tension  $V_{out(MPX10)}$  par rapport à une tension de référence  $V_{refMPX10} \approx 2.5V$ . Cette tension de référence est donc également transmise à l'autopilote. Dans certaines situations, il y a un manque de précision quand aux vitesses lues par l'autopilote, surtout lorsque les vitesses sont faibles. Par exemple, l'autopilote ne voit que 0 m/s lorsqu'il lui est transmis des valeurs de tension correspondant à des vitesses de 0 à 5 m/s. Aussi, l'autopilote ne voit que 10 m/s lorsqu'il lui est transmis des vitesses de 5 à 10 m/s. Ainsi, la résolution de la carte d'acquisition n'est pas tout à fait assez élevée comparativement à celle du capteur mais s'améliore lorsque la vitesse  $V_a$  augmente. Ceci s'explique par le fait que la vitesse par rapport à l'air est proportionnelle à la racine carrée de la différence de pression  $P_0 - P_s$  (et donc la tension générée par le capteur), tel qu'illustré aux figures 3.7 et 3.8. Ainsi, l'autopilote reçoit la tension du capteur proportionnelle à la différence de pression  $P_0 - P_s$  et calcule la vitesse par rapport à l'air. Il doit effectuer le calcul suivant :

$$V_a = \sqrt{\frac{2(P_0 - P_s)}{\rho}} \quad (3.8)$$

Sur les graphiques il est possible de constater qu'à faible vitesse  $V_a$ , une petite variation de la différence de pression  $P_0 - P_s$  (et de la tension générée par le capteur) est interprétée par une assez grande variation de la vitesse par rapport à l'air. Il est également possible de vérifier la faible variation de la tension à de faibles vitesses  $V_a$  en calculant la dérivée de la tension  $V_{out(MPX10)}$  en fonction de la vitesse  $V_a$ . Selon l'équation 3.7 :

$$V_{out(MPX10)} = m_{(MPX10)} \times (P_0 - P_s) + V_{o(MPX10)}$$

En estimant que  $m_{(MPX10)} = 5mV/Kpa$  et que  $V_{o(MPX10)} = 20mV$ , la sortie du capteur en mV est donnée par

$$V_{out(MPX10)} = 5 \times (P_0 - P_s) + 20$$

En substituant  $P_0 - P_s$  par son expression en fonction de la vitesse  $V_a$  à l'aide de l'équation de Bernoulli 3.6 (en divisant par 1000 pour tenir compte du fait que l'équation 3.6 fournit la différence de pression  $P_0 - P_s$  en Pa et que la droite du capteur de l'équation 3.7 utilise  $P_0 - P_s$  en KPa), l'expression de la sortie du capteur  $V_{out(MPX10)}$  en mV en fonction de la vitesse  $V_a$  est obtenue :

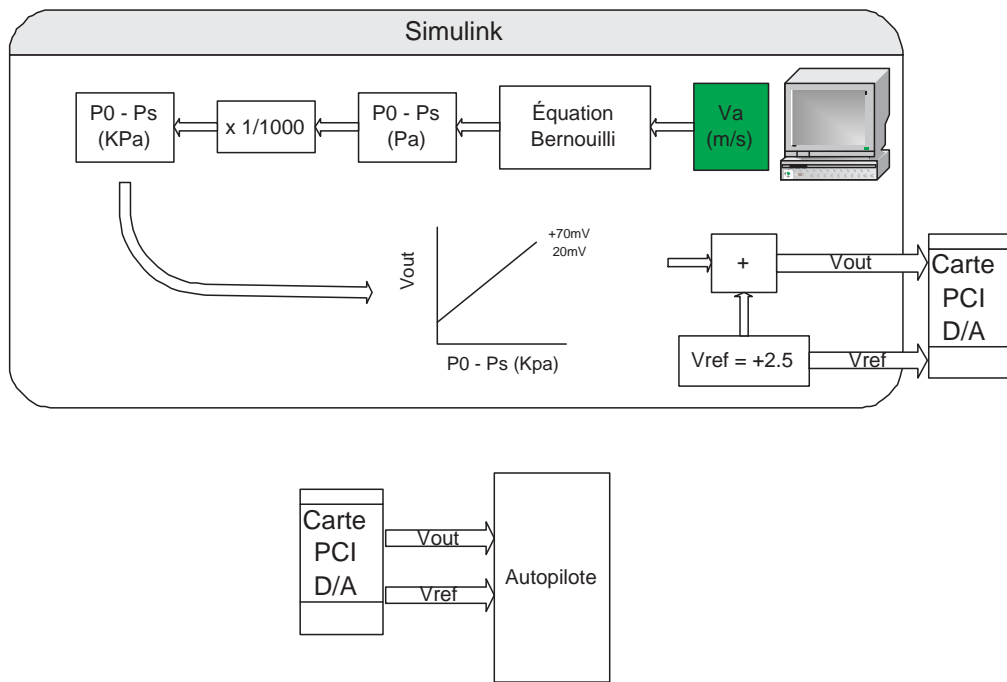


FIG. 3.6 – La reproduction de la tension lorsqu'elle est entièrement faite à l'intérieur de Simulink



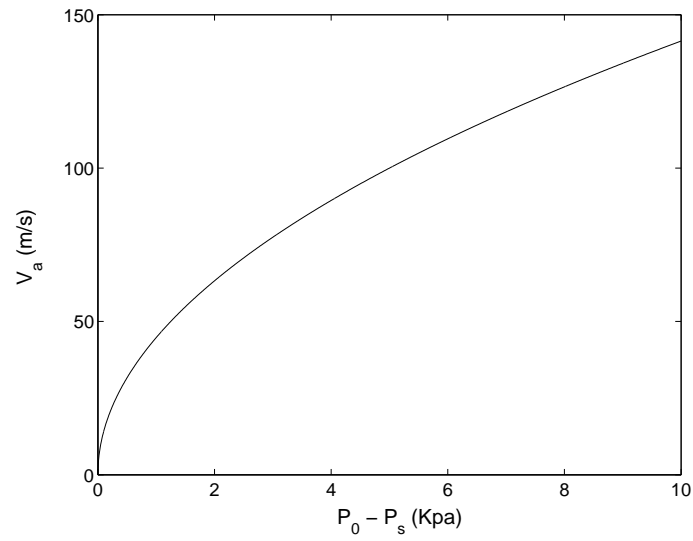


FIG. 3.7 – La vitesse par rapport à l’air  $V_a$  en fonction de la différence de pression  $P_0 - P_s$  selon l’équation de Bernoulli 3.8

