

Filière Systèmes industriels

Orientation Power & Control

Diplôme 2010

Michaël Briguet

*Optimisation des trajectoires
d'un robot Delta*

Professeur Jean-Daniel Marcuard

Expert Christian Vouillamoz

Sion, le 12 juillet 2010



Optimisation des trajectoires d'un Robot Delta



Diplômant

Michaël Briguet



Objectif du projet

Ce projet a pour but d'analyser et de développer une méthode permettant d'optimiser les consignes d'accélération et de vitesse en fonction de la trajectoire du robot tout en limitant les échauffements moteurs.

Méthodes | Expériences | Résultats

Le travail s'est déroulé en plusieurs étapes. Dans un premier temps, une certaine période a été nécessaire à l'apprentissage du robot et de son environnement de travail. Une analyse théorique a ensuite été effectuée afin de déterminer les comportements à étudier. A l'aide d'un programme d'acquisition, les informations relatives aux mouvements du robot ont été récupérées puis traitées avec le logiciel Matlab. Le choix du paramètre à améliorer s'est porté sur l'accélération du plan de transition du robot. Pour se faire, le script du logiciel Matlab a été optimisé pour effectuer différentes analyses et définir les accélérations critiques selon les trajectoires. Finalement, ces valeurs ont permis de réaliser une cartographie en 3D du plan de transition.

Un travail conséquent doit encore être fourni pour obtenir une optimisation concrète du robot. Néanmoins, une progression significative a été réalisée dans cette direction. De plus, le rapport contient des analyses ainsi que des développements théoriques offrant des perspectives intéressantes vers la concrétisation de l'objectif premier du projet.

Travail de diplôme | édition 2010 |



Filière

Systemes industriels

Domaine d'application

Power & Control

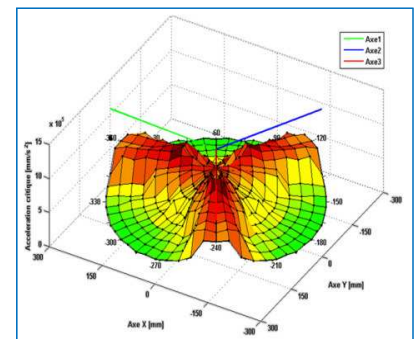
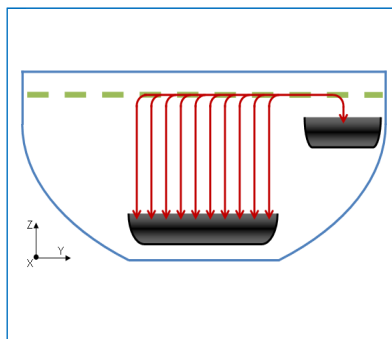
Professeur responsable

Marcuard Jean-Daniel

jdaniel.marcuard@hevs.ch

Partenaire

Bosch Packaging Technology SA



Bleu : Vue de face du volume de travail

Noir : Convoyeurs

Red dashed line : Trajectoires radiales

3D de la cartographie des accélérations
 sur le plan de transition
 des trajectoires radiales et

TABLE DES MATIERES

1	Introduction	1
1.1	Choix du sujet	1
1.2	Historique du Robot Delta	1
2	Cahier des charges.....	2
3	Robot Delta	2
3.1	Généralités	2
3.2	Trajectoire initiale du robot	4
4	Déroulement du projet.....	5
5	Travail à temps partiel.....	6
5.1	Initiation	6
5.2	Première visite à Romanel.....	6
5.3	Apprentissage du robot.....	6
5.3.1	Environnement de travail.....	6
5.3.2	Architecture de la programmation	7
5.3.3	Régulation du robot	9
5.3.4	Programme d'acquisition	10
5.3.5	Logiciel D95CONF	12
5.4	Exactitude des valeurs.....	13
5.5	Problèmes rencontrés	13
5.6	Idées initiales d'optimisation.....	14
6	Premières améliorations	16
7	Bilan intermédiaire.....	16
8	Travail à plein temps	16
8.1	Conséquences du bilan intermédiaire.....	16
8.2	Seconde visite à Romanel.....	17
8.3	Principe pratique de la trajectoire du robot	17
8.3.1	Choix du paramètre à améliorer	18
8.4	Pseudo couple	19
8.5	Symétrie du robot	19
8.6	Réflexions sur l'algorithme d'assignation.....	20
8.6.1	Zones circulaires.....	21
8.6.2	Zones triangulaires.....	21
8.7	Etude de l'accélération limite d'un mouvement.....	22
8.7.1	Secteur de mesures.....	22
8.7.2	Variance	24
8.7.3	Cycle d'essais.....	24
8.7.4	Détermination de l'accélération critique.....	25
8.8	Implémentation du cycle de variance	25
8.8.1	Programme du cycle	26
8.8.2	Echantillonnage et deuxième dérivée sous Matlab	26
8.9	Décalage des données.....	29
8.9.1	Principe des microsamples.....	29
8.9.2	Correction du décalage sous Matlab	31
8.10	Séries d'essais.....	32
8.10.1	Implémentation sous le programme « EX_10.PAS »	33
8.11	Améliorations du programme Matlab.....	33

8.11.1	Navigation	34
8.11.2	Nouvelle acquisition.....	34
8.11.3	Types d'analyse et variance plateau	35
8.11.4	Traitement de l'acquisition	36
8.11.5	Sauvegarde.....	38
8.11.6	Fichier TXT.....	38
8.11.7	Retraitement d'acquisition.....	39
8.11.8	Première série d'essais : Série 1.....	39
8.12	Conception de la cartographie 3D sous Matlab	40
8.12.1	Script de représentation polaire en 3D.....	40
8.12.2	Script de recherche d'une variable dans un fichier	41
8.12.3	Mise au point d'un secteur de cartographie.....	41
8.12.4	Création secteur par secteur.....	42
8.13	Première cartographie : Série 1.....	42
8.14	Augmentation des points de mesures.....	44
8.15	Restructuration des séries d'essais	44
8.16	Révision du programme d'acquisition « EX_10.PAS ».....	45
8.17	Programme final de représentation cartographique	46
8.18	Nouvelles cartographies.....	47
8.18.1	Série 10.....	48
8.18.2	Série 11.....	50
8.18.3	Série 30.....	51
8.18.4	Série 31.....	53
8.18.5	Série 40.....	54
8.18.6	Série 50.....	56
8.19	Synthèse des cartographies.....	57
9	Echauffements moteurs	58
10	Discussion des résultats.....	59
10.1	Problèmes rencontrés	59
10.2	Améliorations et perspectives.....	60
11	Conclusions.....	61
12	Bibliographies.....	62
13	Remerciements	63
14	Table des figures.....	64
15	Table des Tableaux	66
16	Table des Codes.....	66
17	Table des équations.....	66
18	Annexes.....	66

1 INTRODUCTION

C'est en 1998 que Bosch Packaging Technology SA a intégré la société Demaurex SA à Romanel, située près de Lausanne. Cette dernière a développé depuis la fin des années 80 des solutions d'applications industrielles associées aux robots à géométrie parallèle Delta. Particulièrement utilisé dans l'agro-alimentaire pour la mise en boîte, ce type de robot est caractérisé par une rapidité élevée et une répétabilité très intéressante grâce à sa légèreté. De plus, sa conception formée de 3 parallélogrammes permet de maintenir une charge quelconque dans une même orientation.

Le robot évolue à une vitesse et à une accélération initialement fixées à partir des mouvements davantage contraignants pour les moteurs avec un facteur de correction limitant les échauffements moteurs. Les trajectoires les plus défavorables se situent aux abords de son volume de travail.

Bosch Packaging Technology SA estime que cette méthode a pour conséquence de limiter la rapidité du robot lorsque celui-ci effectue des trajectoires non-contraignantes. La productivité pourrait augmenter en variant les consignes d'accélération et de vitesse en fonction des itinéraires effectués par le robot, en limitant les échauffements moteurs. L'HES-SO de Sion dispose, depuis quelques années, d'un Robot Delta de type « Presto » provenant de Demaurex SA. Un programme permet de recueillir les vitesses ainsi que les positions théoriques et réelles du robot pour ensuite les interpréter.

C'est aussi avec une certaine approche que ce rapport a été élaboré, donnant un accès simplifié et plus rapide à cette machine initialement conçu dans les années 80 à l'EPFL par Monsieur Reymond Clavel.

1.1 *Choix du sujet*

La robotique m'ayant toujours intrigué, ce sujet de Travail de Diplôme c'est présenté comme une opportunité de découvrir ce domaine au-delà de ce qui a été abordé à l'HES-SO.

1.2 *Historique du Robot Delta*

- ◆ 1983 : Création de Demaurex SA par Monsieur Marc-Oliver Demaurex
- ◆ 1985 : Développement du robot à l'EPFL par Monsieur Reymond Clavel
- ◆ 1987 : Dépôt du brevet
- ◆ 1988 : Développement du robot par Demaurex SA avec l'EPFL
- ◆ 1992 : Première vente pour une boulangerie : 6 robots / 85 cycles/min
- ◆ 1996 : Acquisition chez Nestlé : 7 robots / 120 cycles /min
- ◆ 1999 : Association au groupe SIG-Pack
- ◆ 2004 : Rachat de la division SIG-Pack par Bosch
- ◆ 2007 : Tombée du brevet du Robot Delta dans le domaine public
- ◆ 2009 : Vente du 1300^{ème} robot et Demaurex SA devient "Bosch Packaging Technology SA"

2 CAHIER DES CHARGES

- ◆ Se former à la programmation du Turbo Pascal
- ◆ Se familiariser à l'environnement de travail du Robot Delta Presto de l'HES
- ◆ Effectuer une analyse théorique du robot et sélectionner les variables à étudier
- ◆ Améliorer l'interface de mouvement de robot sous DOS et faciliter son accès
- ◆ Améliorer le logiciel d'analyse de données sous MATLAB, simplifier son utilisation et augmenter les fonctionnalités
- ◆ Mettre au point un cycle de test permettant de recueillir les données significatives
- ◆ Analyser les données issues des expériences.
- ◆ Développer une stratégie attribuant une accélération et une vitesse optimale à chaque mouvement du robot
- ◆ Implémenter la stratégie sous forme d'un algorithme dans la commande du robot.
- ◆ Valider par des essais l'optimisation d'accélération et de vitesse
- ◆ Développer une solution permettant de limiter l'échauffement des moteurs
- ◆ Programmer l'algorithme de limitation des pertes calorifiques
- ◆ Valider par des essais, la stratégie d'atténuation des échauffements.

3 ROBOT DELTA

3.1 Généralités

Aujourd'hui, Bosch Packaging Technology SA est le leader mondial reconnu dans le domaine de l'automation flexible. Cette réputation a été acquise en grande partie grâce à la robotique de type parallèle dont le robot Delta y est issu.

La machine est composée de peu de pièces et son assemblage est relativement simple. Cela lui confère une grande rapidité et répétabilité. 3 moteurs AC brushless actionnent la crémaillère de chaque axe. L'extrémité d'une crémaillère agit sur une paire parallèle d'avant bras rattachés aussi au châssis, au dessous des moteurs, par un bras en forme d'Y grec pivotant sur un axe décalé de quelques centimètres du centre du robot. Sur le bas du robot, les 3 avant bras se rattachent à une nacelle qui a la caractéristique de toujours maintenir la même orientation. La nacelle peut embarquer une simple buse ou un système permettant de saisir plusieurs objets en même temps. C'est une technique de préhension par le vide qui permet de saisir les objets. Par son architecture, le robot peut se déplacer dans les 3 axes X, Y et Z. Un 4^{ème} axe est issu de la rotation du système de saisie de la nacelle par l'intermédiaire d'un tube télescopique au centre du robot. Son volume de travail correspond à une demi-sphère aplatie sur le fond et sur les cotés comme le démontre la figure suivante :



Figure 1 : Robot Delta de l'HES-SO

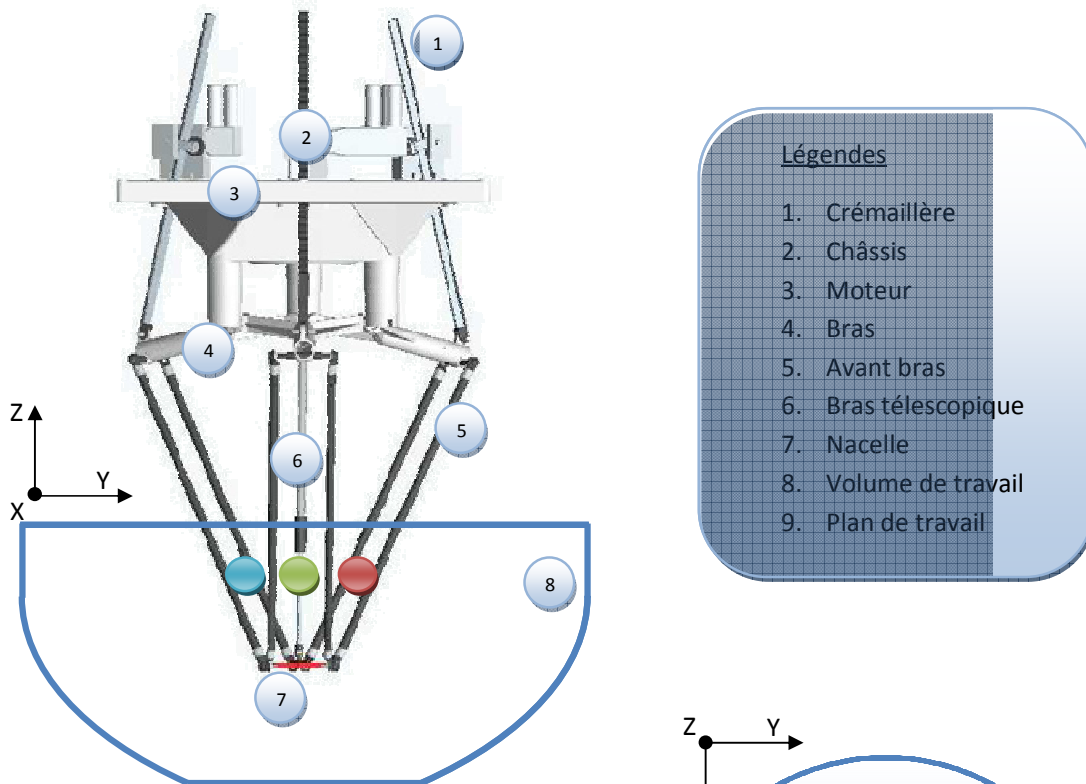
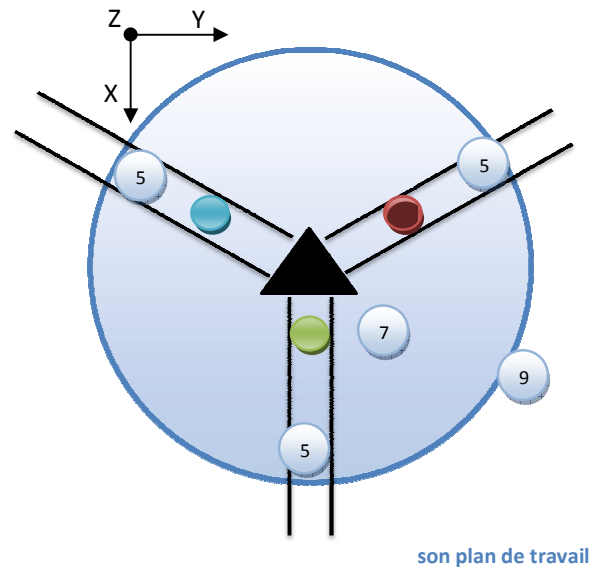


Figure 2 : Robot Delta vu de face avec son volume de travail

Ci-contre, il s'agit d'une vue schématique du robot depuis le dessus à la limite de son volume de travail. Le plan de travail représente les limite où le robot peut évoluer en X et en Y à une hauteur bien précise.

Pour une lecture facilitée du rapport, les axes du robot sont associés à une couleur. **Le vert pour l'axe 1**, **le bleu pour l'axe 2** et **le rouge pour l'axe 3**.



Remarque :

Pour une meilleure lisibilité, les axes du plan cartésien sont affichés sur le coté des figures. En réalité ils sont définis comme ci-contre :

Le point d'origine se situe au milieu des 3 moteurs à la hauteur de l'axe des bras. Il est à noter que l'axe X suit toujours le moteur 1 du robot, donc avec la figure ci-contre, l'axe X sort de la page en direction du lecteur. L'axe Y est toujours opposé au moteur 2.

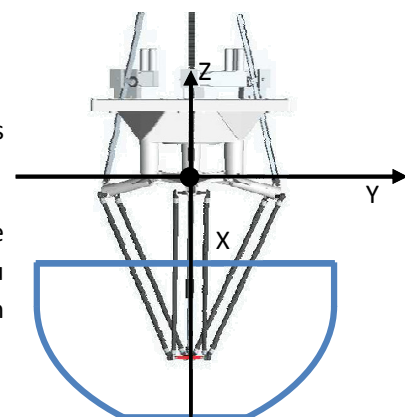


Figure 4 : Axes cartésiens du robot

3.2 Trajectoire initiale du robot

Les objets arrivent sur une bande transporteuse pour être déposés dans un blister par l'intermédiaire du robot. Ce dernier effectue alors une trajectoire dite "Pick and Place" comme le démontre la figure suivante :

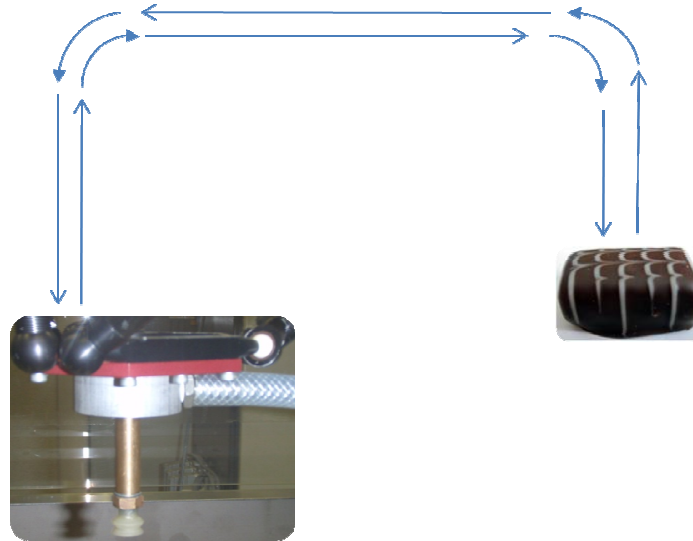


Figure 5 : Cycle "Pick and Place"

La nacelle du robot se déplace d'abord jusqu'au dessus de l'objet à saisir à une hauteur définie. Il effectue ensuite un mouvement vertical jusqu'à l'objet pour le saisir avec la préhension par le vide. Ensuite, le robot remonte à sa hauteur initiale, se déplace jusqu'au dessus du blister, puis redescend verticalement et relâche l'objet.