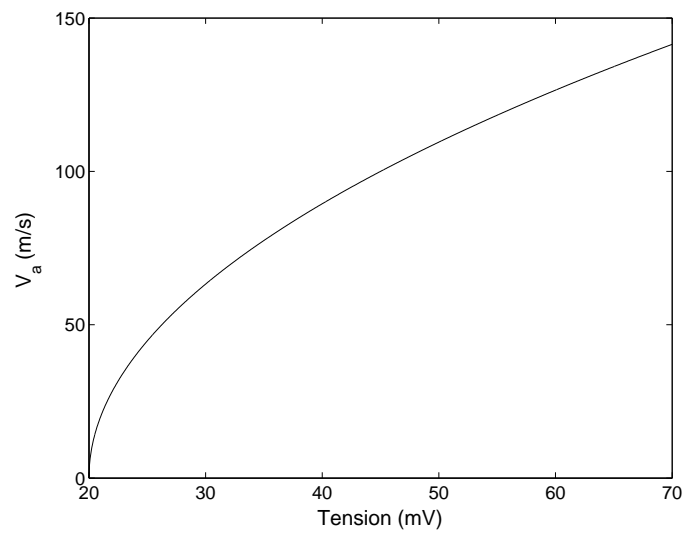


FIG. 3.8 – La vitesse par rapport à l’air en fonction de la tension générée par le capteur selon l’équation 3.9

$$V_{out(MPX10)} = 0.005 \times \rho \frac{V_a^2}{2} + 20 \quad (3.9)$$

Pour connaître la variation de la tension  $V_{out(MPX10)}$  en fonction de la vitesse  $V_a$ , la dérivée de cette tension en fonction de  $V_a$  est calculée (en considérant la masse volumique de l'air  $\rho \approx 1Kg/m^3$ ) :

$$\frac{dV_{out(MPX10)}}{dV_a} = 0.005 \times V_a \quad (3.10)$$

Ainsi, à 1m/s par exemple, la tension de sortie du capteur  $V_{out(MPX10)}$  variera de 0.005mV/(m/s). Ce n'est qu'à  $V_a = 60m/s$  que la variation sera de 0.3mV, soit égale à la résolution de la carte d'acquisition.

Pour augmenter la résolution du signal afin que l'autopilote puisse mieux distinguer les différentes vitesses relatives par rapport à l'air, il est possible de traiter dans Simulink la variation de tension du capteur due à la pente  $m_{(MPX10)}$  du graphique 3.4 (tel qu'illustré à la figure 3.9). À partir de la tension générée entre 0 et 50mV environ, celle-ci est multipliée par un facteur  $F_{(MPX10)}$  et est additionnée à -10V afin de ramener cette tension sur une échelle de -10V à 10V. Cette valeur est alors transmise en sortie à la carte d'acquisition. Un circuit est alors ajouté à la sortie de la carte d'acquisition, permettant d'atténuer le signal de ce même facteur  $F_{(MPX10)}$  et d'ajouter +10V afin de le ramener à l'intérieur de la plage 0 à 50mV. Il reste alors à ajouter la tension à l'origine  $V_{O(MPX10)}$  et la tension de référence  $V_{ref(MPX10)}$ .

Expérimentalement, le circuit schématisé à la figure 3.9 a été monté sous forme de circuit imprimé. Ce circuit est présenté à la figure 3.10. Différents potentiomètres sont présents afin d'ajuster les gains des amplificateurs et la valeur des tensions de référence (référence 10V, tension à l'origine  $F_{(MPX10)} \times V_{O(MPX10)}$  et  $V_{ref(MPX10)}$ ). Aussi, le circuit permet de mettre chacune des tensions de référence (référence 10V, tension à l'origine  $F_{(MPX10)} \times V_{O(MPX10)}$  et  $V_{ref(MPX10)}$ ) à la masse, afin de bien calibrer les gains des amplificateurs. Les amplificateurs U1 et U2 représentent un sommateur suivi d'un inverseur de gain  $\frac{1}{F_{(MPX10)}}$ . Ils permettent d'effectuer la première addition sur le schéma 3.9 suivi de l'atténuation du facteur  $F_{(MPX10)}$ . Les amplificateurs U3 et U4 représentent un sommateur suivi d'un inverseur de gain unitaire, afin d'ajouter au signal compris approximativement entre  $\approx 20$  et 70mV le signal de référence  $V_{ref_{MPX10}} \approx 2.5V$ .

Il a été constaté qu'effectivement ce circuit permet d'atteindre une résolution suffisante. Il semble ajouter un peu de bruit ( bruit  $\approx \pm 0.01\text{mV}$  ) aux données par rapport à lorsque tout le système est monté à l'intérieur de Simulink (comme à la figure 3.6), ce qui a des effets plus marqués à basses vitesses (vitesse  $\approx \pm 3\text{m/s}$ ). Il est important d'utiliser des résistances dont les valeurs varient peu en fonction de la température. Également, il est important de ne pas chauffer le TL431 utilisé pour faire la tension à l'origine. Pour ce faire, la résistance R21 doit être choisie de façon à laisser passer un courant à peine au-dessus du courant requis pour son bon fonctionnement (courant  $\approx 1.25\text{mA}$ ), sans quoi un bruit de dérive s'ajoute au signal de sortie.