La transition entre le mouvement vertical et horizontal ne se fait pas instantanément. C'est là que rentre en jeu la notion des « Overlaps ». Ce procédé permet au mouvement horizontal de se débiter avant que le mouvement vertical ne soit terminé. Cela permet de gagner du temps et d'arrondir les trajectoires. La figure ci-dessous représente le fonctionnement des overlaps. A noter que ce sont des distances qui sont paramétrables et que c'est dans cette portion de trajectoire que les efforts sur les moteurs sont les plus rigoureux.

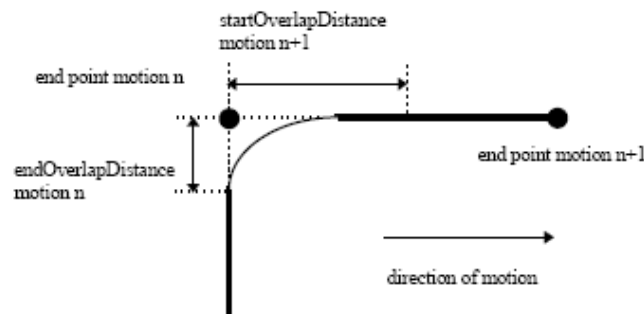


Figure 6 : Principe des "Overlaps"

Si l'on représente la vitesse en fonction du temps en effectuant mouvement purement horizontal ou vertical, on obtient la courbe suivante :

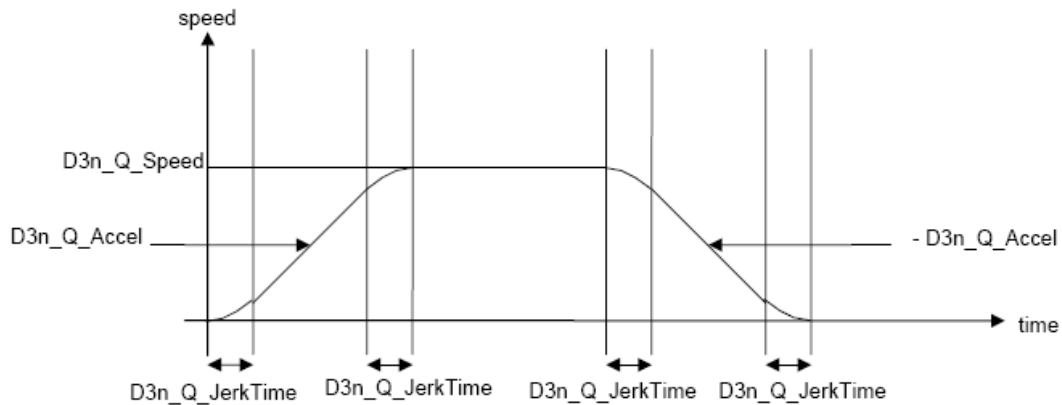


Figure 7 : Profil théorique de vitesse d'un mouvement

- ◆ "D3n_Q_Speed" correspond au retour de la vitesse auquel le robot doit se déplacer.
- ◆ "D3n_Q_Accel" correspond au retour de l'accélération auquel le robot doit se déplacer.
- ◆ "D3n_Q_JerkTime" correspond au retour du temps de jerk auquel le robot doit se déplacer.
Le jerk est le temps pour atteindre l'accélération désirée.

Idéalement, le robot devrait suivre parfaitement cette courbe, mais ce n'est pas le cas en pratique. En effet, les efforts sur les axes du robot varient en fonction de la trajectoire à effectuer. Cette interprétation prouve que l'optimisation des trajectoires du robot mérite une analyse plus approfondie, de plus les résultats pourraient apporter de grandes améliorations.

4 DÉROULEMENT DU PROJET

Le Travail de Diplôme se déroule en plusieurs étapes :

1. Travail sur le projet à temps partiel : 22 février 2010
2. Rapport intermédiaire : 7 mai 2010
3. Présentation intermédiaire : 25 mai 2010
4. Travail sur le projet à plein temps : 18 mai 2010
5. Remise du rapport : 12 Juillet 2010 à 12h00
6. Portes ouvertes : 27 aout 2010
7. Défense du Travail de Diplôme : 2 septembre 2010

5 TRAVAIL À TEMPS PARTIEL

Durant cette première partie, le temps de travail accordé au projet était d'une journée par semaine du 22 février 2010 au 17 mai 2010

5.1 Initiation

Le début du projet a commencé par une présentation de la machine par Monsieur Jean-Daniel Marcuard, le 17 février 2010. Ce point de départ a permis de prendre les premiers repères sur le robot et d'avoir un aperçu de l'architecture software et hardware à la veille de la première visite chez Bosch Packaging Technology à Romanel

5.2 Première visite à Romanel

Le 18 février 2010, accompagné de Monsieur Jean-Daniel Marcuard, un premier contact a été réalisé au sein de l'entreprise Bosch Packaging Technology à Romanel. Il y a initialement eu un entretien avec Monsieur Alain Teklits, chargé du suivi du Travail de Diplôme, et Monsieur Michel Huser, chargé de répondre aux questions techniques, sous l'œil de Monsieur Marc-Olivier Demaurex, fondateur de Demaurex SA et directeur actuel de l'entreprise. Le principe de fonctionnement du robot, ainsi que les objectifs du Travail de Diplôme, ont été approfondis. L'apprentissage du robot est l'objectif premier avant d'envisager quoi que ce soit. Par la suite, les propositions d'améliorations seront validées progressivement. Une visite très intéressante des locaux de l'entreprise a ensuite conclu le déplacement sur Romanel.

5.3 Apprentissage du robot

Connaître la machine ainsi que son fonctionnement est une étape primordiale avant d'envisager tout développement. Ce chapitre résume la manière au robot d'évoluer dans son cadre de travail.

5.3.1 Environnement de travail

Le robot Delta Presto de l'HES-SO est commandé depuis un ordinateur « PC Îlot » évoluant sous un environnement MS DOS. Sa programmation est élaborée en TurboPascal avec le logiciel Borland 7. Les cartes d'axes intégrées au PC Îlot commandent des amplis avec des consignes en tension de ± 10 [Vdc]. Ces dernières se chargent de piloter les moteurs. Des codeurs renvoient les informations de vitesse et de positions à l'ordinateur ÎLOT. Un deuxième ordinateur (PC SUPERVISEUR), évoluant sous Windows XP, permet la gestion du contrôle-commande de la machine, l'acquisition de données des 3 axes du robot et le traitement de celles-ci.

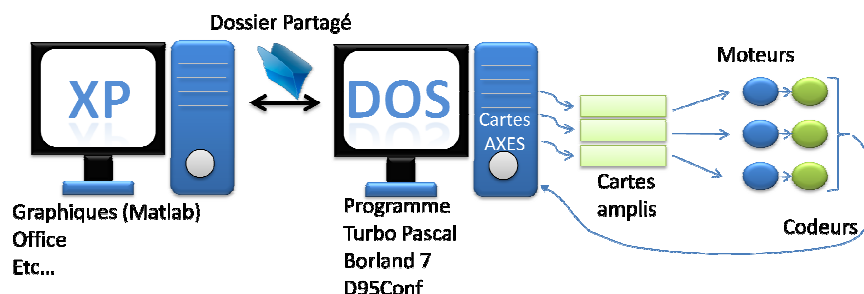
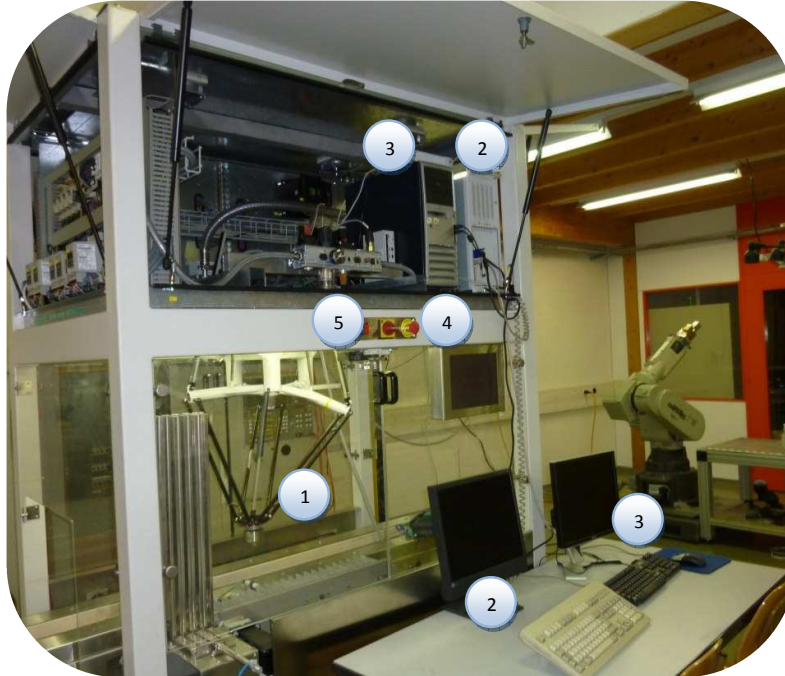


Figure 8 : Résumé de l'environnement de travail



Légendes

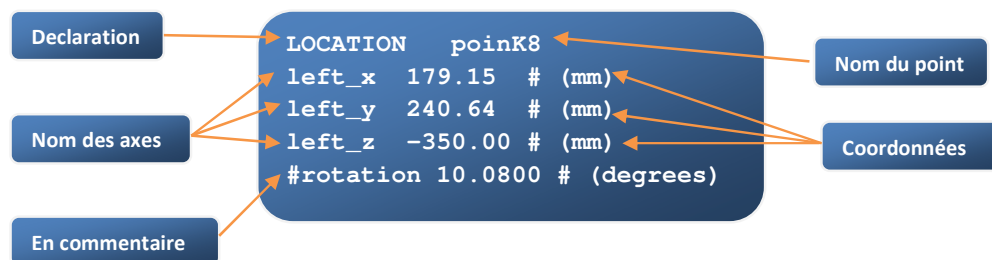
- 1 : Robot
- 2 : PC Îlot
- 3 : PC Superviseur
- 4 : Arrêt d'urgence
- 5 : Interrupteur principal

Figure 9 : Environnement de Travail

5.3.2 Architecture de la programmation

Afin qu'un programme puisse mettre le robot en mouvement, il doit être associé à une série de fichiers.

- ◆ Le programme de base.
Celui-ci intègre le fonctionnement que l'on veut appliquer au robot. C'est ce fichier qui va être principalement développé au cours du Travail de Diplôme. Il s'agit ici à la base du programme "EX_10.PAS"
- ◆ SOFT_REC.CFG
Ce fichier contient les données "software" associées au robot comme les coordonnées des points de mouvements. La figure suivante démontre la manière dont les points sont définis



- ◆ HARD_REC.CFG
Ce fichier contient les données "Hardware" du robot comme les paramètres de régulations, du volume de travail, d'inertie, de géométrie du robot etc. L'image suivante représente un exemple de paramètres modifiables dans le fichier.

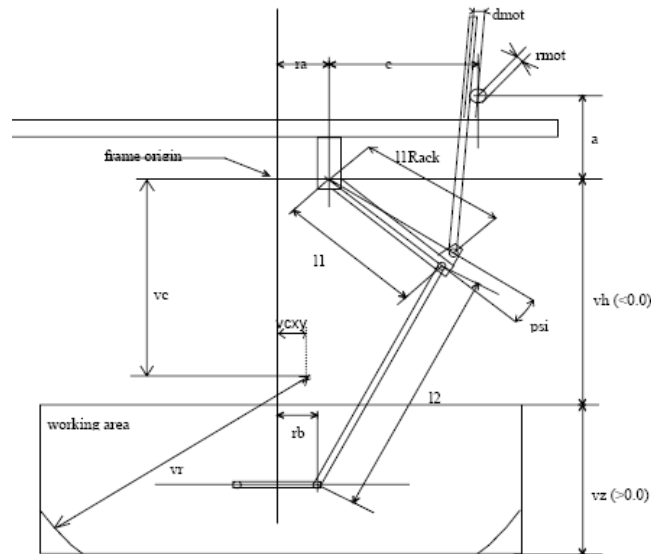


Figure 10 : Exemple de paramètres Hardware

◆ Bibliothèque .TPU

Cette série de fichiers intègre toutes les procédures nécessaires au fonctionnement de la machine.

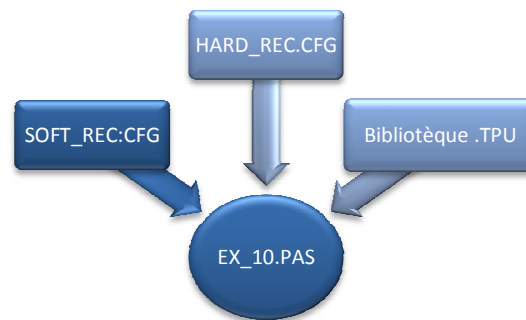
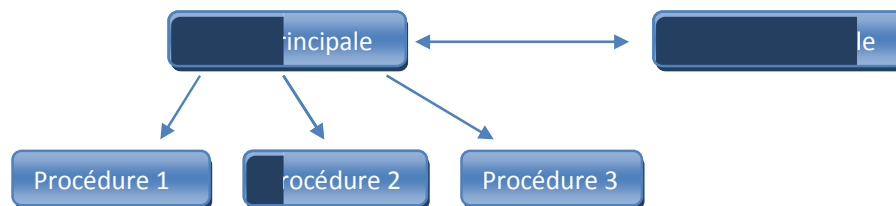


Figure 11 Architecture du programme

L'environnement DOS sous lequel évolue le PC Îlot ne permet pas de gérer le multitâche. Mais Demarex SA a développé un noyau multitâche permettant de gérer la tâche principale, où les diverses procédures du programme sont lancées, ainsi qu'une tâche de contrôle en parallèle.



5.3.3 Régulation du robot

Un asservissement avec une boucle PIDF se situe en amont des moteurs. Le schéma suivant illustre le procédé.

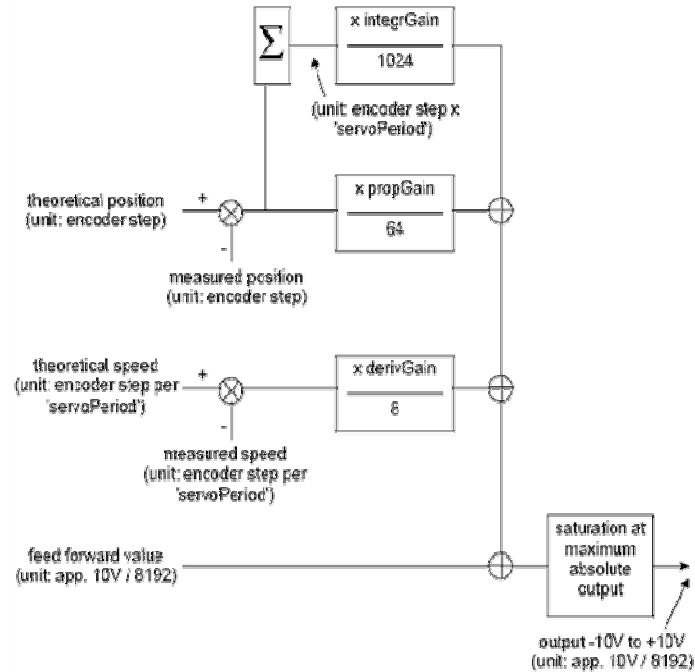


Figure 13 : Principe de la boucle de régulation PIDF

Du ± 10 [V] sort de la boucle PID, représentant une consigne de vitesse, afin de commander les cartes d'amplis qui se chargent ensuite de l'asservissement de vitesse et de courant. Les gains de régulation peuvent être modifiés pour tous les axes dans le fichier HARD_REC.CFG. A noter que la commande à priori, « feed-forward » est calculée à partir de beaucoup de paramètres dont :

- ◆ Le facteur de conversion des unités de l'encodeur.
« *positionUnitsPerEncoderStep* »
- ◆ Les frottements visqueux.
« *viscousDragsSeenByMotor* »
- ◆ Les frottements secs.
« *dryDragsSeenByMotor* »
- ◆ L'inertie totale du robot vue par les moteurs.
« *inertiaSeenByMotor* »
- ◆ De la masse embarquée.
« *loadMass* »
- ◆ Etc.

5.3.4 Programme d'acquisition

Lors du précédent Travail de Diplôme, réalisé par Pierre Fournier en 2009, un programme d'acquisition a été développé. Il est intégré au fichier « EX_10.PAS ». Celui-ci permet au robot d'exécuter des mouvements simples et d'en tirer les informations théoriques et réelles de la position, de la vitesse et du couple.

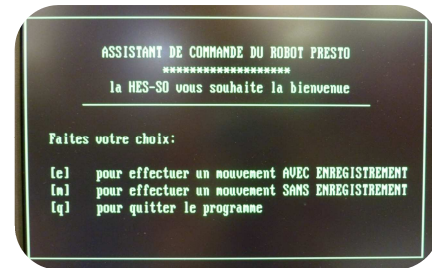


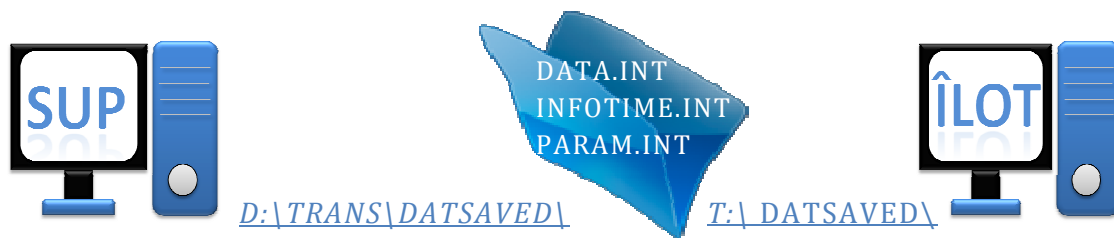
Figure 14 : Accueil EX_10.PAS

Ces données sont écrites et stockées dans le dossier partagé entre le PC Îlot et le PC Superviseur à l'intérieur des fichiers suivants :

- ◆ DATA.INT : Données brutes des mouvements.
- ◆ INFOTIME.INT : Informations relatives au temps.
- ◆ PARAM.INT : Paramètres associées aux mouvements.

Vu depuis le PC Îlot, l'adresse du dossier partagé est « T:\ » .

Vu depuis le PC Superviseur, l'adresse du dossier partagé est « D:\TRANS \DATSAVED\ » .



Ces 3 fichiers sont interprétés par un script évoluant sous le logiciel Matlab. Il s'agit du script « read_and_plot.m », lui aussi développé par Pierre Fournier en 2009. Celui-ci se charge de représenter graphiquement les 3 axes avec comme informations :

- ◆ La position réelle et théorique.
- ◆ L'erreur de position.
- ◆ La vitesse réelle et théorique.
- ◆ L'erreur de vitesse.
- ◆ Le couple réel et théorique.
- ◆ L'erreur de couple.

Les courbes sont automatiquement sauvegardées dans un dossier « D:\Plot Robot Delta\ » mais ne permettent pas un traitement ultérieur car se sont des simples représentations graphiques au format « .fig ». De plus, les fichiers de données, « DATA.INT », « INFOTIME.INT » et « INFOPARAM.INT » sont automatiquement écrasés lors d'un nouvel enregistrement

Remarque 1 : EX 10.PAS

Le logiciel de commande du robot ne permet pas une grande liberté. Les seules possibilités de mouvement sont préprogrammées et l'on peut choisir de réaliser un mouvement en enregistrant, ou pas, les informations de la trajectoire pour les traiter par la suite. Il y avait tout de même la possibilité de modifier les consignes d'accélération et de vitesses du robot

Remarque 2 : « Read_and_plot.m »

Les graphiques ne sont pas très lisibles et un retard d'une dizaine de millisecondes apparaît entre les courbes théoriques et réelles. Ce qui rend la représentation des erreurs faussé. Il est aussi important de savoir que le couple affiché n'en est au fait pas un. En effet, il représente une valeur de consigne d'à-priori. Cette problématique est traitée au chapitre 8.4. Les unités font aussi défauts, que ce soit sur l'axe X ou Y. De plus, les consignes de vitesses et d'accélération sont exprimées en [pu/s] et [pu/s²], ce qui n'est pas très représentatif. Idem pour les informations de la trajectoire, un simple numéro donne une correspondance à celle-ci, alors que les coordonnées complètes seraient bien plus intéressantes. Les images suivantes représentent un graphique Matlab initial et le décalage entre les courbes réelles et théoriques.

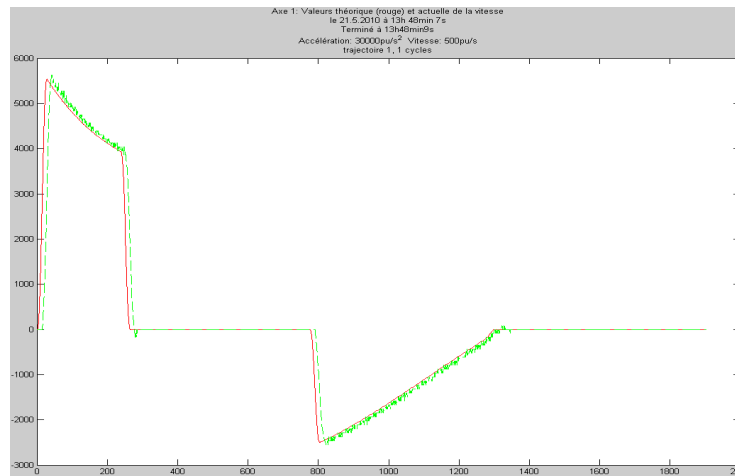


Figure 16 : Graphique Matlab initial

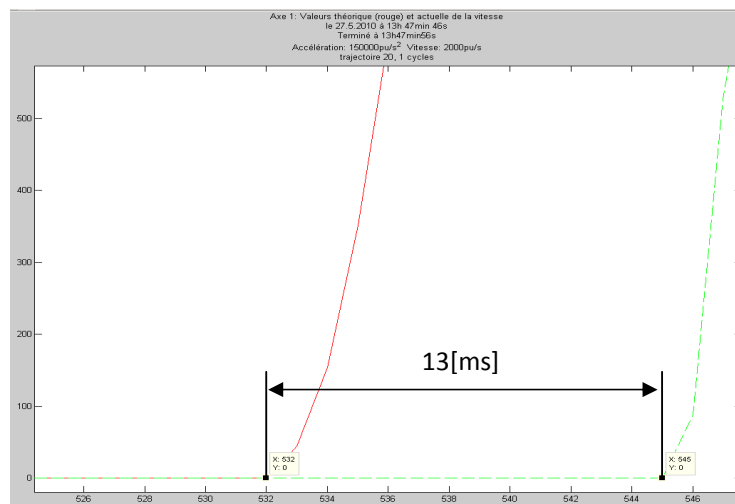


Figure 17 : Décalage de la courbe théorique, en rouge, et réelle, en vert

5.3.5 Logiciel D95CONF

Afin de mettre le robot en mouvement et de ressentir ses limites, le soft « D95CONF » a été utilisé. Il se trouve sur le PC îlot à l'adresse « C:\ARIA\CONFIG95\ »

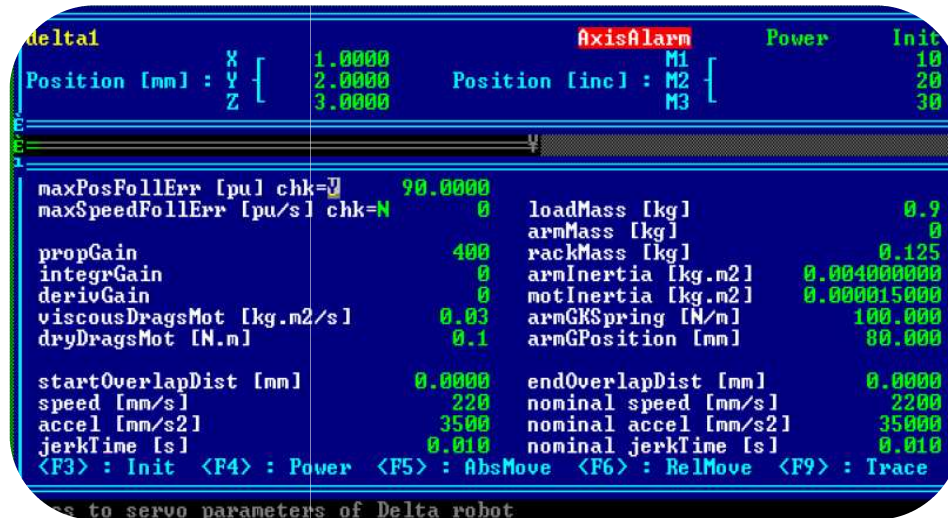


Figure 18 : D95Conf

La position du robot, en haut de l'image, est représentée dans le référentiel cartésien (X, Y, Z) ainsi que le référentiel d'axe du robot (M1, M2, M3). Les 2 modèles géométriques, direct et inverse, permettent de passer d'un plan à un autre.

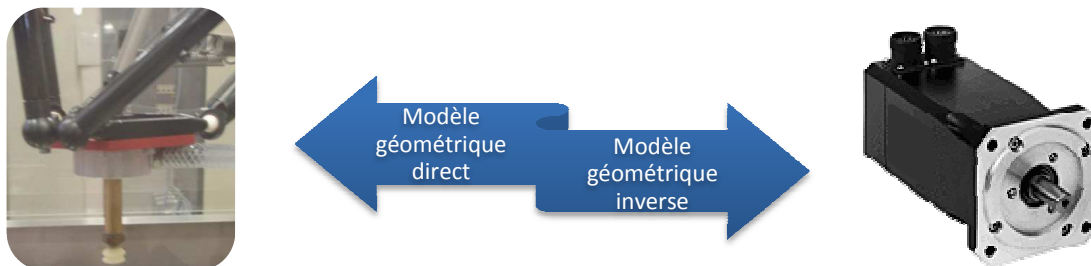


Figure 19 : Principe des modèles géométriques

Une majorité de paramètres peuvent y être modifiés. Mais ceux qui nous intéressent sont essentiellement la vitesse « speed [mm/s] et l'accélération « accel [mm/s²] » sur la colonne de gauche du « D95Conf ». Les mouvements peuvent être effectués en absolu, la position du robot dans le plan cartésien, ou en relatif, la position du robot par rapport à l'angle des moteurs. Avant un mouvement, il y a possibilité d'activer le traceur qui affiche diverses courbes issues du déplacement.

Remarque : D95Conf

Pour valider un paramètre ou lancer un mouvement, il faut appuyer sur « CTRL+ENTER ». Le mode d'emploi officiel du « D95Con » est disponible sur le CD en annexe dans le dossier « Documentation Demarex »

5.4 Exactitude des valeurs

Il est important que les valeurs affichées sur les graphiques Matlab correspondent à la réalité. Pour se faire, un mouvement a été effectué sur le logiciel « D95Conf » ainsi que sur le programme d'acquisition « EX_10.PAS ». Les courbes issues des deux mouvements similaires ont été comparées.

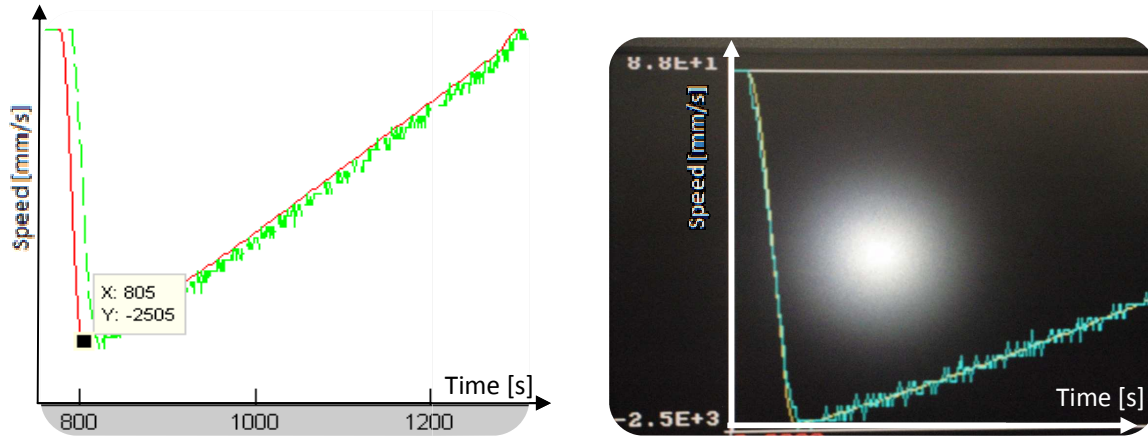


Figure 20 : Comparaison des valeurs

On peut aisément noter que les deux pics de vitesse se situent à 2'500 [mm/s], ce qui veut dire que les mesures issues du logiciel Matlab correspondent bien à la réalité.

5.5 Problèmes rencontrés

Une grande quantité de temps a été consacré à la résolution de problèmes liés au fonctionnement du robot. Le premier était la communication entre le PC Îlot et le PC Superviseur, car le dossier partagé ne se trouvait plus accessible après un échange de fichier. Cela provoquait de gros ralentissements sur les deux ordinateurs et rendait le travail impossible. Le problème se situait au niveau de l'antivirus qui bloquait les ports de communications, dont le port « 427 ». Le service informatique de l'école était tout d'abord très réticent à ma demande d'ouverture des ports et assurait que ce n'était pas l'antivirus la cause du problème malgré les fichiers LOG de l'antivirus attestant l'activité du logiciel de protection. Après avoir insisté, diverses restrictions ont été levées sur le PC Superviseur et le problème de stabilité de communication n'est plus réapparu.

Il est aussi arrivé occasionnellement que le PC Îlot ne démarre pas, mais aucune cause n'a été trouvée. Idem pour le robot, il arrive que ce dernier ne s'enclenche pas. Après quelques recherches, c'était le relais de sécurité qui ne tirait pas. Après quelques déclenchements et ré-enclenchement de la machine, le tout fonctionnait à nouveau normalement. La cause possible était les relais contrôlant la fermeture des portes de sécurité, mais ceux-ci sont déclenchés en permanence. Ce phénomène se manifestant que très rarement, il n'a pas été jugé nécessaire de chercher plus loin car un temps considérable avait déjà été utilisé pour remédier aux autres problèmes informatiques.

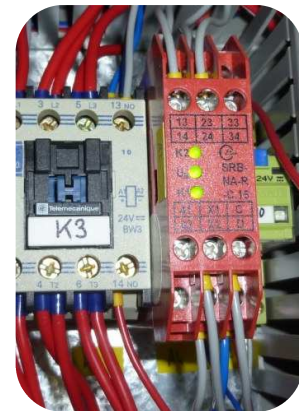


Figure 21 : Relais de sécurité du robot (en rouge)

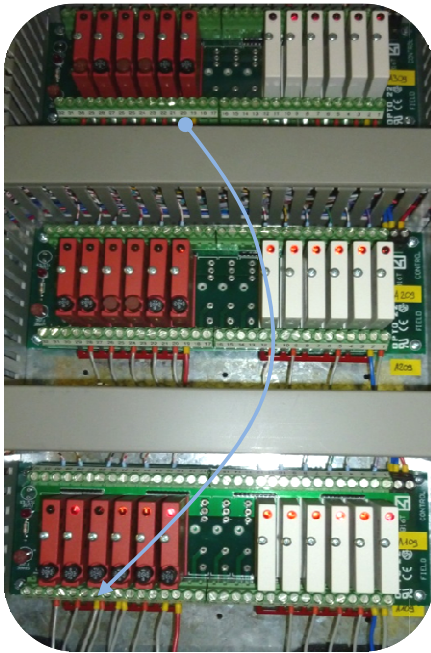
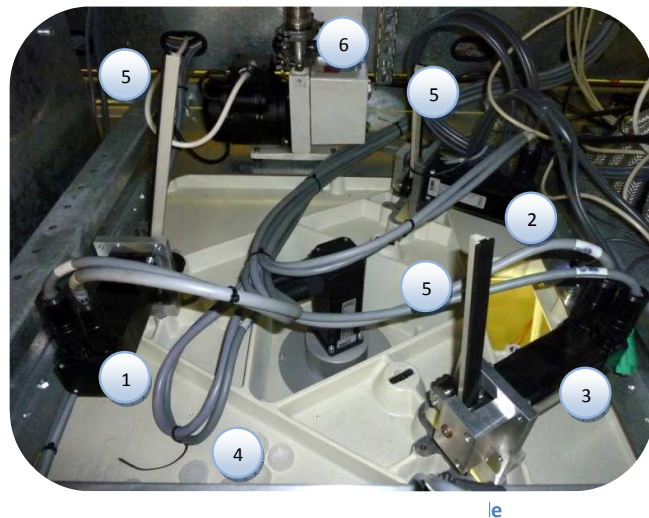


Figure 22 : Câblage de la pompe à vide

Un autre problème était la mise en route de la pompe à vide afin de provoquer un vacuum sur la buse de la nacelle. Celle ci ne s'activait pas malgré son l'activation de sa variable associée depuis le logiciel « D95conf ». La sortie correspondant de la carte d'entrées-sorties ne fonctionnant à priori plus, une solution provisoire a été trouvée en modifiant le câblage. La pompe à vide se met désormais en route avec la sortie de la lumière du bouton d'alarme qui n'est pas vraiment pris en compte actuellement. Le câble du relais de la pompe à vide était sur la borne 20 de la carte A309 avec un câble gris et un embout jaune, il est désormais raccordé sur la sortie 26 de la carte A109.

Légendes

1. Moteur axe 1
2. Moteur axe 2
3. Moteur axe 3
4. Moteur rotation axe 4
5. Crémaillères
6. Pompe à vide



5.6 Idées initiales d'optimisation

La première idée a été de réaliser une cartographie du volume de travail afin de représenter graphiquement les zones favorables et défavorables au robot. La cartographie serait tout d'abord établie sur un plan de travail à une hauteur précise. Ce procédé permettrait ensuite de délimiter des zones d'accélération et de vitesse, à associer au robot comme le démontre la figure ci-contre avec des nombres non-exhaustifs.