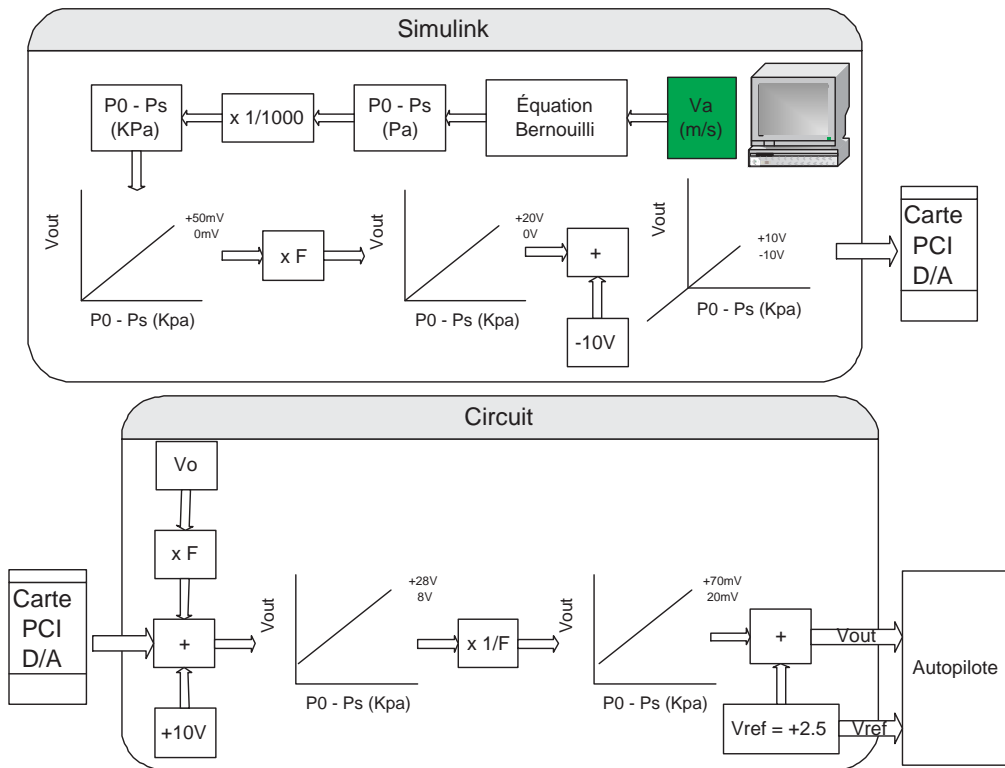


FIG. 3.9 – La reproduction de la tension lorsqu'un circuit imprimé est ajouté à la sortie de la carte d'acquisition afin d'augmenter la résolution du signal

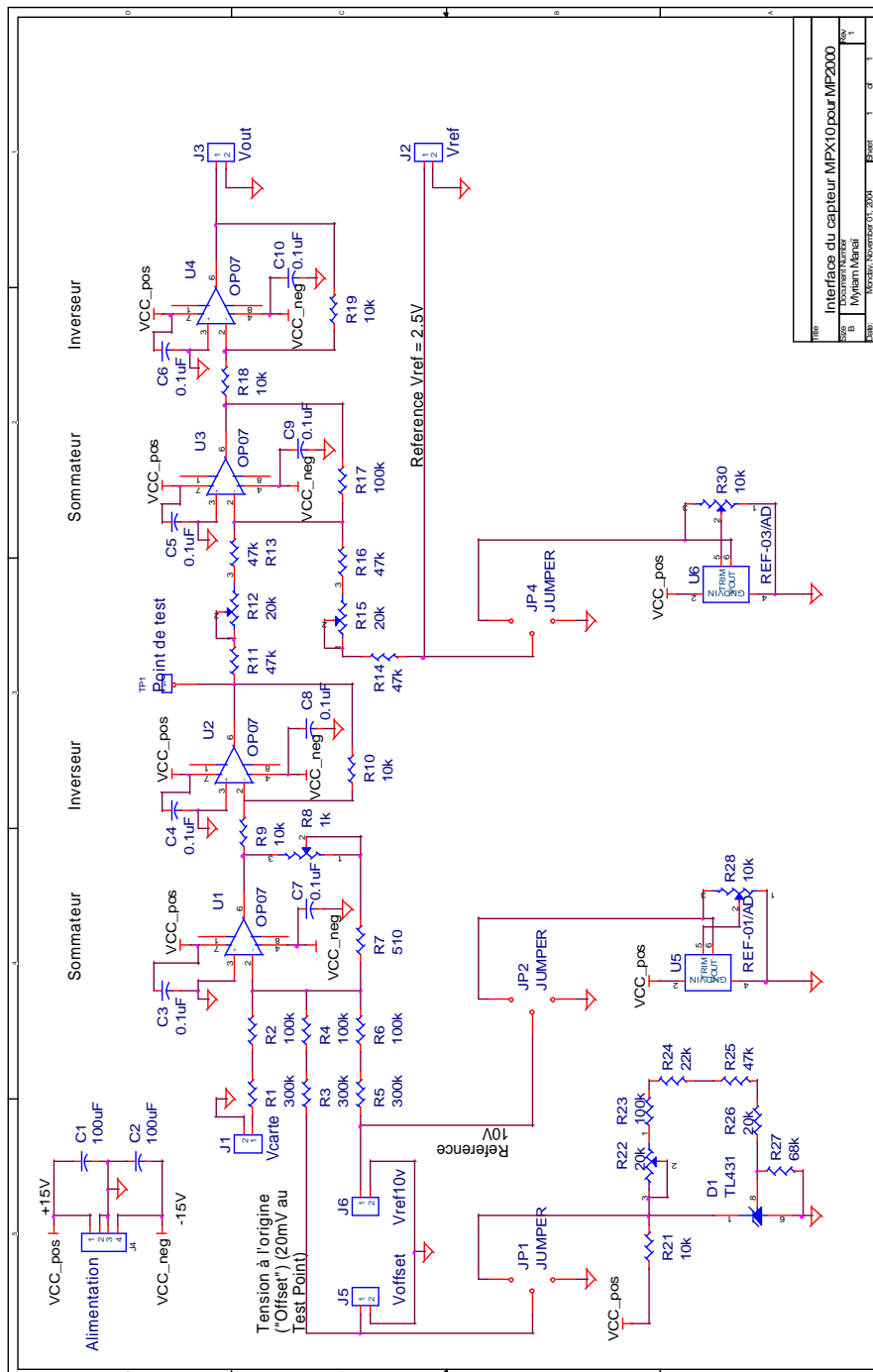


FIG. 3.10 – Circuit de l'interface pour le capteur MPX10

### 3.5 La reproduction du signal du capteur de l'altitude

Le capteur MPX4115A mesure en fait la pression statique absolue  $P_S$ , soit la pression atmosphérique. Il renvoie donc une tension proportionnelle à la pression mesurée tel qu'illustré à la figure 3.11 (selon les spécifications du capteur [4]), où  $P_S$  est symbolisé simplement par  $P$ . La tension  $V_{out(MPX4115A)}$  que le capteur renvoie s'exprime donc par l'équation suivante, soit une droite de pente  $m_{(MPX4115A)}$  avec une tension à l'origine  $V_{o(MPX4115A)}$ .

$$V_{out(MPX4115A)} = m_{(MPX4115A)} \times (P_s - 15) + V_{o(MPX4115A)} \quad (3.11)$$

L'autopilote déduit à partir de cette pression l'altitude à laquelle l'avion se situe. Plusieurs approximations existent pour calculer l'altitude à partir de la pression. Il n'est pas mentionné dans la documentation de l'autopilote quelle approximation est utilisée.

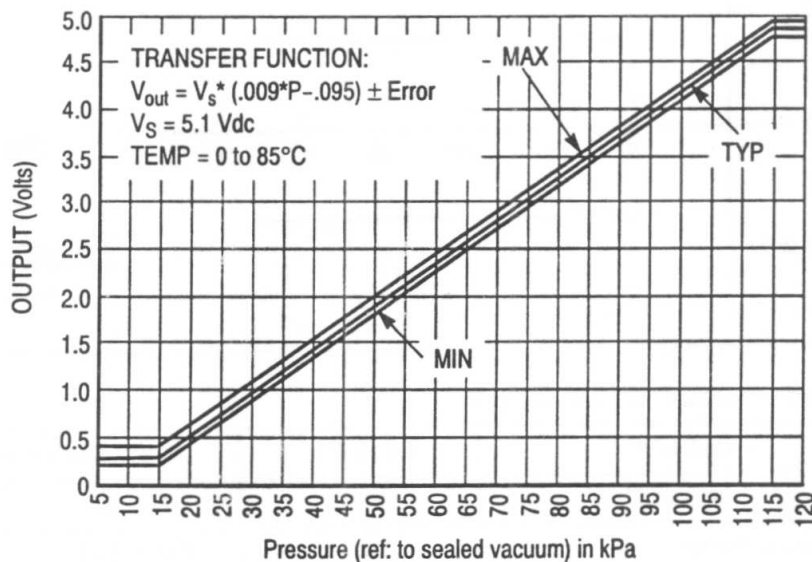


FIG. 3.11 – La sortie en tension du capteur MPX4115A en fonction de la pression absolue mesurée [4]

Les étapes pour reproduire le comportement du capteur sont représentées à la figure 3.5. À noter que dans les figures de cette section, les indices des variables dénotant qu'elles s'appliquent pour le capteur MPX4115A (ex :  $Vout_{(MPX4115A)}$ ) ne sont pas indiqués pour alléger la présentation. Le modèle fournit l'altitude  $Alt$  à laquelle se situe l'avion de même que la pression atmosphérique  $P_{abs}$  à cette altitude selon une table de données. La première étape est d'approximer le plus précisément possible la pression  $P_s$  à une altitude donnée. Ne sachant pas quelle approximation l'autopilote utilise pour déduire l'altitude de la pression mesurée par le capteur, deux approximations différentes sont utilisées pour tenter de retrouver la pression à partir de l'altitude. Celle convenant le mieux à nos besoins sera choisie. La première façon est d'utiliser la table déjà présente dans le modèle de l'avion (provenant de la librairie Aerosym). Cette table permet entre autres d'obtenir la pression atmosphérique  $P_{abs}$  selon l'altitude (ainsi  $P_s = P_{abs}$ ). La seconde façon est d'utiliser l'altitude provenant du modèle et de calculer la pression atmosphérique à l'aide de l'approximation suivante :

$$P_s = P_{s0} \times e^{\frac{-h}{26216}} \quad (3.12)$$

La pression  $P_s$  ainsi obtenue est en Kpa et l'altitude  $h$  est en pieds au-dessus du point de référence où la pression est  $P_{s0}$  (Ici  $P_{s0} \approx 101.325 Kpa$  est utilisé). Puisque l'altitude  $Alt$  du modèle est fourni en m,  $h = Alt/0.3048$ .

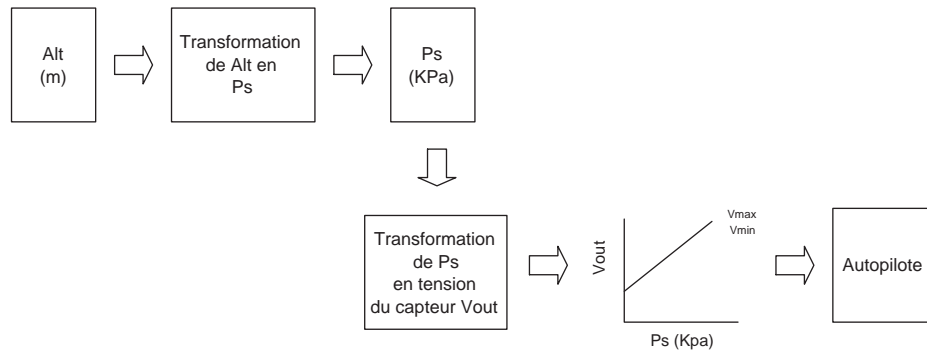


FIG. 3.12 – Les étapes de transformation de l'altitude  $Alt$  du modèle en la tension normalement générée par le capteur MPX4115A

Il est possible d'implanter l'équation 3.11 entièrement à l'intérieur de Simulink, tel que présenté à la figure 3.13. La tension  $Vout_{(MPX4115A)}$  obtenue

est alors dirigée vers la sortie de la carte d'acquisition qui est directement connectée à l'autopilote. Ici, le taux de variation de la tension diminue en fonction de l'augmentation de l'altitude.

En effet, en substituant l'expression de la pression en fonction de l'altitude (équation 3.12) dans l'équation de la droite du capteur décrivant la tension qu'il génère 3.11, l'expression suivante est obtenue.

$$Vout_{(MPX4115A)} = m_{(MPX4115A)} \times (P_{s0} \times e^{\frac{-Alt}{26216 \times 0.3048}} - 15) + Vo_{(MPX4115A)} \quad (3.13)$$

En calculant la dérivée de la tension  $Vout_{(MPX4115A)}$  en fonction de l'altitude  $Alt$  en m, l'équation 3.14 est obtenue.

$$\frac{dVout_{(MPX4115A)}}{dAlt} = \frac{-1}{26216 \times 0.3048} \times m_{(MPX4115A)} \times P_{s0} \times e^{\frac{-Alt}{26216 \times 0.3048}} \quad (3.14)$$

Plus l'altitude augmente, plus la résolution de la tension de sortie doit être précise. Puisque fort probablement l'approximation de la relation entre l'altitude et la pression n'est pas exactement la même que celle utilisée par l'autopilote, les paramètres de la droite du capteur (équation 3.11) sont ajustés expérimentalement afin de permettre d'obtenir de la part de l'autopilote des lectures précises (ils n'ont donc pas exactement la même valeur que celles mentionnées dans les spécifications). Ainsi, pour la méthode utilisant l'approximation de la pression donnée par l'équation 3.12, les paramètres sont  $m_{(MPX4115A)} \approx 41.3\text{mV}$  et  $Vo_{(MPX4115A)} \approx 0.6V$ . Avec de telles valeurs, ce n'est qu'à environ 4452m d'altitude que la résolution de la tension sera égale ou plus faible que 0.3mV, soit la résolution de la carte d'acquisition.