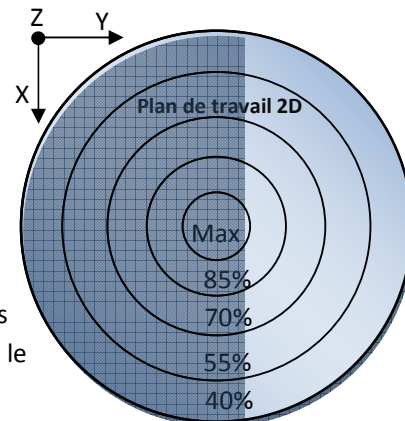
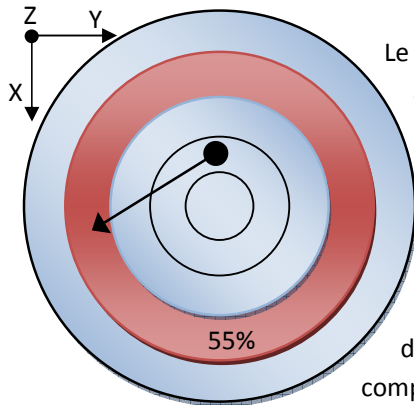


Figure 24 : Idée de carte de valeurs

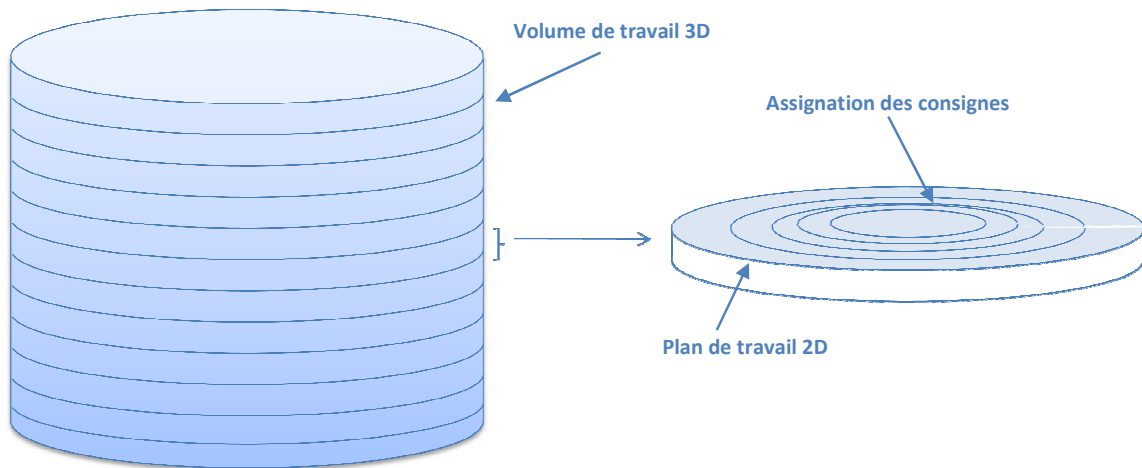


Le principe de sélection d'une consigne d'accélération et de vitesse serait ensuite basé sur le point de trajectoire le plus éloigné du centre du plan de travail, donc on prendrait encore le cas le plus défavorable. La figure à gauche démontre cette logique, avec le point le plus éloigné de la trajectoire représenté par la pointe de la flèche

La cartographie réalisée, l'algorithme de modification d'accélération et de vitesse serait implémenté et une série de test comparant les performances, avec et sans l'algorithme, serait réalisée.

Figure 25 : Idée d'assignation

Si cet algorithme s'avérait performant, le concept serait ensuite adapté sur l'ensemble de volume de travail du robot. La forme donnée au volume de travail est paramétrable dans le fichier « HARD_REC.CFG ». Pour simplifier, il prendrait l'allure d'un cylindre.



Les échauffements moteurs devraient être ensuite considérés. Il serait possible de reconstituer les pertes calorifiques avec les équations du moteur à courant continu.

- ◆ M = couple [Nm]
- ◆ C = constante du moteur [V s/rad]
- ◆ φ = flux d'excitation [V s]
- ◆ i = courant d'induit [A]
- ◆ U = tension d'alimentation [V]
- ◆ R = résistance des enroulements [Ω]
- ◆ L = inductance des enroulements [H]
- ◆ ω = vitesse angulaire du moteur [rad]
- ◆ P_{mec} = Puissance mécanique [W]
- ◆ P_{el} = Puissance électrique [W]
- ◆ P_{th} = Puissance thermique dissipée [W]
- ◆ E_{th} = Energie thermique dissipée [J]

$$U = R \cdot i + C \cdot \varphi \cdot \omega + \frac{L \cdot di}{dt}$$

$$M = C \cdot \varphi \cdot i$$

$$P_{el} = U \cdot i$$

$$P_{mec} = M \cdot \omega = C \cdot \varphi \cdot i \cdot \omega$$

$$P_{th} = P_{el} - P_{mec} = U \cdot i - M \cdot \omega$$

$$E_{th} = \int P_{th} \cdot dt = R \int i^2 \cdot dt$$

Formule 1 : Equations du moteurs DC

En négligeant la différentielle de courant avec l'inductance, $\frac{L \cdot di}{dt}$, les équations deviennent relativement simples et quasiment toutes les variables sont connues. Un oscilloscope à mémoire permettrait de représenter le courant lors d'un mouvement. Attention à ne pas utiliser le couple afficher via Matlab, car il s'agit en faite d'un pseudo couple, cette problématique est développée au chapitre 8.4.

6 PREMIÈRES AMÉLIORATIONS

Après avoir corrigé les bugs, analysé le fonctionnement du robot et établi une stratégie d'optimisation, le programme d'acquisition, "EX_10.PAS" s'est vu modifié dans l'optique d'effectuer plus de trajectoires. Un paramétrage du temps d'attente entre les mouvements afin de les séquencer a été implémenté. Une première réflexion concernant la quantification des erreurs de vitesse, lors d'un mouvement, a été menée. Celle-ci consistait en l'intégrale de la différence entre la vitesse théorique et réelle d'un mouvement. Mais le sujet n'a pas été poussé plus loin car le rapport intermédiaire devait être rédigé sous peu et les examens de semestres se présentaient aussi durant cette période. Ce bilan intermédiaire a aussi été l'occasion de faire le point avec le professeur responsable de ce Travail de Diplôme, Monsieur Jean-Daniel Marcuard, ainsi que de planifier une seconde visite à Romanel afin d'avoir un échange sur la méthode de la cartographie.

7 BILAN INTERMÉDIAIRE

Un rapport intermédiaire noté a été rendu le 7 mai 2010. Une présentation du projet à ce stade a été effectuée le 25 mai 2010. Le rapport, disponible en annexe 1, résume très brièvement ce qui a été développé jusqu'ici. On est bien loin d'une amélioration concrète des trajectoires du robot mais ce dernier fonctionne désormais correctement et une stratégie d'optimisation a été imaginée. Ce qui donne une direction à explorer pouvant fournir des résultats intéressants.

8 TRAVAIL À PLEIN TEMPS

Les cours et les examens terminés, le projet bénéficiait, dès lors, de toutes les heures disponibles à partir du 18 mai 2010.

8.1 Conséquences du bilan intermédiaire

La manière de réaliser une cartographie est étudiée un peu plus en détail avec Monsieur Jean-Daniel Marcuard. Il en ressort d'analyser un plan de travail, mais uniquement $\frac{1}{6}$ de celui-ci car il est possible de reconstituer, par symétrie des axes décalés de 120° , toute la surface de travail. Les accélérations et les vitesses critiques seraient déterminées en effectuant des mouvements courts, sans jamais atteindre la vitesse maximale, mais en augmentant systématiquement l'accélération. En calculant la somme des erreurs, au carré, de la différence entre la vitesse réelle et théorique du mouvement, l'on obtiendrait une valeur représentative de l'erreur. Cette dernière augmenterait soudainement dès qu'un axe du robot ne serait plus en mesure de suivre l'accélération.

La seconde visite au sein de l'entreprise Bosch Packaging Technology SA tombait à point afin de débattre de cette méthode.

8.2 Seconde visite à Romanel

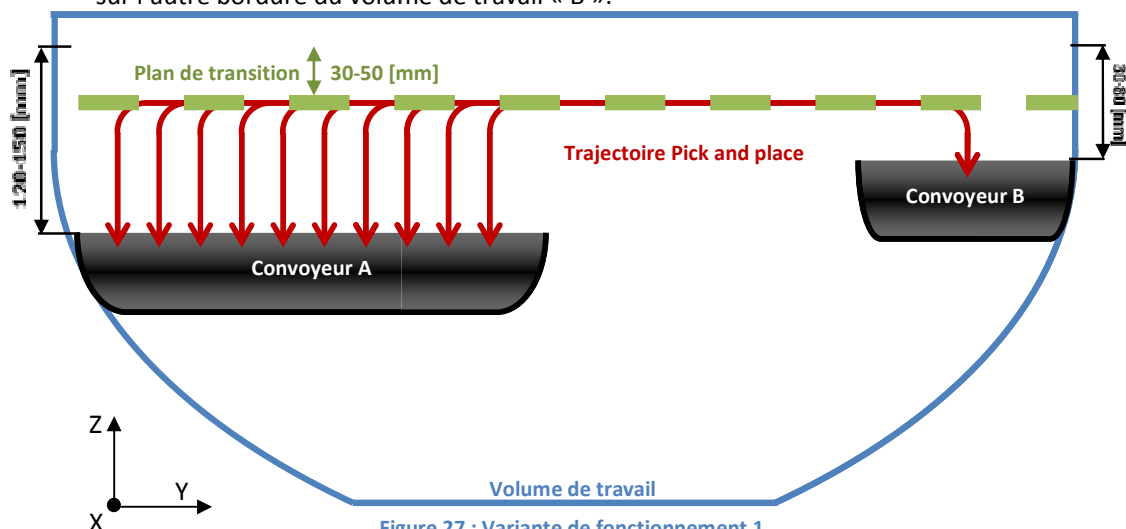
Le 31 mai 2010, un entretien s'est déroulé tout d'abord avec Monsieur Alain Teklits, afin de discuter du travail effectué et idées pour la suite du projet. Après un échange sur la situation, Monsieur Michel Huser prit le relais pour les questions techniques liées au robot. Pendant environ 2h30, les méthodes et directions à prendre pour la suite du projet ont été abordées, dont le fait qu'il serait impossible de fournir une optimisation totale du robot avec le temps restant à disposition. Mais le développement d'une manière pour réaliser une cartographie, à l'aide de la variance, voir 8.7.2, semble prometteur et serait une première grande étape de l'optimisation du robot.

Monsieur Michel Huser donna ensuite des précisions sur le fonctionnement de la machine. Ces dernières changèrent considérablement la manière d'aborder la problématique d'optimisation des trajectoires adoptées jusqu'ici. Plus précisément la manière au robot d'effectuer les mouvements, voir chapitre 8.3, et les précisions au niveau de l'acquisition des données avec la notion des microsamples, voir chapitre 8.9.1. Seule question restée sans réponse, le problème de décalage d'environ 13 milliseconde entre les courbes réelles et théoriques issues des acquisitions, voir chapitre 8.9.2.

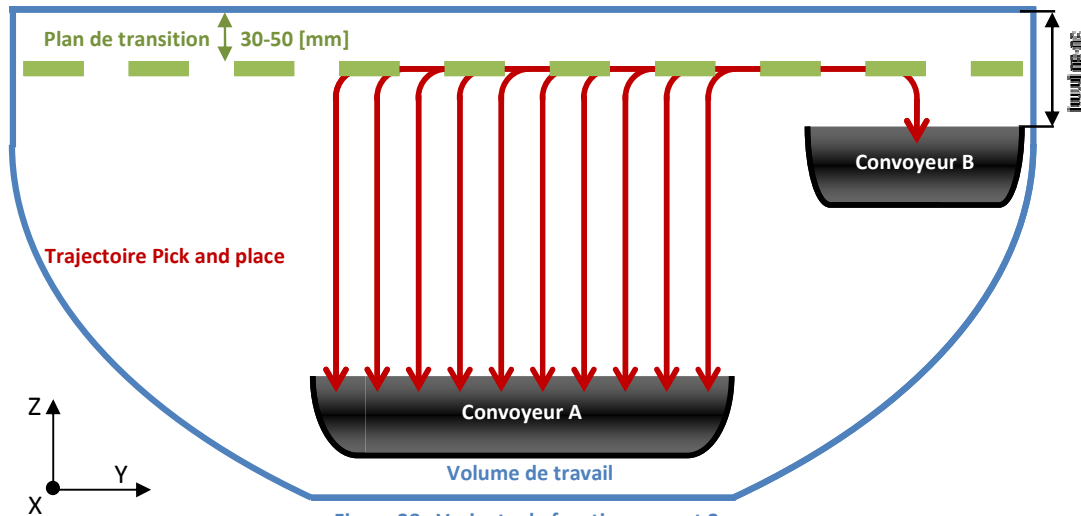
8.3 Principe pratique de la trajectoire du robot

Initialement, il était considéré que le robot effectuait ses mouvements « Pick and Place » dans tout le volume de travail avec des déplacements horizontaux à n'importe quelle hauteur. En réalité, le robot effectue toujours ce type de mouvement à une hauteur bien précise située entre 30 et 50 [mm] sous la limite supérieur du volume de travail. Ce plan de travail obtenu est appelé « plan de transition ». Il y a ensuite deux variantes possibles quand à la manière de faire fonctionner le robot en fonction du convoyeur d'amenée et celui des blisters.

- 1) Le convoyeur d'amenée « A » se situe en bordure du volume de travail et les blisters arrivent sur l'autre bordure du volume de travail « B ».



- 2) Le convoyeur d'amenée « A » se situe au centre du volume de travail et les blisters arrivent en bordure du volume de travail « B ».



8.3.1 Choix du paramètre à améliorer

Il y a donc plusieurs possibilités d'améliorations :

- 1) L'accélération verticale
- 2) La vitesse verticale
- 3) L'accélération horizontale
- 4) La vitesse horizontale

Augmenter l'accélération ou la vitesse lors de la phase de descente du robot, pour saisir ou déposer un objet, pénaliserait fortement la précision de ce dernier. En effet, les "cibles" que le robot doit atteindre sont souvent de petites tailles et il ne serait pas optimal de dégrader la prise et la pose de composants.

Augmenter l'accélération ou la vitesse lors de la phase de montée du robot peut-être une solution mais uniquement lorsque le mouvement se fait à vide. En effet, lorsque celui-ci vient de saisir un objet, une accélération trop élevée aurait pour conséquence que le détachement de celui-ci. Car le système de préhension par le vide ne serait pas assez puissant pour maintenir l'objet et supporter l'accélération du robot en plus de la gravité terrestre. De plus, augmenter la puissance d'aspiration pourrait endommager l'objet à déplacer.

L'accélération et la vitesse horizontale entrent en fonction uniquement dans le plan de transition, si l'on excepte les overlaps. Une vitesse plus élevée ne poserait pas de problèmes majeurs, mais l'accélération serait plus critique. Car c'est lors des phases d'augmentation de vitesse et de freinage que les moteurs sont le plus sollicités.

C'est par cette déduction qu'il a été décidé d'optimiser uniquement l'accélération horizontale et de fixer la vitesse du plan de travail avec une valeur de 4000 [mm/s], ce qui correspond à une consigne utilisée en milieu industriel par Bosch Packaging Technology SA.

8.4 Pseudo couple

Une notion importante a été abordée lors de la seconde visite à Romanel à propos du couple moteur acquisitionné. Comme annoncé avant ce chapitre, ce qui est nommé couple, ou torque dans la documentation de Demarex SA, ne représente pas entièrement la correspondance avec la dénomination du couple en [Nm] (newton mètre). Ce qui est acquisitionné est en fait une valeur de consigne à-priori réelle de la vitesse. Cette valeur a pour destination la carte d'ampli de l'axe en question. Ce problème de dénomination vient du fait qu'initialement, Demarex SA utilisait des cartes d'axe afin de piloter les amplis par des signaux de ± 10 [V] donnant une consigne de courant aux moteurs et non de vitesse comme cela se fait actuellement.

8.5 Symétrie du robot

Une cartographie nécessite un grand nombre de points de mesures afin d'être considérée comme précise. La conception permet de diviser le nombre d'échantillonnage par six. En effet, la disposition des axes décalés par 120° et les données recueillies sur chacun des axes donnent la possibilité de reconstituer entièrement un plan de transition en analysant uniquement un secteur de 60° comme le démontre le schéma ci-dessous.

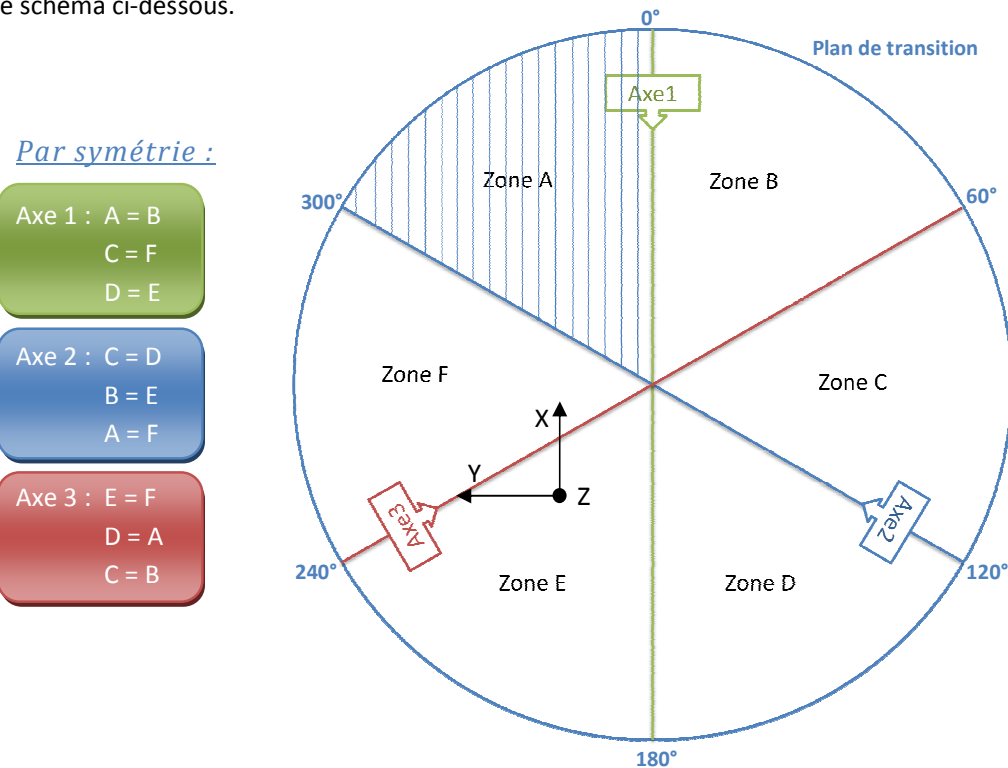


Figure 29 : Zones symétriques du robot

Par déduction, on constate que la zone A pour l'axe 1 est similaire à la zone C pour l'axe 2; de même que la zone E pour l'axe 3.

Ce qui nous donne le tableau de correspondance suivant :

Base	Zone A	Zone B	Zone C	Zone D	Zone E	Zone F
Axe 1	1	1	3	2	2	3
Axe 2	2	3	1	1	3	2
Axe 3	3	2	2	3	1	1

Tableau 1 : Correspondances symétriques

Pour utiliser ce tableau, il faut procéder de cette manière. Si l'on cherche la correspondance de la zone A par rapport à la zone F, on déduit que l'axe 1 dans la zone A correspond à l'axe 3 dans la zone F. Que l'axe 2 dans la zone A correspond à l'axe 2 dans la zone F et que l'axe 3 dans la zone A correspond à l'axe 1 dans la zone F. La figure ci-contre illustre cet exemple.

Zone A	Zone F
1	3
2	2
3	1

Tableau 2 : Exemple de référence

Pour ce Travail de Diplôme, la zone analysée sera la zone A, donc de 0° à 60°.

8.6 Réflexions sur l'algorithme d'assignation

Quelques idées sur la manière d'assigner les consignes d'accélération et de vitesse ont été émises lors de la partie du Travail de Diplôme à temps, voir chapitre 5.6. Il a été défini que la cartographie s'effectuerait désormais uniquement à une hauteur précise, celle du plan de transition. De plus, la vitesse a été fixée, ce qui donne un seul paramètre à analyser, l'accélération.

Lors des premières analyses, les différents types de trajectoires que le robot effectue n'ont pas été pris en considération. En effet, le mouvement peut être purement radial, purement tangentiel ou, une composante des deux. Dans une portion de plan de transition, les efforts sur les moteurs sont différents selon ce type de trajectoire. Cette notion est importante, car cela veut dire que lors de la conception de la cartographie, il faudra réaliser une série d'essais radiaux et une série d'essais tangentiels.

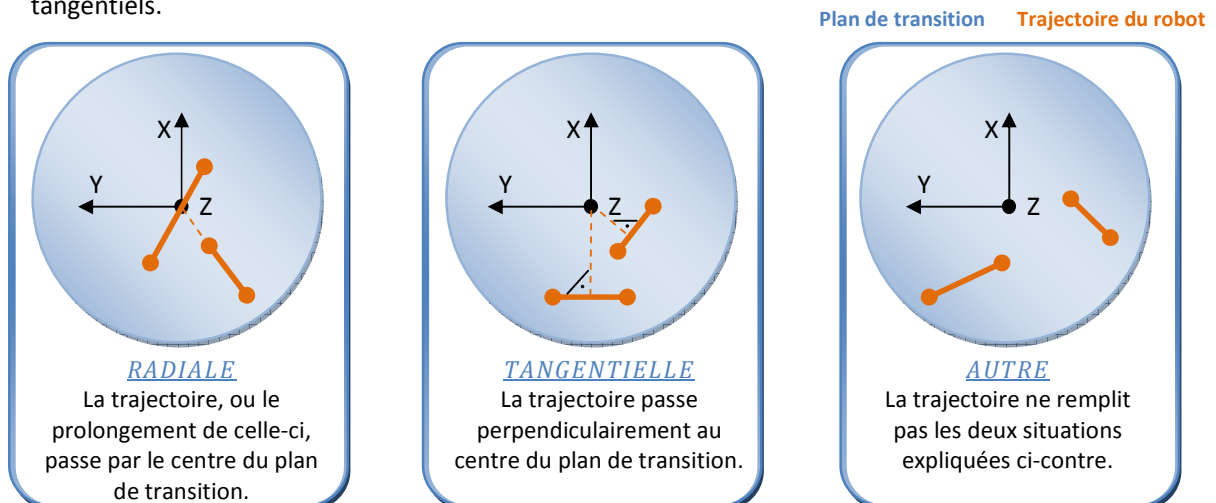
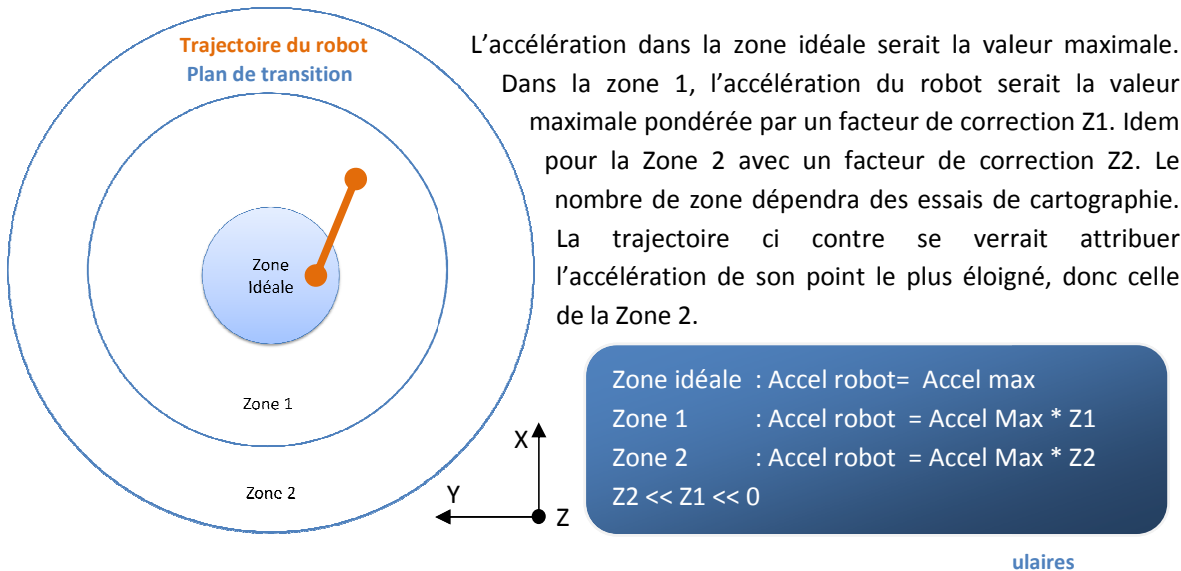


Figure 30 : Types de trajectoires

Cela signifie que lors de l'algorithme d'assignation de la consigne d'accélération, il faudra aussi prendre en compte le fait qu'une trajectoire ne soit pas totalement radiale. La stratégie des zones circulaires assimilée à un triangle qui contrôle l'orientation de la trajectoire remplit cette fonction.

8.6.1 Zones circulaires

Le but de la cartographie est d'obtenir un plan représentant les accélérations maximales admissibles que le robot peut suivre. En prenant en compte ce qui a été dit au chapitre 5.6 et ce qui a été déduit de la visite à Romanel, on obtiendrait un schéma simplifié similaire à la figure ci-contre.



Si les mouvements étaient purement radiaux cette méthode d'assignation serait suffisante.

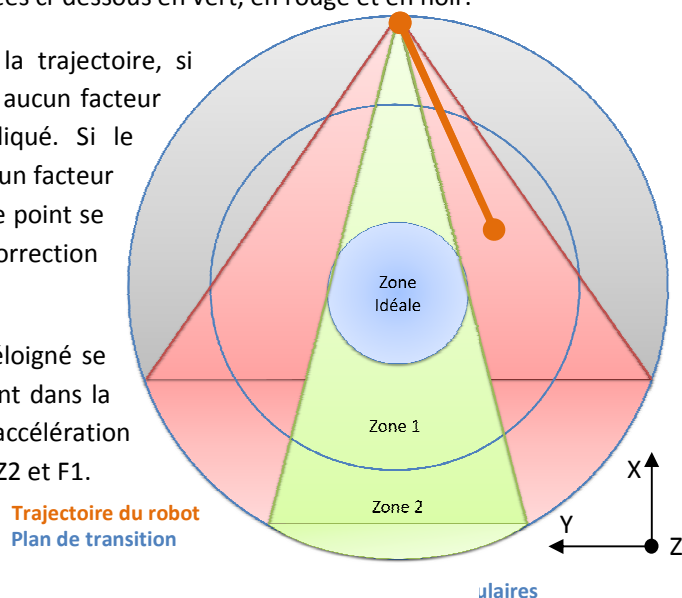
8.6.2 Zones triangulaires

Lorsque la trajectoire, ou son prolongement, se décale par rapport au centre du robot, les efforts sur les moteurs ne sont pas les mêmes. Un premier raisonnement a été de définir, à partir du point de trajectoire le plus éloigné du centre du robot, deux triangles qui diviseraient le plan de transition encore par trois.

Les trois zones supplémentaires sont illustrées ci-dessous en vert, en rouge et en noir.

Par rapport au point le plus éloigné de la trajectoire, si l'autre point se situe dans le triangle vert, aucun facteur de correction supplémentaire n'est appliqué. Si le second point se trouve dans la zone rouge, un facteur de correction est ajouté : F1. Si le deuxième point se situe dans la zone noire, un facteur de correction plus sévère encore sera ajouté : F2.

Pour l'exemple ci-contre, le point le plus éloigné se trouve dans la zone deux, et le second point dans la zone rouge, ce qui donnerait comme accélération finale, la consigne maximale multipliée pas Z2 et F1.



Zone idéale	: Accel robot = Accel max
Zone 1 Verte	: Accel robot = Accel Max * Z1
Zone 1 Rouge	: Accel robot = Accel Max * Z1 * F1
Zone 1 Noire	: Accel robot = Accel Max * Z1 * F2
Zone 2 Verte	: Accel robot = Accel Max * Z2
Zone 2 Rouge	: Accel robot = Accel Max * Z2 * F1
Zone 2 Noire	: Accel robot = Accel Max * Z2 * F2
Z2 << Z1 << 0 et F2 << F1 << 0	

Tableau 4 : Relations des zones triangulaires

Remarque :

A ce stade, les triangles seraient délimités uniquement à partir du point de trajectoire le plus éloigné. L'angle qu'ils forment à partir de ce point n'a pas été étudié plus loin. Pour cet exemple, les angles des triangles ont été choisis arbitrairement afin d'offrir la meilleure compréhension possible.

8.7 Etude de l'accélération limite d'un mouvement

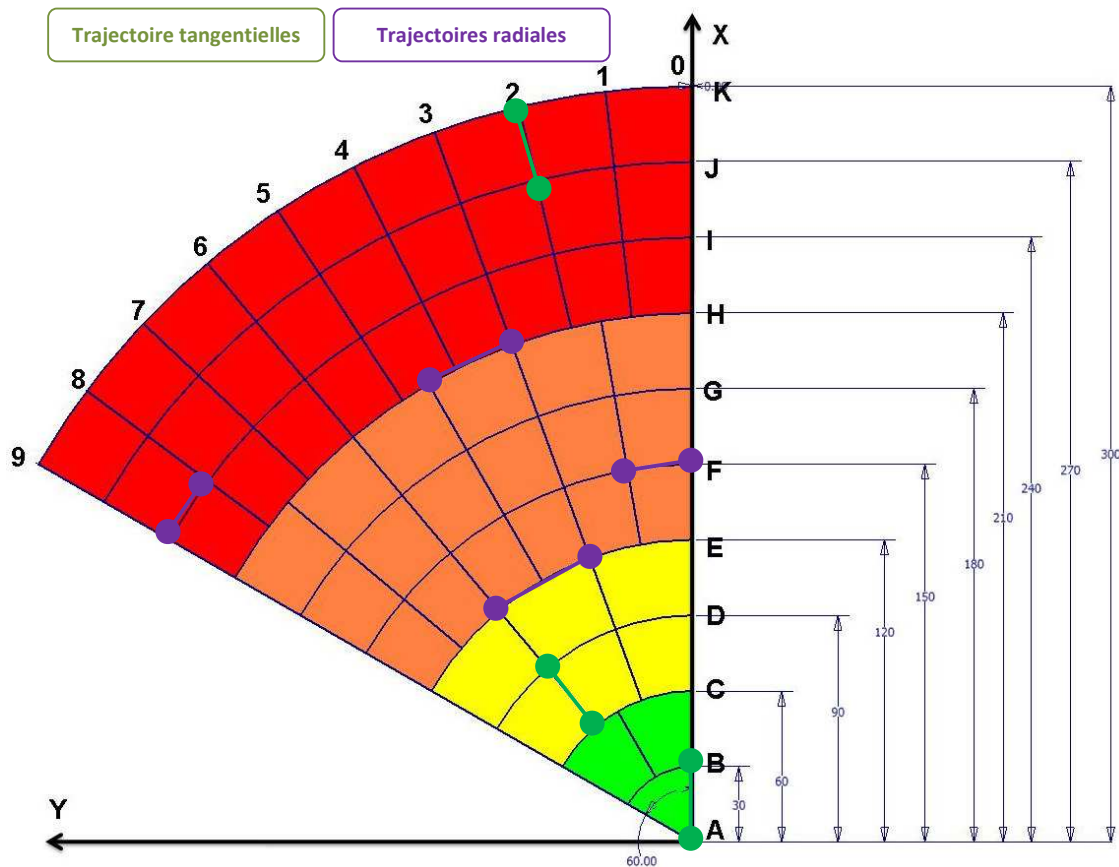
Deux questions importantes ont nécessités une intention particulière lors de ce Travail de Diplôme.

- ◆ Quels mouvements doivent être réalisés afin d'en tirer une cartographie ?
- ◆ Comment trouver l'accélération limite d'un mouvement ?

Le but est d'analyser le comportement du robot à un paramètre d'accélération, qui fait aussi office de paramètre de décélération. Afin d'étudier uniquement ce cas, le robot ne devrait jamais atteindre la vitesse fixée lors d'un mouvement. De plus, afin d'obtenir une bonne précision de la cartographie, les mouvements devraient être relativement courts.

8.7.1 Secteur de mesures

Pour se faire, il a été fixé que le robot effectuera des mouvements d'environ une centaine de millimètres sur toute la surface à analyser. Par le principe de symétrie du robot développé au chapitre 8.5, cette zone se limite à un secteur de 0° à 60°. Afin de donner une bonne représentativité du travail du robot sur cette surface, le diamètre du plan de transition à analyser a été défini à 600[mm] et à une hauteur Z de -350[mm]. Les trajectoires s'effectueront sur une distance de 30[mm]. Cette longueur est facile à définir pour les mouvements radiaux, mais pas pour les tangentiels. En effet, la distance augmente en fonction du rayon du secteur, donc il ne serait pas possible de réaliser systématiquement des mouvements tangentiels de précisément 30[mm]. Une contrainte supplémentaire est à prendre en compte : plus l'on s'éloigne du centre, plus la surface à analyser est grande. Un compromis a été trouvé afin d'aboutir à un secteur de mesure avec plus de mouvement effectués en bordure de celui-ci et une longueur des mouvements tangentiels oscillant entre 26[mm] et 36[mm]. Cette analyse est représentée avec le plan de mesure suivant :



Les points de départs et d'arrivées ont été nommés afin de mieux se repérer, par exemple B0, J3, K9 etc. A noter que le point de l'origine est appelé A0 et ceux qui se situent entre deux nombre portent se dénomment de la manière suivante : B45, G12, H78 etc. En exemple, les codes des trajectoires radiales sont : A0-B0, C6-D6, J2-K2. Elles s'effectuent toujours du centre du secteur vers l'extérieur. Les trajectoires tangentielles en exemple sont représentées par : E3-E5, F0-F1, H3-H45, I8-I9. A ce stade, il y a 64 trajectoires radiales et 45 trajectoires tangentielles. Ce qui donne une résolution totale de 6×64 , donc de 384 points de mesures et 6×45 , donc 270 points de mesures.

L'adresse des points de mesures, ainsi que la distance précise entre eux ont été calculés via un tableau Excel. Une représentation de ce tableau est disponible en annexe 2 ainsi que sur le CD d'annexes informatiques. Toutes ces coordonnées de points sont insérées dans le fichier SOFT_REC.CFG comme démontré au chapitre 5.3.2.

Ce développement répond alors à la question « Quels mouvements doivent être réalisés afin d'en tirer une cartographie ? »

Remarque :

La possibilité d'effectuer, à partir d'un seul point, plusieurs mouvements partant dans des directions différentes, a été abordée. Mais dans l'optique de ne pas rendre encore plus complexe la stratégie de cartographie, ce principe n'as pas été développé plus loin.

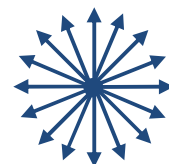


Figure 33 : Point de départs de plusieurs trajectoires

8.7.2 Variance

Avant de parler d'un cycle d'essai, il est important de comprendre la notion statistique de la variance. C'est une mesure permettant de quantifier la dispersion de plusieurs échantillons. Textuellement parlant, c'est la moyenne des écarts au carré et elle est mathématiquement représentée par le symbole sigma au carré « σ^2 ».

$$\sigma^2 = \frac{1}{n} \sum_{i=1}^n (x_i - x)^2$$

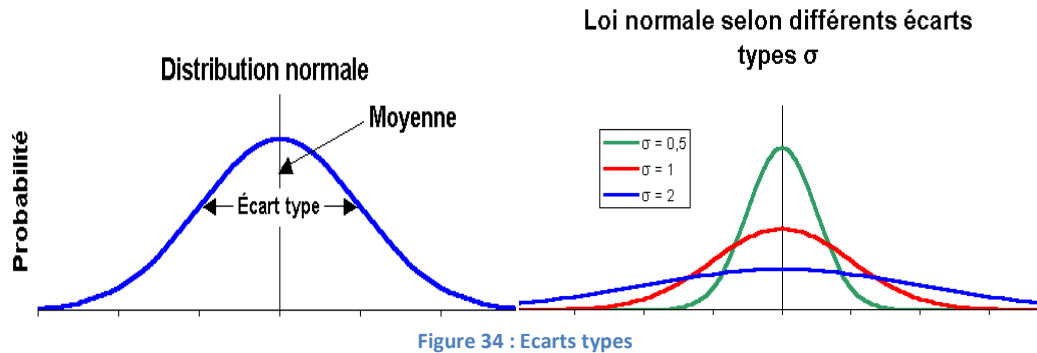
Formule 2 : Variance

L'écart type est une mesure de la dispersion de plusieurs échantillons autour de leur moyenne. C'est la racine carrée de la variance, représentée par le symbole sigma « σ ».