Quant à la table du modèle fournissant  $P_{abs}$ , celle-ci utilise le modèle «US Standard Atmosphere 1976», qui représente la pression atmosphérique en KPa pour une altitude de zéro à 11Km par l'équation suivante :

$$P_{abs} = P_{s0} \times \left(1 - \frac{Alt}{44329}\right)^{5.25586} \quad (3.15)$$

En substituant cette expression de la pression statique dans l'équation de la droite du capteur représentée par l'équation 3.11 puis en calculant la dérivée de  $Vout_{(MPX4115A)}$  en fonction de l'altitude  $Alt$  en m dans l'équation résultante, l'expression suivante de la variation de la tension du capteur en fonction de l'altitude est obtenue.



$$\frac{dV_{out(MPX4115A)}}{dAlt} = \frac{-m_{(MPX4115A)} \times 5.25586 \times P_{s0}}{44329} \times \left(1 - \frac{Alt}{44329}\right)^{4.22586} \quad (3.16)$$

Lorsque la table est utilisée, la valeur de la pente utilisée est  $m_{(MPX4115A)} = 42.3\text{mV}$ . Ainsi, selon l'équation 3.16, le capteur a une résolution égale ou plus faible que  $0.3\text{mV}$  à partir de  $5163.36\text{m}$ .

Toutefois, pour simuler l'avion à des altitudes plus élevées (le capteur fonctionnant jusqu'à au moins  $10\text{Km}$  d'altitude), il est possible d'améliorer la résolution de ce signal. Il s'agit encore une fois d'implanter dans Simulink la variation due à la pente de la droite du capteur  $m_{(MPX4115A)}$ , tel qu'illustré à la figure 3.14, ce qui représente selon les spécifications une sortie en tension entre  $0.0\text{V}$  et  $4.59\text{V}$ . Cette valeur de tension peut alors être multipliée par un facteur  $F_{(MPX4115A)}$  puis additionnée à  $-10\text{V}$ , de façon à ramener le signal entre  $-10\text{V}$  et  $+10\text{V}$ . Cette valeur de tension est alors transmise à la carte d'acquisition. Le signal de la carte d'acquisition est par la suite traité par un circuit qui lui additionne  $10\text{V}$  et qui atténue la tension totale du même facteur  $F_{(MPX4115A)}$  afin de ramener la tension entre  $0.204\text{V}$  et  $4.764\text{V}$  (selon les spécifications du capteurs). La tension est alors transmise à l'autopilote.

Expérimentalement, le circuit schématisé à la figure 3.14 a été monté sous forme de circuit imprimé. Le schéma de ce circuit est présenté à la figure 3.15. Des potentiomètres permettent d'ajuster les gains des amplificateurs et les tensions de référence (référence  $10\text{V}$  et tension à l'origine  $V_{O(MPX4115A)}$ ). Également, il est possible de mettre chacune des tensions de référence à la masse afin de faire l'ajustement des gains. Le circuit est composé de deux amplificateurs, U1 et U2, permettant de créer un sommateur et un inverseur de gain  $\frac{1}{F_{(MPX4115A)}}$ . Ces amplificateurs additionnent donc les tensions de références à la tension provenant de la carte d'acquisition et atténuent la somme d'un facteur  $F_{(MPX4115A)}$ , tel qu'illustré à la figure 3.14. Pour créer la tension à l'origine  $V_{O(MPX4115A)}$ , deux circuits ont été implantés, chacun d'entre eux utilisant un TL431. Seul l'un des deux circuits est utilisé à la fois, selon que la pression  $P_s$  provienne de la table de données ou de l'équation 3.12.

La pente  $m_{(MPX4115A)}$  et le facteur  $F_{(MPX4115A)}$  sont légèrement différents selon que tout le traitement est implanté dans Simulink (figure 3.13) ou avec l'ajout du circuit (figure 3.14). Également, ils diffèrent un peu selon que la pression  $P_s$  provienne de la table de données de Aerosym ou de l'équation

3.12. Pour les plages d'altitude auxquelles la résolution de la carte d'acquisition est suffisante, une fois les paramètres  $m_{(MPX4115A)}$  et  $V_{o(MPX4115A)}$  ajustés correctement, il est possible d'atteindre une précision  $\approx \pm 1m$  lorsque la table de données de la librairie Aerosym est utilisé pour obtenir la pression, et ce pour une altitude jusqu'à  $\approx 2000m$ . En utilisant plutôt l'équation 3.12 pour obtenir la pression  $P_s$ , il est possible d'atteindre une précision de  $\pm 5m$  jusqu'à une altitude d'environ 6000m lorsque le circuit est utilisé. Ainsi, pour le moment, le circuit est requis pour ce dernier cas, puisqu'il est possible d'atteindre une altitude à partir de laquelle la résolution de la carte d'acquisition n'est plus suffisante. Pour le cas où la table de données d'Aerosym est plutôt utilisée, le circuit n'est pas requis puisque pour l'instant, une altitude d'au plus 2000m peut être atteinte en gardant une précision raisonnable. Le circuit ne semble pas influencer la précision de l'altitude lue par la carte peu importe l'approximation de la pression en fonction de l'altitude utilisée.