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - x)^2}$$

Formule 3 : Ecart type

Ci-dessous, la courbe de gauche représente l'écart type en fonction du nombre d'échantillons. Plus l'écart type est élevé, plus il y a de dispersion, comment le démontre les courbes de droite. Vu que la variance n'a pas de racine carrée, sa valeur est donc beaucoup plus élevée et pénalise fortement un écart par rapport à la moyenne. En résumé, plus il y a de dispersion autour de la moyenne de la courbe d'un mouvement, plus la variance sera élevée.



8.7.3 Cycle d'essais

Afin de ressortir l'accélération critique d'une trajectoire, le mouvement est réalisé depuis une petite consigne, puis répétée avec une valeur de consigne d'accélération de plus en plus élevée jusqu'à une valeur choisie arbitrairement. A chaque mouvement, l'erreur de vitesse de chaque axe est traitée afin d'obtenir sa variance. Ci-contre, le graphique en haut représente le cas où la vitesse réelle d'un axe moteur parvient à suivre la vitesse théorique, spécialement sa pente d'accélération. Ce qui donnera une variance faible. Le graphique en bas représente le cas où l'axe moteur ne parvient pas à suivre la consigne d'accélération, ce qui donnera une variance élevée.

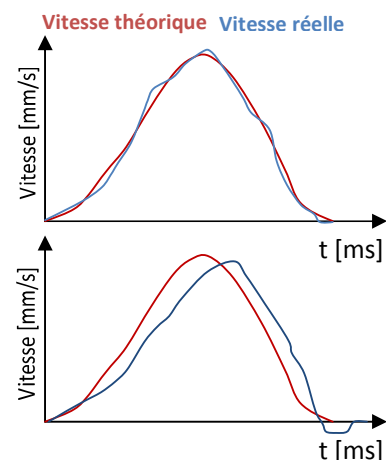


Figure 35 : Allure d'une erreur de vitesse

La dénomination de la variance pour l'erreur de vitesse est définie comme suit : $\sigma^2 = \left[\frac{mm^2}{s^2} \right]$. L'exemple théorique a été vérifié par un essai pratique avec le « D95Conf ». A gauche, le robot suit parfaitement la courbe théorique, tandis qu'à droite, le robot est en retard.

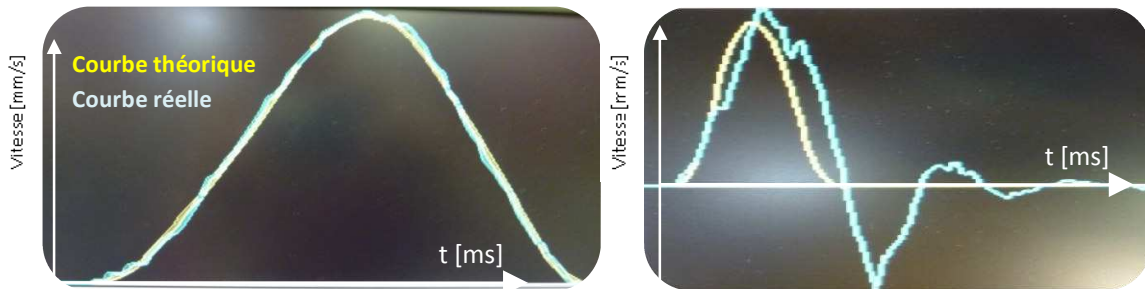


Figure 36 : Allure de l'erreur de vitesse sous "D95Conf"

8.7.4 Détermination de l'accélération critique

Ces valeurs de variance sont ensuite représentées sur un graphique. Dès qu'un axe voit sa variance soudainement augmenter, cela signifie qu'il ne parvient plus à suivre la consigne d'accélération à l'instant où la courbe se met à croître rapidement. Le graphique suivant illustre ce principe.

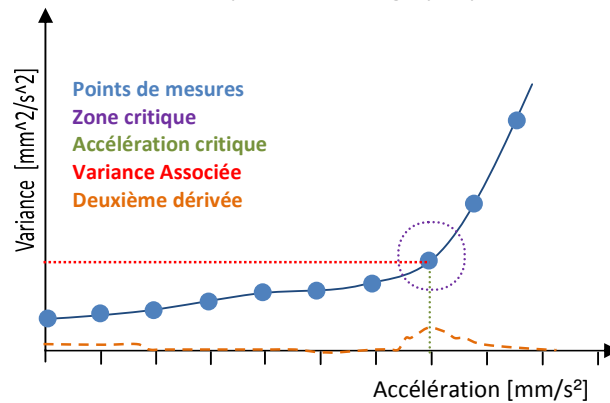


Figure 37 : Augmentation de la variance

L'accélération critique se situe alors à la hauteur de la zone critique. Une méthode pour trouver automatiquement cette zone critique peut-être réalisée à l'aide du maximum de la 2^{ème} dérivée de la courbe. En effet, la 2^{ème} dérivée détermine si une fonction mathématique effectue une courbe concave, dans ce cas la valeur de la différentielle au carré est négative, ou si une fonction réalise une courbe convexe, la valeur est alors positive. Dans notre cas, l'on désire détecter le « virage » de notre graphique de forme convexe. Donc si les graphiques se présente comme ci-dessus en pratique, le principe serait d'une grande efficacité et permettrait d'automatiser les cycles d'essais. L'accélération critique serait ainsi déterminée par le premier des trois axes du robot voyant sa variance soudainement augmenter.

8.8 Implémentation du cycle de variance

Afin de confirmer la théorie de l'essai de variance, le principe a été implémenté dans le fichier « EX_10.PAS » avec la possibilité de choisir manuellement les coordonnées de départ et d'arrivée du mouvement ainsi que son accélération de départ « Accelmin », celle d'arrivée « Accelmax » et l'incrément de la consigne « AccelStep ».

8.8.1 Programme du cycle

L'index du programme entre dans une boucle de cycle du nombre de mouvements à réaliser « NbStepReal ». Le robot se déplace lentement jusqu'au point de départ, reçoit la consigne de vitesse par défaut (4000[mm/s]) ainsi que sa consigne d'accélération relative au mouvement.

Le temps du premier mouvement, celui avec l'accélération minimale, est ensuite calculé afin de réaliser l'échantillonnage des mouvements, mais ce paramètre n'a finalement pas été utilisé. Une pause de stabilisation de 200[ms] précède l'enclenchement de l'enregistrement de la mesure avec la variable « Startmoving ».

Le mouvement jusqu'au second point est lancé, suivi d'une pause de stabilisation de 400[ms], puis la fin de

```
{Boucle de cycle}
FOR RunStep:= 0 TO NbStepReal DO
BEGIN
{Mouvement lent pour le point de départ}
D3n_Accel(left,500);
D3n_Speed(left,500);
TestingMovement(point1,locName1);
{Assignment des paramètres d'accélération et de
vitesse}
NewAccel:=AccelMin+AccelStep*(RunStepReal);
D3n_Accel(left,newAccel);
D3n_Speed(left,4000);
{Calcul de temps du premier mouvement}
IF RunStep = 0 THEN
BEGIN
D3n_PrepareAbsMove(left,End_X,End_Y,End_Z,nil,d3mp);
PrepareTargetTime := D3n_Q_PrepareTargetTime(left);
END;
{Pause de stabilisation}
TimeWait(0.2);
{Lancement de la procédure d'enregistrement}
StartMoving:=true;
{Mouvement}
TestingMovement(point2,locName2);
{Pause de stabilisation}
TimeWait(0.4);
Arret de la procédure d'enregistrement}
StartMoving:=false;
{Incrementation du mouvement}
RunStepReal := RunStepReal+1;
END
```

Code 1 : Cycle de mouvement

l'enregistrement de la mesure. L'index est ensuite incrémenté pour réaliser le mouvement suivant avec la consigne d'accélération supérieure. Cette partie du code se situe à la ligne 2502 de « Cartra1.PAS » en annexe 3 et sur le CD.

8.8.2 Echantillonnage et deuxième dérivée sous Matlab

Le logiciel Matlab, sous le PC Superviseur, récupère les données des du cycle d'essais dans les fichiers DATA.INT, INFOTIME.INT et INFOPARAM.INT et il intègre dans un tableau « tab_final », les valeurs des mesures. Les valeurs théoriques et réelles de la vitesse sont extraites du tableau. Leurs différences sont stockées dans un nouveau tableau « Diff1 ». Celui-ci comporte alors l'erreur de vitesse d'un cycle, il va être échantillonné. L'affichage d'une partie des valeurs du tableau « Diff1 » donne le graphique ci-contre.

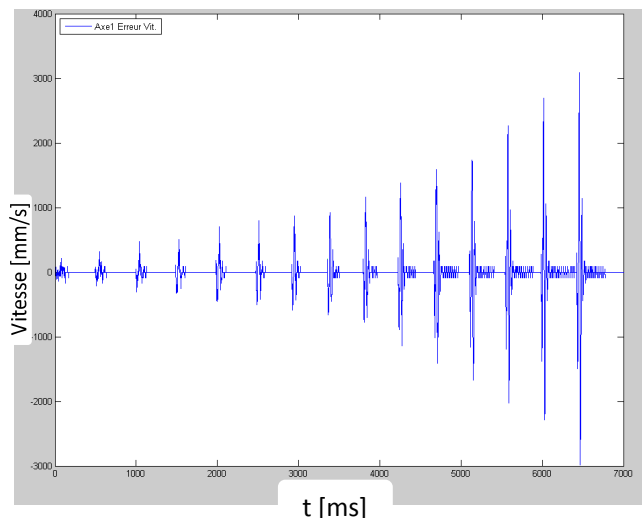


Figure 38 : Graphique Matlab des erreurs de vitesse

Le but est maintenant d'échantillonner ces pics d'erreur de vitesse qui correspondent chacune à un mouvement à une accélération précise.

Une boucle d'échantillonnage a pour but de remplir un tableau qui contient la valeur de la variance de chaque mouvement. Elle est directement suivie par une boucle de recherche de début de mouvement. La principale difficulté a été de trouver les bonnes conditions pour détecter le début d'un mouvement sur la courbe.

```
Boucle d'échantillonnage
for Step=0 : NbStep-1,
next=0;
%Boucle de début de mouvement
while next==0;
.....
end;
end;
```

Code 2 : Boucle d'échantillonnage

La formule finale de cette condition n'a pas été facile à trouver et a été déterminée au fil des essais. Cette condition est la suivante :

```
if ( t == 1 || ((abs(Diff1(t-3))+abs(Diff1(t-2)))+ abs(Diff1(t-1)))<=5) &&
Diff1(t)~=0 && ((abs(Diff1(t+1)) + abs(Diff1(t+2)) + abs(Diff1(t+3)) + abs(Diff1(t+4)) +
abs(Diff1(t+5)) + abs(Diff1(t+6))+ abs(Diff1(t+7)))>150) ),
...
end :
```

Code 3 : Condition de détection de mouvement

Mise à part le début du premier mouvement, la somme des 3 valeurs précédentes l'index doit être plus petite ou égale à 4 [mm/s]. La valeur de l'index doit être différente de 0 [mm/s] et la somme des 7 valeurs suivante doit être plus grande que 150 [mm/s].

Dès qu'un début de mouvement est détecté, sa valeur est inscrite dans le tableau « Startendstep », de même que pour sa valeur de fin de mouvement. Celle-ci correspond toujours à 350[ms] après le début d'un mouvement. Ceci afin de délimiter dans le tableau des erreurs de vitesse « Diff1 », les bonnes valeurs à traiter pour en sortir la variance. Toutes les variances sont ensuite stockées dans le tableau « TabVar1 », ici pour l'axe 1, et les accélérations du cycle dans le tableau « TabAccel1 », ici toujours pour l'axe1. La recherche du mouvement suivant démarre automatiquement 350[ms] après un début de mouvement trouvé.

```
%Début de l'échantillonnage
startendstep(1,(Step+1)) = t;
%Fin de l'échantillonnage
startendstep(2,(Step+1)) = t+350;
%si l'index dépasse le max du tableau
if startendstep(2,(Step+1)) >1,
endstep = 1 ;
end;
Définition de la variance
Variance1 =
var(Diff1(startendstep(1,(Step+1)):startendstep(2,(Step+1))));
%Recherche de mouvement suivant
next=1;

%Assignment dans un tableau de la variance correspondante
TabVar1(Step+1)=Variance1;
%Assignment dans un tableau de l'accélération liée à la variance
TabAccel1(Step+1)=(AccelMin+AccelStep*Step);
%Recherche du mouvement suivant
t=t+350;
```

Code 4: Echantillonnage

Afin de faciliter la représentation, une barre de couleur verte en indique le début et une rouge la fin sur le graphique des erreurs de vitesses.

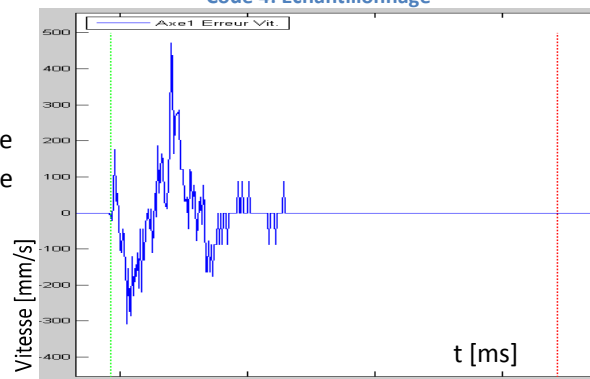


Figure 39 : Délimitation de l'échantillonnage

Comme le démontre le code ci-contre, le tableau contenant les variances est ensuite affiché en fonction des accélérations correspondantes. La deuxième dérivée de la courbe des variances est aussi calculée et affichée en plus de droites indiquant l'accélération critique ainsi que la valeur de sa variance correspondante. Le graphique suivant est alors obtenu :

```
%Plot de la courbe d'échantillonnage
plot(TabAccel1,TabVar1, '-g');

%Dérivée première et seconde
deriv11 = gradient(TabVar1);
deriv12 = gradient(deriv11);
%maximum de la deuxième dérivée
[yd1, vd1]=max(deriv12);
%Assignment des valeurs correspondantes
AccelCritAxe1=TabAccel1(vd1-1);
VarCritAxe1=TabVar1(vd1-1);
%lignes des valeurs critiques
plot([AccelCritAxe1 AccelCritAxe1],[0 VarCritAxe1], '-.g')
plot([AccelMin AccelCritAxe1],[VarCritAxe1 VarCritAxe1], '-.g')
```

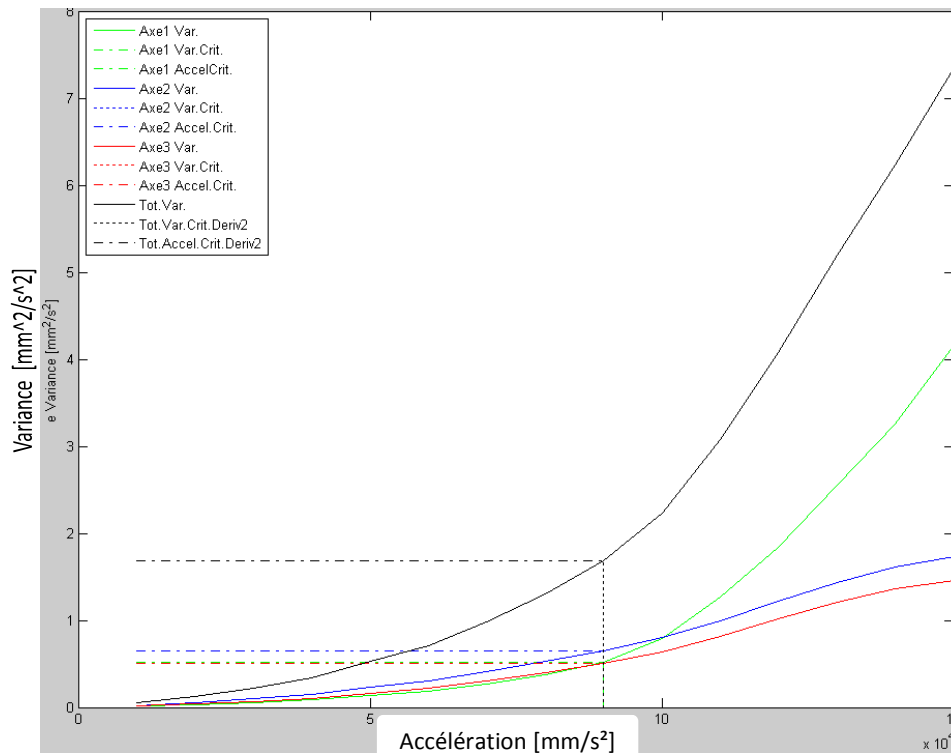


Figure 40 : Graphique Matlab de la variance

Les courbes ressemblent bien à ce qui a été avancé théoriquement, mais la méthode du maximum de la deuxième dérivée pourrait être imprécise selon l'allure de la courbe.

Remarque :

La manière dont Matlab traite ces données et les stocke dans le tableau « tab_final » n'est pas abordée en détail dans ce travail de diplôme. La raison est que ce début de script a été développé lors du précédent Travail de Diplôme réalisée par Monsieur Pierre Fournier en 2009. Mais vu que ce bout de programme a été repris, il a subi une modification et il a été commenté. Voir chapitre 8.9.2

8.9 Décalage des données

Comme annoncé au chapitre 5.3.4, les courbes théoriques et réelles issues de l'acquisition sous Matlab possèdent un décalage d'environ 13 millisecondes, alors que sur le logiciel D95CONF, ce décalage n'est pas présent. Afin d'obtenir des courbes vraiment représentatives de la réalité, il était important de corriger ce problème. Il a été nécessaire de se plonger dans le procédé d'acquisition des données du robot ainsi que la manière de les traiter, ce qui a été développé lors du Travail de Diplôme précédent.

8.9.1 Principe des microsamples

Lors de la seconde visite à Romanel, Monsieur Michel Huser était convaincu que le problème de décalage ne pouvait pas provenir de la programmation du robot mais uniquement du traitement sous Matlab si la configuration des microsamples était bien réalisée. Monsieur Michel Huser a approfondi la théorie sur ce sujet et a recommandé de contrôler que les paramètres soient bien les bons. Cette notion des microsamples est relativement délicate à comprendre et difficile à expliquer mais un petit explicatif est présenté ci-dessous. Le sujet est bien plus approfondi dans le manuel d'utilisateur « User's Guide » de Bosch Packaging Technology SA à la page 25 environs. Ce document se trouve en annexe sur le CD dans le dossier « Documentation Demaurex ».

Le robot dispose d'une interruption régulière paramétrable qui a pour but de lire les données des mouvements des trois axes des dernières millisecondes, de les traiter et d'envoyer les consignes aux trois axes pour les prochaines millisecondes. Ce nombre de milliseconde est paramétrable avec les variables et leur valeur par défaut suivantes :

- ◆ IrqPeriodCount = 10 [ms] : Représente la période à laquelle l'interruption se déclenche.
- ◆ IrqDelayCount = 1 [ms] : Temps d'attente entre la réception de la commande périodique du PC îlot et la génération de l'interruption suivante.
- ◆ BasicPeriod = 1 [ms] : Période basic de travail, la microsample

Avec ces valeurs, paramétrable dans le fichier HARD_REC.CFG, le diagramme de temps correspond à ce qui est défini dans le manuel utilisateur et représenté par la figure de la page suivante.

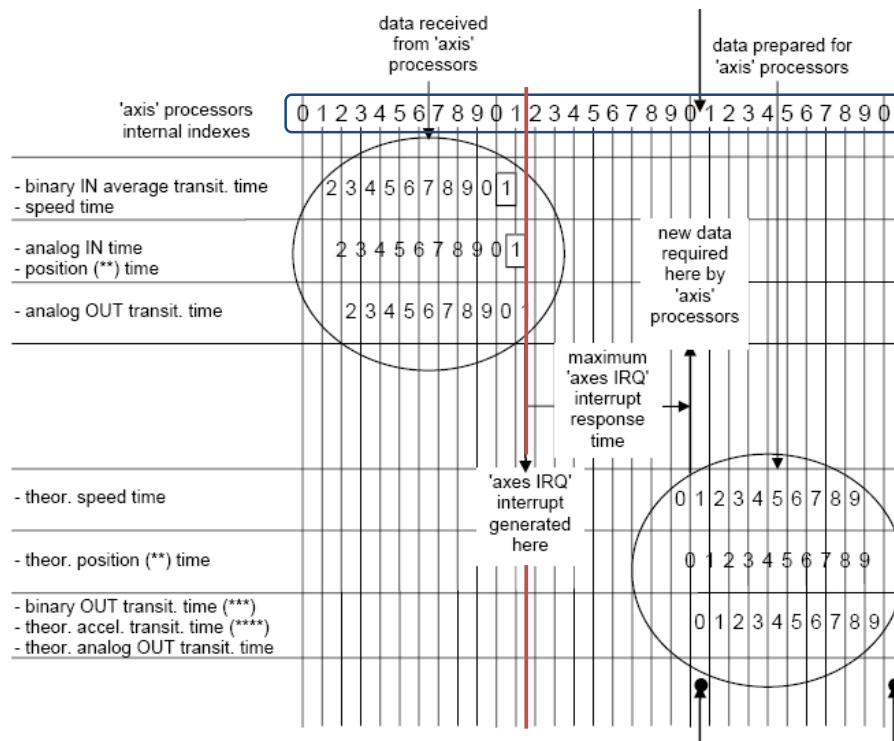


Figure 41 : Principe des microsamples

L'interruption analyse donc les 10 dernières millisecondes et les interprète unitairement. Ce qui est important de comprendre c'est qu'il subsiste différents décalages entre l'acquisition des valeurs mesurées des axes et l'envoi des valeurs théoriques calculées pour la suite. Tout en haut du graphique, en bleu, ce situe l'index interne du processeur des axes, la base de temps en quelque sorte. Sur la gauche du graphique, les différents paramètres analysés. L'on constate que pour la mesure de la vitesse, sa valeur est au fait enregistrée une demi-milliseconde en avance par rapport à l'index du processeur, ce qui n'est pas pris en compte dans la gestion de l'acquisition. Il est intéressant de constater que le premier microsample analysé de la vitesse n'est pas 0, ni 1, mais 2, ce qui influe donc le remplissage des tableaux de l'acquisition dans le programme en TurboPascal.

Sur le graphique est représenté trois périodes d'interruptions, si l'on se base par rapport au temps, l'interruption est actuelle est représentée par la barre rouge. A cet instant, le robot dispose d'au maximum 10 [ms] pour traiter les 10 [ms] précédentes et envoyer la commande pour les 10 prochaines [ms]. Cela signifie que les valeurs réelles et théoriques sont alors décalées mais de 10 [ms], ce qui ne correspond pas totalement avec le décalage de 13[ms] initialement observé.

Suite à ce raisonnement, un contrôle des paramètres a été mené dans le programme d'acquisition de Monsieur Pierre Fournier qui a été initialement repris pour ce Travail de Diplôme. Il s'est avéré qu'il ne comportait pas tout à fait les mêmes paramètres. En effet, la variable « irqPeriodCount » avait comme valeur 5[ms]. Ce qui veut dire que l'interruption d'acquisition était réalisée chaque 5[ms] et non chaque 10[ms] comme expliqué ci-dessus. Néanmoins, la suite de la programmation était adaptée pour cette modification, donc ne pouvait pas avoir d'incidence sur ce décalage. Dans l'optique d'avoir une configuration similaire au manuel d'utilisateur et afin de faciliter la compréhension, la valeur de l'« irqPeriodCount » a été ramenée à 10[ms]. Les modifications nécessaires on été menées dans le programme d'acquisition « EX_10.PAS », afin de le rendre

compatible avec la nouvelle configuration. Ces modifications concernent la taille des tableaux enregistrant les acquisitions, passant de l'index maximum 5 à 10. Voir annexe 3.

8.9.2 Correction du décalage sous Matlab

Le problème du décalage n'étant toujours pas résolu, le bout de programme réalisant le décodage et la structuration des valeurs mesurée a été étudié plus en détail. Cette partie de code se trouve en annexe 4. Dans la boucle de remplissage du tableau des valeurs finales « tab_final », le code effectuait un décalage de l'index de ligne « l » de 5 si l'index de colonne « c » correspondant à un chiffre pair jusqu'à 18.

```
if c==2 || c==4 || c==6 || c==8 || c==10 || c==12 || c==14 || c==16 || c==18  
    l = l+5;  
end ;
```

Chaque colonne correspond à un paramètre mesuré comme la position réelle de l'axe 1, la vitesse théorique de l'axe 3 etc. Une colonne paire correspond toujours à une valeur théorique tandis qu'une colonne impaire, une valeur réelle. Ce procédé permet donc de décaler les mesures réelles par rapport aux théoriques. L'incréméntation étant en lien avec l'« irqPeriodCount » de la version précédente du code, elle a aussi été ramenée à 10 [ms] afin de correspondre avec les modifications.

Sans comprendre finalement pourquoi ce décalage de 13[ms] était toujours présent alors qu'en théorie tout paraissait juste, il a été décidé de réaliser un petit programme permettant de trouver et de fixer la valeur de l'incréméntation du décalage. Un essai de variance particulier a alors été conçu avec le robot. En effet, celui-ci effectuait toujours le même mouvement avec une même consigne d'accélération et de vitesse relativement lente afin de minimiser les erreurs de mouvement. Un traitement de variance a ensuite été réalisé mais avec comme variable, la valeur de l'incréméntation du décalage. Avec cet essai, vu que les erreurs de vitesses et de positions on été minimisés, le minimum de la courbe en forme de U indiquerait la valeur d'incréméntation juste. Ce graphique est représenté ci-dessous. La valeur d'incréméntation a alors été fixée à 18 et le décalage n'est plus apparu.

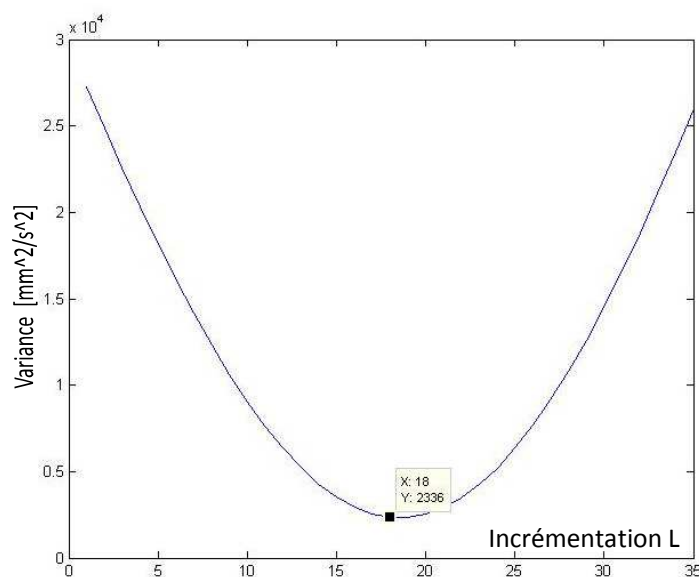


Figure 42 : Variance déterminant l'incréméntation L

8.10 Séries d'essais

Un cycle de mouvement de variance entre deux points correspond à un essai, il y en a 64 pour les essais radiaux et 45 pour les tangentiels. Afin de faciliter la programmation de ceux-ci dans le fichier EX_10.PAS, ils ont été divisés en séries. Donc 8 séries des 8 essais pour les mouvements radiaux et 5 séries de 9 essais pour les mouvements tangentiels. Chaque cycle de mouvements est assimilé à un code numérique dans la programmation afin d'identifier rapidement les points de mesures lors de la création de la cartographie. Les graphiques suivants illustrent ce codage.

Série cartographique radiale 64 points											
Série 1			Série 2			Série 3			Série 4		
Code	Start	End	Code	Start	End	Code	Start	End	Code	Start	End
201	A0	B0	209	C9	D9	217	E45	F45	225	F6	G6
202	A0	B9	210	D0	E0	218	E6	F6	226	F78	G78
203	B0	C0	211	D3	E3	219	E78	F78	227	F9	G9
204	B45	C45	212	D6	E6	220	E9	F9	228	G0	H0
205	B9	C9	213	D9	E9	221	F0	G0	229	G12	H12
206	C0	D0	214	E0	F0	222	F12	G12	230	G3	H3
207	C3	D3	215	E12	F12	223	F3	G3	231	G45	H45
208	C6	D6	216	E3	F3	224	F45	G45	232	G6	H6
Série 5			Série 6			Série 7			Série 8		
Code	Start	End	Code	Start	End	Code	Start	End	Code	Start	End
233	G78	H78	241	H6	I6	249	I4	J4	257	J2	K2
234	G9	H9	242	H7	I7	250	I5	J5	258	J3	K3
235	H0	I0	243	H8	I8	251	I6	J6	259	J4	K4
236	H1	I1	244	H9	I9	252	I7	J7	260	J5	K5
237	H2	I2	245	I0	J0	253	I8	J8	261	J6	K6
238	H3	I3	246	I1	J1	254	I9	J9	262	J7	K7
239	H4	I4	247	I2	J2	255	J0	K0	263	J8	K8
240	H5	I5	248	I3	J3	256	J1	K1	264	J9	K9

Figure 43 : Tableau des codes radiaux 64 points

Série cartographique tangentielle 45 point														
Série 1			Série 2			Série 3			Série 4			Série 5		
Code	Start	End	Code	Code	Code	Code	Start	End	Code	Start	End	Code	Start	End
301	B0	B9	310	337	337	337	G45	G6	328	I0	I1	337	J0	J1
302	C0	C45	311	338	338	338	G6	G78	329	I1	I2	338	J1	J2
303	C45	C9	312	339	339	339	G78	G9	330	I2	I3	339	J2	J3
304	D0	D3	313	340	340	340	H0	H12	331	I3	I4	340	J3	J4
305	D3	D6	314	341	341	341	H12	H3	332	I4	I5	341	J4	J5
306	D6	D9	315	342	342	342	H3	H45	333	I5	I6	342	J5	J6
307	E0	E3	316	343	343	343	H45	H6	334	I6	I7	343	J6	J7
308	E3	E6	317	344	344	344	H6	H78	335	I7	I8	344	J7	J8
309	E6	E9	318	345	345	345	H78	H9	336	I8	I9	345	J8	J9

Figure 44 : Tableau des codes tangentiels 45 points

La série tangentielle n'est pas réalisée sur la dernière bordure en « K » car cela ferait onze rayons de mesures alors qu'avec la série radiale on parvient à dix rayons de mesures.

Remarque :

Le numéro de code « 200 » est réservé pour un essai de variance avec des coordonnées sélectionnées manuellement. Dans ce cas, les code « start » et « end » prennent comme nom : « Manuel ».

8.10.1 Implémentation sous le programme « EX_10.PAS »

Les séries sont programmées dans une procédure qui est appelée avant chaque cycle. Le code des trajectoires est simplement incrémenté pour passer à l'essai suivant. Le code ci-dessous est une partie tronquée de cette procédure. Ligne 1676 du code « CARTRA1.PAS » en annexe 3.

```
{-----}
PROCEDURE AssignSerie;
{Cette procédure assigne les coordonnées          }
{du mouvement en fonction du code de trajectoire }
{-----}
BEGIN
{Série Tangentielle 1}
  IF (trajectoire >= 301) AND (trajectoire <=309) THEN
  BEGIN
    CASE Trajectoire OF

      301: BEGIN
        point1:= pointB0;
        point2:= pointB9;
        locName1:='B0';
        locName2:='B9';
        END;
      ...
      309: BEGIN
        point1:= pointE6;
        point2:= pointE9;
        locName1:='E6';
        locName2:='E9';
        END;
    END;
  END;
  ...
END;
```

Code 5: Procédure "AssignSerie"

8.11 Améliorations du programme Matlab

Initialement, le traitement des fichiers de données sous Matlab était lancé en appelant le script « read_and_plot .m ». Ce dernier affichait les différentes courbes de valeurs des différents axes. Afin d'améliorer le traitement des données, le programme a été complètement restructuré. La nouvelle architecture permet de :

- ◆ Naviguer dans le programme de menu en menu.
- ◆ Choisir les courbes à afficher, ce qui ne ralentit plus le PC.
- ◆ Choisir le type d'analyse de la variance et la personnaliser.
- ◆ Sauvegarder l'analyse de l'essai et ses données pour un traitement ultérieur ou pour la réalisation d'une série de cartographie
- ◆ Retraiter une archive d'essai et en retirer les courbes désirées
- ◆ Afficher une cartographie en 3D selon un paramètre précis
- ◆ Choisir le type de cartographie

Le programme est désormais lancé en entrant « Delta » dans la ligne de commande de Matlab. Elle fait en réalité appel au script de base « Delta.m »

8.11.1 Navigation

L'affichage des différents menus ainsi que la navigation entre eux sont détaillés dans le mode d'emploi en annexe 5 et sur le CD. Les structogrammes en annexe 6, détaillent le fonctionnement des différents scripts sous Matlab.

```

+-----+
| GESTION DE DONNEES DU ROBOT DELTA |
| Logiciel integre basé sur MATLAB |
+-----+
| Auteurs : Briguet Michael          |
| Version : 2.0                      |
| Sion, le 05 Juillet 2010           |
+-----+
|                                     |
| VEUILLEZ SIGNALER TOUT DEFAUT AUX AUTEURS SVP ! |
|                                     |
+-----+
| MENU PRINCIPAL                     |
+-----+
| 0 : Quitter                         |
|                                     |
| 1 : Nouvelle acquisition            |
| 2 : Ancienne acquisition            |
|                                     |
| 3 : Cartographie d'une série de test |
|                                     |
| 4 : Informations sur le programme   |
+-----+
| Votre reponse SVP : |
+-----+
```

Figure 45 : Accueil du programme "Delta" sous Matlab

Remarque :

La programmation de la navigation et la gestion des fichiers a été inspirée à partir du logiciel « CAO réglage » fourni par Monsieur Jean-Daniel Marquard. Ce programme est disponible en annexe sur le CD.

8.11.2 Nouvelle acquisition

La première fonction du programme Delta sous Matlab est le traitement d'une nouvelle acquisition. C'est le script « New_Acqui » qui est appelé afin de réaliser cette fonction. Il permet de choisir les graphiques à afficher, de sélectionner quel type de traitement de variance à réaliser, si c'est un essai de variance.

Les graphiques suivants peuvent être affichés pour chacun des trois axes :

- ◆ Position réelle et théorique
- ◆ Erreur de position
- ◆ Vitesse réelle et théorique
- ◆ Erreur de vitesse
- ◆ Variance

A noter que les graphiques du couple ne sont plus affichés car il ne s'agit pas du vrai couple, ce qui n'apporte pas de réel intérêt pour l'instant, voir le chapitre 8.4. Mais son code a été tout de même conservé et adapté au nouveau concept du programme. Son code a simplement été mis en commentaire.