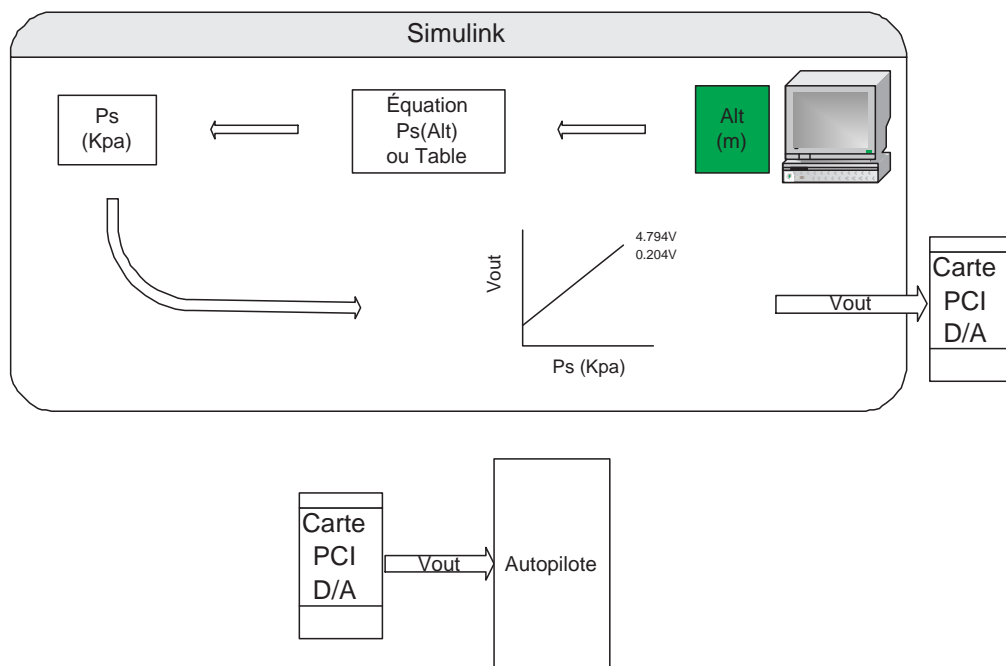


FIG. 3.13 – La reproduction de la tension lorsqu'elle est entièrement faite à l'intérieur de Simulink

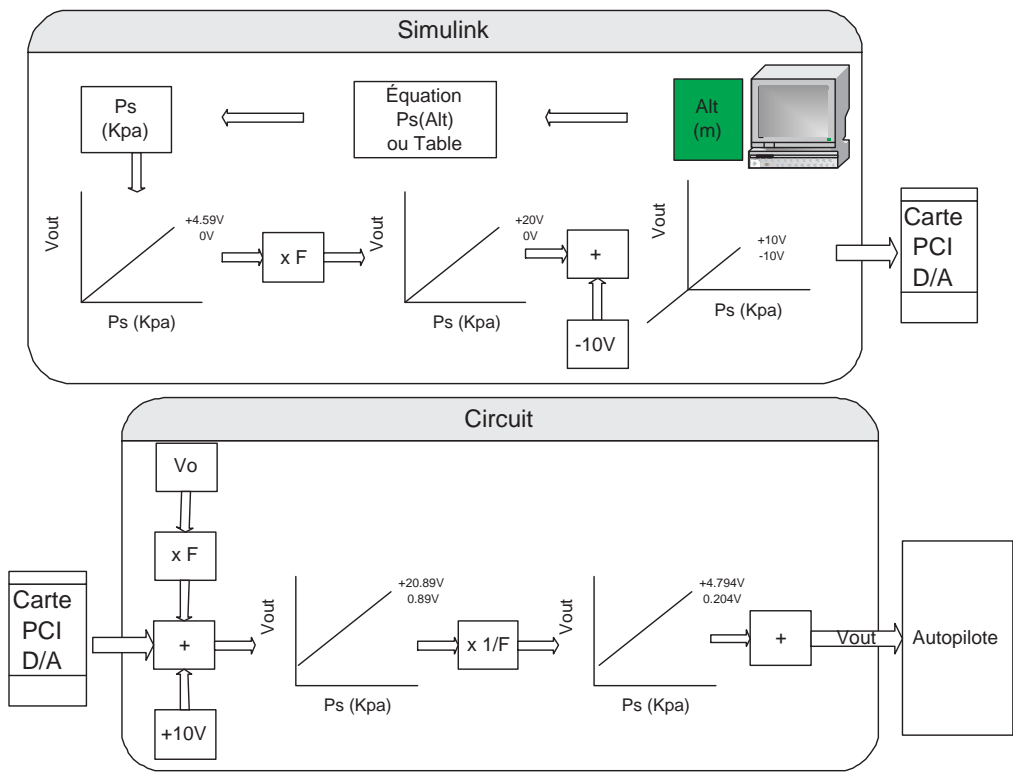


FIG. 3.14 – La reproduction de la tension lorsqu'un circuit est ajouté à la sortie de la carte d'acquisition afin d'augmenter la résolution du signal

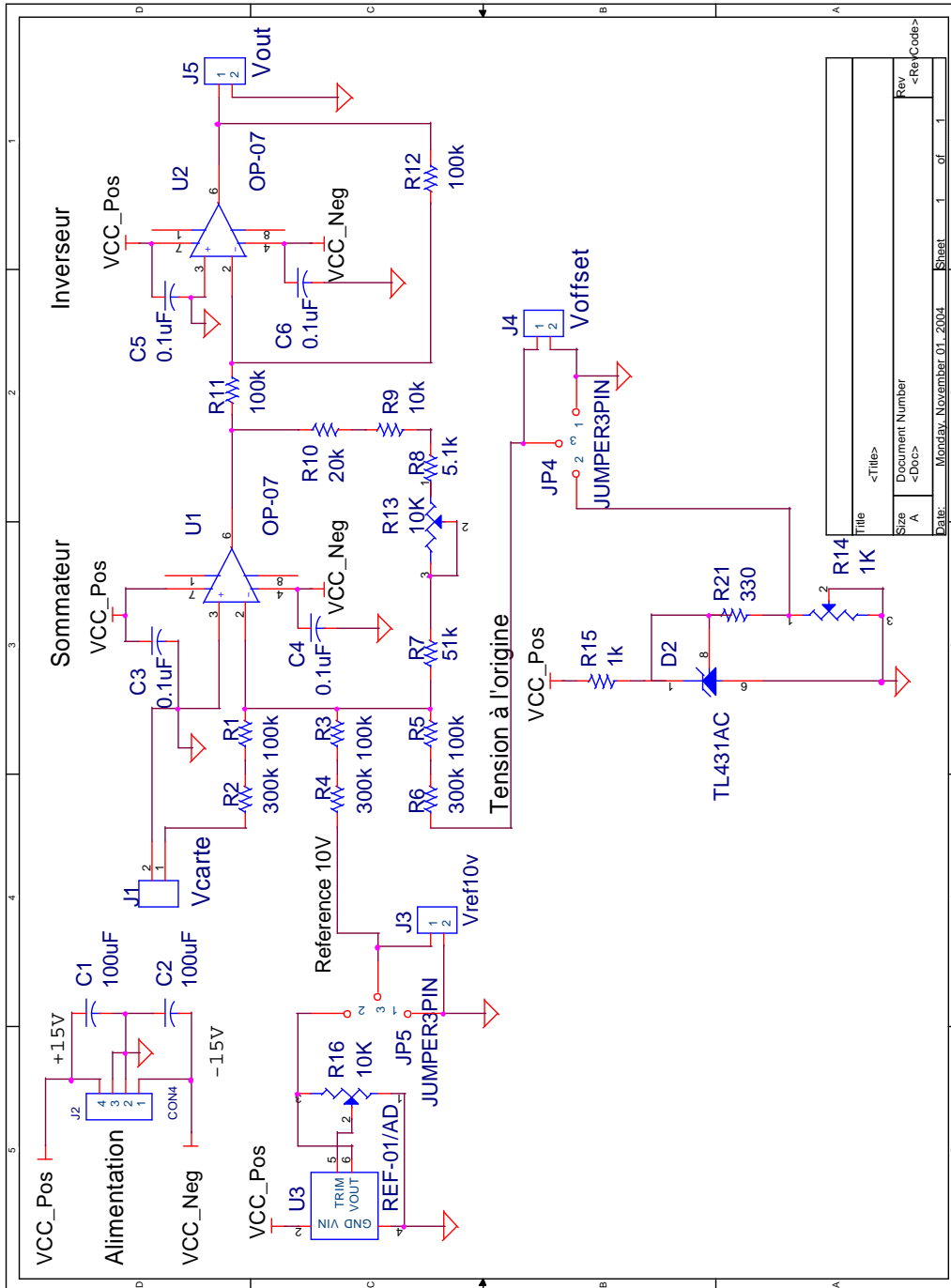


FIG. 3.15 – Circuit de l'interface pour le capteur MPX4115A

### 3.6 La reproduction des paquets d'information du système de positionnement global (GPS)

Le système de positionnement global (GPS) permet d'obtenir la position géodésique et la vitesse de l'avion dans le référentiel géographique NED («Local Geodetic, North-East-Down»). Le système peut communiquer les renseignements via trois protocoles. Le protocole utilisé par l'autopilote est le TSIP («Trimble Standard Interface Protocol»). Le système GPS transmet à l'autopilote de façon automatique les paquets contenant la position, la vitesse, et le temps de même que certains paquets de maintenance. À toutes les secondes, les paquets dont les numéros d'identification («id») sont 6D, 82, 4A, 56 sont transmis. À toutes les 5 secondes, les paquets 41, 46, 4B sont transmis à la suite des paquets à 1Hz. Le tableau 3.1 résume les paquets transmis, l'information qu'ils contiennent de même que la fréquence à laquelle chacun d'eux est transmis. Les données sont transmises à l'autopilote via une communication RS232 (avec 1 seul bit d'arrêt et 1 bit de parité impaire). L'envoi d'un paquet se fait selon la séquence suivante [7] :

<DLE> <id> <octets de données> <DLE> <ETX>

où <DLE> a la valeur 10h, <id> est le numéro d'identification du paquet (voir tableau 3.1) et <ETX> a la valeur 03h. Lorsqu'un octet dans les octets de données transmis a la valeur 10h, un octet supplémentaire de valeur 10h également est ajouté juste après celui-ci, de façon à ne pas confondre un octet de données 10h avec l'octet <DLE> indiquant la fin du paquet. Ainsi, une séquence d'envoi d'un paquet commence par la valeur 10h, suivie d'un octet identifiant de quel paquet il s'agit puis des données transmises, et se termine par un nombre impair d'octets de valeur 10h puis d'un octet 03h. Également, lorsqu'une donnée dans les octets de données est d'un type contenant plusieurs octets (une donnée de type «single» par exemple), les octets pour cette donnée sont transmis de l'octet le plus significatif à l'octet le moins significatif. Aussi, les types de données pour les nombres à virgule flottante («single» et «double») suivent le standard ANSI/IEEE Std. 754. Les différents types de données utilisés dans les paquets sont décrits dans le tableau 3.2.

Le simulateur doit transmettre ces paquets à l'autopilote de la même façon que le module GPS réel le ferait. Toutefois, l'ordinateur transmet tous les paquets à une fréquence de 1Hz (la librairie xPC Target de Simulink ne

permettant pas de varier le nombre d'octets transmis d'un pas de simulation à l'autre). Ainsi, à toute les secondes, le simulateur transmet dans l'ordre les paquets 4Ah, 56h, 41h, 46h, 4B, 6D, 82.

Le modèle fournit la position géodésique et la vitesse NED. Le simulateur crée alors le paquet 4Ah à partir de la position géodésique obtenue ( $Lat, Lon, Alt_{HAE}$ ) de même que le paquet 56h à partir de la vitesse NED ( $V_{nord}, V_{est}, V_{bas}$ ). Le paquet 41h transmet le temps à l'autopilote, en indiquant le numéro de la semaine courante et le temps écoulé à l'intérieur de la semaine courante en secondes (voir le manuel de l'utilisateur du GPS [7] pour plus de détails). Il n'est pas essentiel que l'heure transmise soit juste. Les secondes à l'intérieur de la semaine courante sont transmises à l'aide d'une horloge fournie par la librairie générale de Simulink. Les autres données du paquet 41h sont fixées constantes.

Les paquets de maintenance contiennent des données indiquant le bon fonctionnement de l'appareil. Pour la simulation, les valeurs de ces données sont fixées constantes et sont pour la plupart semblables à celles observées lors de prises de données faites avec le GPS en fonctionnement.

Les données transmises dans chacun des paquets sont présentées dans les tableaux 3.3, 3.4, 3.5, 3.6, 3.7, 3.8 et 3.9 respectivement. Le format de ces paquets (de même que l'information qu'ils contiennent) sont définis dans le manuel de l'utilisateur du GPS [7].

Pour assurer une transmission identique à celle faite par le module GPS réel, deux fonctions ont été conçues. Il s'agit de S-Functions dans Matlab/Simulink, soit des fonctions codées en langage C et compilées par le compilateur Matlab. Ces fonctions peuvent par la suite être appelées à l'intérieur de Simulink afin d'être intégrées au modèle. À la différence de simple fonctions Matlab appelées dans Simulink à l'aide du bloc «Matlab function», les S-Function peuvent être compilées par xPC Target (lorsqu'elles sont codées dans certains langages, dont le langage C).

Une première fonction permet d'ajouter l'octet 10h suite à un octet 10h déjà présent dans la séquence des octets de données transmises. Une seconde fonction permet d'inverser les octets d'une donnée de type single, le bloc «Byte Reversal» fourni par xPC Target ne fonctionnant pas pour ce type de données.

Lors des tests réalisés, tous les paquets ont été créés et différentes valeurs de position et de vitesse NED ont été transmises à l'autopilote et lues correctement par celui-ci. Les cas où les données de position et de vitesse NED contiennent des octets de valeur 10h ont été testés afin de vérifier si la fonction

TAB. 3.1 – Les paquets TSIP transmis par le GPS : Fréquence et contenu

Paquets	Informations	Fréquence de transmission
4Ah	Position géodésique	1 Hz
56h	Vitesse linéaire NED	1 Hz
41h	Temps	0.2 Hz
46h	État du récepteur	0.2 Hz
4Bh	Status	0.2 Hz
6Dh	Satellites en vue	1 Hz
82h	Mode DGPS	1 Hz

TAB. 3.2 – Les paquets TSIP : Les types de données transmises

Type de données	Description
UINT8	1 octet, non signé
INT16	2 octets, signé
Single	4 octets, nombre à virgule flottante

chargée de détecter de tels octets et d'insérer un octet 10h supplémentaire à la suite de ces octets fonctionne bien. Tout a très bien fonctionné.

TAB. 3.3 – Les paquets TSIP : Le paquet 4Ah

Octets	Informations	Types de données	Valeurs constantes	Unités
0-3	Latitude	Single		rad ; + vers nord, - vers sud
4-7	Longitude	Single		rad ; + vers est, - vers ouest
8-11	Altitude	Single		m (HAE)
12-15	Biais de l'horloge	Single	0	m
19-16	Temps positionnement	Single	0	s

TAB. 3.4 – Les paquets TSIP : Le paquet 56h

Octets	Informations	Type de données	Valeurs constantes	Unités
0-3	Vitesse est	Single		m/s ; + vers est, - vers ouest
4-7	Vitesse nord	Single		m/s ; + vers nord, - vers sud
8-11	Vitesse bas	Single		m/s ; + vers haut, - vers bas
12-15	Taux biais horloge	Single	0	m/s
19-16	Temps positionnement	Single	0	s

TAB. 3.5 – Les paquets TSIP : Le paquet 41h

Octets	Informations	Type de données	Valeurs constantes	Unités
0-3	Temps à l'intérieur de la semaine	Single		s
4-7	No de la semaine	INT16	1249	semaines
8-11	Délai par rapport au temps UTC	Single	13	s



TAB. 3.6 – Les paquets TSIP : Le paquet 46h

Octets	Informations	Types de données	Valeurs constantes
0	Status	UINT8	0
1	État de la batterie et de l'antenne	UINT8	0

TAB. 3.7 – Les paquets TSIP : Le paquet 4Bh

Octets	Informations	Types de données	Valeurs constantes
0	Machine id	Single	90
1	Status 1	Single	0
2	Status 2	Single	1

TAB. 3.8 – Les paquets TSIP : Le paquet 6Dh

Octets	Informations	Types de données	Valeurs constantes
0	3D/Mode Automatique/nsvs = 4	UINT8	68 (44h)
1-4	PDOP	Single	4.87672
5-8	HDOP	Single	3.225967
9-12	VDOP	Single	3.65738
13-16	TDOP	Single	3.22325
17	nsvs1	Single	20
18	nsvs2	Single	4
19	nsvs3	Single	7
20	nsvs4	Single	28

TAB. 3.9 – Les paquets TSIP : Le paquet 82h

Octets	Informations	Types de données	Valeurs constantes
0	Auto DGPS inactif	UINT8	2

# Chapitre 4

## Conclusion

En résumé, l'interface pour le système «hardware in the loop» est en place. Des signaux sont transmis du simulateur à l'autopilote et il est possible de lire les données lues par l'autopilote. Ainsi, en transmettant une valeur connue à l'autopilote pour un capteur donné (par exemple une valeur de vitesse par rapport à l'air) et en interrogeant l'autopilote sur ce qu'il lit, il est possible de vérifier si les signaux de ce capteur sont bien reproduits. La reproduction des signaux des capteurs est très avancée. Les signaux des capteurs de vitesse par rapport à l'air, de l'altitude et des accélérations sont reproduits avec une précision acceptable. Les signaux des capteurs de vitesses angulaires sont également reproduits mais il est plus difficile d'évaluer leur exactitude, étant donné que les vitesses angulaires lues par l'autopilote proviennent à la fois des signaux des capteurs de vitesses angulaires et des accéléromètres. Aussi, les paquets GPS sont bien reproduits. Les signaux PWM provenant de l'autopilote et destinés aux servos sont lus et interprétés pour en déduire les angles correspondants aux actionneurs (dans le cas des signaux contrôlant les positions du gouvernail de direction, du gouvernail de profondeur et des ailerons) et la fraction correspondante (dans le cas du signal contrôlant la poussée du moteur). Finalement, la prochaine étape quant à la conception du système «hardware in the loop» est de connecter le modèle à cette interface.

# Bibliographie

- [1] Analog Devices, Inc. *Low Cost  $\pm 2g/10g$  Dual Axis iMEMS Accelerometers with Digital Outputs*, 1999.
- [2] MicroPilot, inc. *MP2000 Installation and Operation*, January 2003.
- [3] Motorola, inc. *Semiconductor Technical Data, 10kPa Uncompensated Silicon Pressure Sensors,MPX10 MPXV10GC Series*, 2001.
- [4] Motorola, inc. *Semiconductor Technical Data, Integrated Silicon Pressure Sensor for Manifold absolute Pressure , Altimeter or Barometer Applications On-Chip Signal Conditioned Temperature Compensated and Calibrated, MPX4115A MPXA4115A Series*, 2001.
- [5] Bruce R. Munson, Donald F. Young, and Theodore H. Okiishi. *Fundamentals of Fluid Mechanics*. John Wiley & Sons, Inc., third edition, 1998.
- [6] NEC/TOKIN. *Ceramic Gyro, NEC TOKIN Sensors*.
- [7] Trimble Navigation Limited. *Lassen TM SK II GPS, System Designer Reference Manual*, Novembre 1999.