La variance des trois axes est représentée sur le même graphique afin de faciliter son analyse et alléger le nombre de fenêtres ouvertes

```

NOUVELLE ACQUISITION
choix de l'affichage
1 : Axes 1,2,3 : Position, Vitesse, Erreurs : 1
2 : Axes 1,2,3 : Position, Vitesse, Erreurs, Variances : 2
3 : Axes 1,2,3 : Variances et Erreurs de vitesse (echant.) : 3
4 : Axes 1,2,3 : Variances uniquement (+ rapide) : 4
votre reponse SVP :
    
```

Figure 46 : Nouvelle acquisition sous Matlab

8.11.3 Types d'analyse et variance plateau

Initialement, l'accélération critique était uniquement déterminée lorsque le premier des trois axes voyait sa valeur de variance soudainement augmenter, voir chapitre 8.7.4 . C'est d'ailleurs ce type de traitement qui a été appliqué sur les premières cartographies, voir chapitre 8.18. Néanmoins, ce type d'analyse comporte un défaut. En effet, selon les points de mesures, les accélérations critiques pourraient être les mêmes mais à des erreurs de variances bien différentes. Cela ne serait pas convenable si l'on désire des mouvements avec une grande précision, donc en limitant les erreurs. Afin de palier à ce problème, une seconde technique pour déterminer l'accélération critique a été mise au point. Au lieu de détecter l'augmentation soudaine de l'erreur de variance pour en déduire l'accélération critique, c'est une valeur de variance limite qui, lorsqu'elle est atteinte, détermine la valeur de l'accélération qui en est associée. L'utilisateur peut alors choisir trois types d'aides de traitement :

- ◆ L'analyse du maximum de la deuxième dérivée.
- ◆ L'analyse de variance plateau par défaut (150'000 [mm²/s²])
- ◆ L'analyse de variance plateau par une valeur entrée par l'utilisateur.

```

NOUVELLE ACQUISITION
Type de Variance
1 : Maximum de la 2ème dérivée de la Variance la plus élevée
2 : Plateau de Variance limite par défaut (150000 [mm2/s2])
3 : Plateau de Variance limite personnalisé
votre reponse SVP :
    
```

Figure 47 : Type de variance sous Matlab

Au niveau du code de l'analyse par plateau, ci-contre, une condition récupère le choix utilisateur « VarType ». Si la valeur est de 1, c'est une analyse par le maximum de la deuxième dérivée qui est effectuée, voir chapitre 8.8.2. Un index « v » permet de parcourir le tableau de variance « TabVar1 » ici pour l'axe 1. La variable « okout » permet de sortir de la boucle de recherche. En incrémentant l'index, dès qu'une valeur dans le tableau dépasse la variance plateau « VarPlat », la variance précédente de l'index est

```
%Variance critique avec la méthode de plateau
if VarType == 2 || VarType ==3,
    v=1;
    okout=0;
    %Boucle de recherche de variance maximum
    while okout == 0
        if TabVar1(v) > VarPlat,
            AccelCritAxe1 = TabAccell(v-1);
            VarCritAxe1 = TabVar1(v-1);
            okout = 1;
        end;
        %Si la variance maximum n'a pas
        %atteint la valeur plateau
        if v == NbStep && okout == 0;
            AccelCritAxe1 = TabAccell(v);
            VarCritAxe1 = TabVar1(v);
            okout = 1;
        end;
        v = v+1;
    end;
end;
```

Code 6 : Variance plateau sous Matlab

enregistrée comme valeur critique ainsi que l'accélération qui en est associée. Si la

variance plateau n'est pas atteinte, c'est la valeur de fin de tableau « TabVar » qui est prise en compte. Cela signifie que l'essai devrait être reconduit avec une valeur d'accélération finale plus élevée afin de dépasser cette variance critique. Le graphique ci-dessous représente théoriquement le principe de la variance plateau

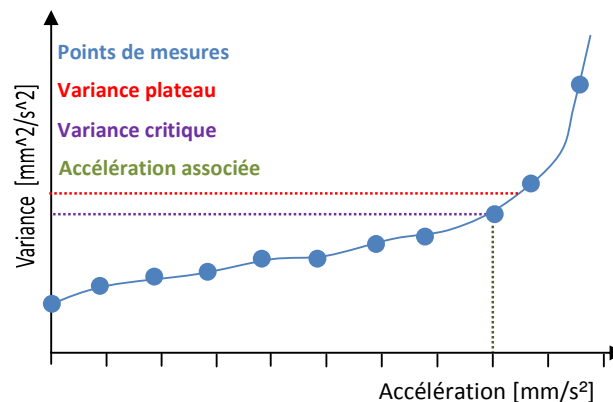


Figure 48 : Analyse plateau théorique

8.11.4 Traitement de l'acquisition

Après que l'acquisition soit effectuée selon les préférences de l'utilisateur, un menu permettant de traiter l'essai s'affiche. Si ce dernier ne contient pas d'analyse de variance, le menu affiche uniquement les coordonnées de départ et d'arrivée du mouvement ainsi que son accélération et sa vitesse. L'utilisateur a la possibilité de sauvegarder cet essai. Si c'est un essai de variance, les paramètres suivant sont affichés :

- ◆ Code de trajectoire.
- ◆ Code de début de trajectoire.
- ◆ Coordonnées de départ.
- ◆ Code de fin de trajectoire.
- ◆ Coordonnées d'arrivée.

- ◆ Axe critique déterminée automatiquement selon le type d'analyse de la variance.
- ◆ Accélération critique déterminée automatiquement selon le type d'analyse de la variance.
- ◆ Variance critique déterminée automatiquement selon le type d'analyse de la variance.
- ◆ Accélération critique de l'axe 1 selon le type d'analyse de la variance.
- ◆ Accélération critique de l'axe 2 selon le type d'analyse de la variance.
- ◆ Accélération critique de l'axe 3 selon le type d'analyse de la variance.
- ◆ Accélération critique totale par la méthode de la deuxième dérivée.
- ◆ Accélération critique totale par la méthode de plateau.
- ◆ Variance critique de l'axe 1 selon le type d'analyse de la variance.
- ◆ Variance critique de l'axe 2 selon le type d'analyse de la variance.
- ◆ Variance critique de l'axe 3 selon le type d'analyse de la variance.
- ◆ Variance critique totale par la méthode de la deuxième dérivée.
- ◆ Variance critique totale par la méthode de plateau.

La courbe de variance totale affiche toujours l'analyse de deuxième dérivée et plateau. Mais lorsque cette dernière n'a pas été sélectionnée par l'utilisateur, la valeur de l'analyse est affichée à « 0 »

```
NOUVELLE ACQUISITION TERMINEE
Informations sur le mouvement
Code de trajectoire = 231
Code de début de trajectoire = D0
Coordonnées START : X=90 Y=0 Z=-350
Code de fin de trajectoire = E0
Coordonnées END : X=120 Y=0 Z=-350
Criticité automatique
Axe Crit auto = 2
Accel Crit auto = 90000[mm/s2]
Variance Auto = 65096 [mm2/s2]
Paramètres généraux
Accel Crit Axe1 = 90000 [mm/s2]
Accel Crit Axe2 = 90000 [mm/s2]
Accel Crit Axe3 = 90000 [mm/s2]
Accel Crit Tot 2ème dérivée = 90000 [mm/s2]
Accel Crit Tot Plateau = 0 [mm/s2]
Var Crit Axe1 = 52109 [mm2/s2]
Var Crit Axe2 = 65096 [mm2/s2]
Var Crit Axe3 = 50667 [mm2/s2]
Var Crit Tot 2ème dérivée = 167872 [mm2/s2]
Var Crit Tot Plateau = 0 [mm2/s2]
0 : Revenir au menu principal
1 : Effectuer une nouvelle acquisition
Test unique
2 : Sauvegarder 1 acquisition AVEC l'accél. critique auto
3 : Sauvegarder 1 acquisition SANS l'accél. critique auto
Serie de test répertoriée
4 : Sauvegarder 1 acquisition AVEC l'accél. critique auto
5 : Sauvegarder 1 acquisition SANS l'accél. critique auto
votre reponse SVP :
```

Figure 49 : Traitement de l'acquisition sous Matlab

8.11.5 Sauvegarde

L'utilisateur peut ensuite sauvegarder ou non son acquisition. Le but de la sauvegarde est de copier et renommer les fichiers de données « DATA.INT », « INFOTIME.INT » et « INFOPARAM .INT » dans un dossier spécifié afin de permettre un traitement ultérieur. Car ceux-ci sont stockés dans le fichier de partage entre le PC Îlot et le PC Superviseur et sont systématiquement écrasés lors d'un nouvel essai du robot. A chaque sauvegarde, est associé un fichier texte qui enregistre toutes les données de l'essai ainsi que les informations qui en ont été extraites lors de l'acquisition. Toutes les sauvegardes sont effectuées dans le fichier « D:\Robot_Delta_INT_History \ ». L'utilisateur peut ensuite affiner la sauvegarde comme suit :

- ◆ S'il s'agit d'un essai unique, un dossier est créé avec comme nom, la date de l'essai. Les fichiers de données y sont intégrés et aussi renommés de la même manière. Par exemple, pour un essai effectué le 5 juillet 2010 à 9h42 et 32 secondes. Le dossier portera comme nom « 2010-7-5-9h42m32s » et à l'intérieur de celui-ci, les fichiers de données et celui de texte de la même manière comme illustre le schéma suivant.

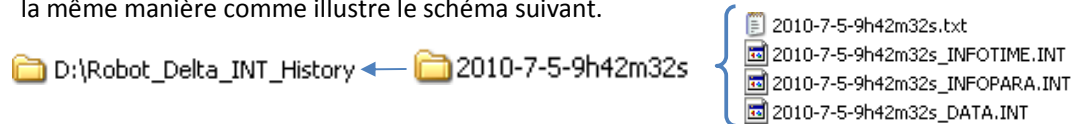


Figure 50 : Principe de sauvegarde d'un essai unique

- ◆ S'il s'agit d'une série de cartographie, l'utilisateur doit entrer le numéro de la série cartographique. Un dossier est alors créé avec ce nombre et les fichiers sont renommés avec le numéro de code du mouvement. La figure suivante illustre un exemple avec les essais 213 et 214 de la série 10.

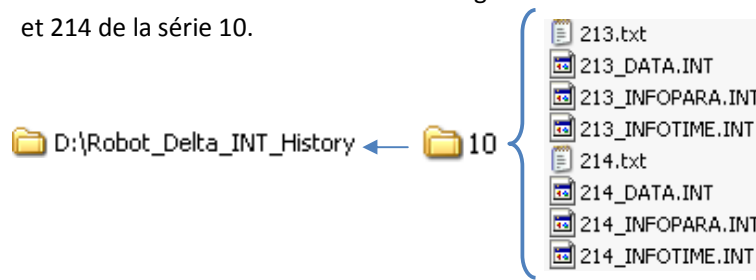


Figure 51 : Principe de la sauvegarde des essais d'une série cartographique

8.11.6 Fichier TXT

Le fichier texte regroupe automatiquement les informations de l'essai ainsi que les données de l'analyse de la variance. Si l'analyse automatique de la variance ne correspond pas avec le graphique ou si les valeurs sont faussées, l'utilisateur peut décider de sauvegarder malgré tout avec les données de l'analyse automatique ou sans celles-ci. Si le second cas est choisi, des menus s'affichent questionnant l'utilisateur sur :

- ◆ L'accélération critique
- ◆ L'axe critique
- ◆ La variance critique

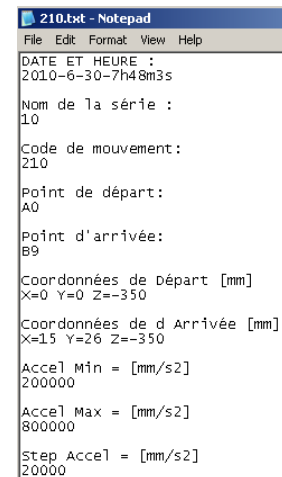


Figure 52 : Début d'un fichier TXT

Les autres paramètres, concernant individuellement chacun des axes, prennent d'office les valeurs du traitement automatique. Un exemple complet de ce fichier TXT est disponible en annexe 7 ou sur le CD.

8.11.7 Retraitement d'acquisition

La particularité du programme Delta sous Matlab est la possibilité de retraiter une ancienne acquisition. En choisissant cette fonction sur le menu principal, l'utilisateur doit tout d'abord préciser s'il recherche :

- ◆ Un essai unique sans analyse de la variance.
- ◆ Un essai unique avec analyse de la variance.
- ◆ Un essai issu d'une série cartographique (toujours réalisé avec une analyse de la variance).

Le programme affiche ensuite une représentation du répertoire « D:\Robot_Delta_INT_History \ » avec sa contenance. Attention, l'affichage ne fait pas de différence entre les fichiers et les dossiers.

- ◆ S'il s'agit d'un essai unique, l'utilisateur entre le nom du fichier correspondant, comme indiqué ci-contre, par exemple : « 2010-7-5-9h42m32s ». Si le fichier est trouvé, un message pop-up apparaît et affiche le contenu du fichier TXT associé à l'essai, et la suite se déroule similairement à une nouvelle acquisition. Si le fichier est introuvable, un message d'erreur l'indique et le programme retourne au menu principal.

```
Lecture d'un modèle sauvé au préalable :
Liste des modèles trouvés dans le répertoire de travail :

.          2010-6-25-17h50m57s  4
..         2010-6-26-13h6m12s  40
1          2010-7-1-18h6m54s   50
10         2010-7-5-9h42m32s   60
11         30
2010-6-25-13h48m14s  31
test

fx Nom de la mesure ? :
```

Figure 53 : Recherche d'un ancien essai

- ◆ S'il s'agit d'un essai de série cartographique, le menu identique s'affiche comme ci-dessus en demandant à l'utilisateur le nom de la série, par exemple « 10 ». Puis le programme affiche les fichiers continus dans le dossier de la série. L'utilisateur entre ensuite le numéro de code de l'essai, par exemple « 213 ». Si le fichier est trouvé, un message pop-up apparaît et affiche le contenu du fichier .TXT associé à l'essai et la suite se déroule similairement à une nouvelle acquisition. Si le dossier et le fichier sont introuvables, un message d'erreur l'indique et le programme retourne au menu principal.

8.11.8 Première série d'essais : Série 1

L'acquisition sous Matlab étant fonctionnel, une première série d'essais en vue d'une cartographie a alors été réalisée. C'est une série radiale, donc avec 64 points de mesure, avec le robot, à vide à une hauteur Z = -350 [mm] avec une analyse de la variance du maximum de la deuxième dérivée. Cette série porte la dénomination « 1 », elle est disponible en annexe sur le CD.

Remarque :

La réalisation d'une série prend beaucoup de temps, environs 4 heures pour cette première série. Il faut exécuter les mouvements du robot qui dure environs 30 secondes, effectuer l'acquisition sous Matlab (10 secondes), interpréter personnellement la courbe de la variance (10 secondes à 1 minute), sauvegarder celle-ci (2 à 20 secondes) et finalement lancer le mouvement suivant. A noter que certains points sont à réaliser plusieurs fois afin de trouver les bons paramètres d'accélération. Le temps approximatif varie en fonction de l'exactitude de l'analyse automatique de la variance.

8.12 Conception de la cartographie 3D sous Matlab

Dans l'optique de représenter l'accélération critique selon la position du robot, une cartographie en 3D a été implémentée. Celle-ci représente le plan de transition sur l'échelle cartésienne du robot en X [mm] et en Y [mm]. Sur l'axe Z est représenté les valeurs du paramètre à analyser comme par exemple l'accélération critique en [mm²/s²].

8.12.1 Script de représentation polaire en 3D

La première étape a été d'étudier la manière dont un graphique 3D est conçu sous Matlab. Ce dernier permet de réaliser facilement une multitude de représentations. Depuis sa librairie de base, il est possible de créer des graphiques en 3D mais uniquement à partir d'une base carrée alors que le plan de transition est un disque. Une longue recherche sur la base de données communautaire de Matlab¹ a permis de trouver plusieurs scripts réalisant le type de cartographie recherché. Après l'essai de plusieurs fichiers, seul le programme « polarplot3d.m »² convenait aux besoins. Les différents scripts testés se trouvent sur le CD en annexe.

Ci-dessous, à gauche, ce que la librairie de Matlab peut réaliser avec la fonction « surfn ». Au centre et à droite, ce que le script « polarplot3d.m » peut concevoir. Ce dernier a la particularité de reproduire qu'une partie du disque, ce qui va permettre de créer la cartographie secteur par secteur.

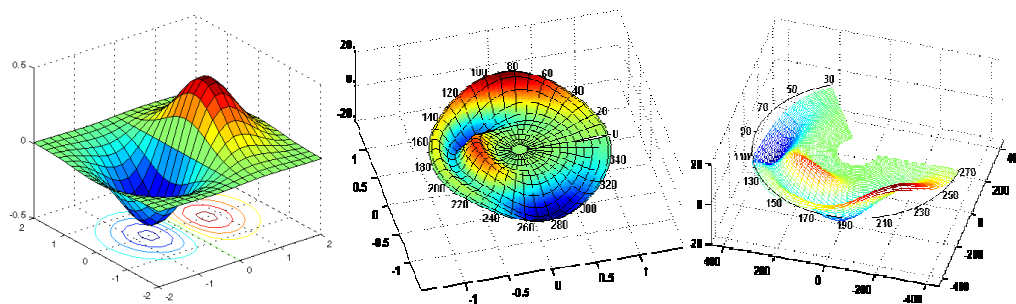


Figure 54 : Graphique Matlab de base et polarplot3d.m

¹ www.mathworks.com/matlabcentral/fileexchange/

² Conçu par Ken Garrard, North Carolina State University, sous licence libre BSD
<http://www.mathworks.com/matlabcentral/fileexchange/13200-3d-polar-plot>

Remarque 1 :

L'intersection des lignes noires correspond à un point de mesure.

Remarque 2 :

Le script « polarplot3d.m » fait appel aux scripts « parse_pv_pairs.m »² et « rotatePoints.m »², du même auteur.

8.12.2 Script de recherche d'une variable dans un fichier

La seconde étape était de récupérer les paramètres à cartographier sachant que ceux-ci sont stockés dans un fichier .TXT. Ce simple procédé n'est pas réalisable avec les fonctions de la librairie initiale de Matlab et aucun fichier sur la base de données communautaire ne correspondait à la recherche. Un script a alors été développé, « SearchParam.m », en annexe 7 et sur le CD. Il a été découvert que la fonction de lecture d'une ligne sous Matlab « fgetl », sautait d'une ligne à l'autre si elle était simplement intégrée dans une boucle, sans paramètre supplémentaire. Cette fonctionnalité a alors été utilisée afin de comparer chaque ligne avec le paramètre recherché dans le fichier TXT, par exemple « AXE CRITIQUE = », et lors de l'incrément suivant, la valeur du paramètre est récupéré. Ce script a ensuite été intégré à la programmation de la cartographie.

8.12.3 Mise au point d'un secteur de cartographie

Un point n'a pas été abordé jusqu'à maintenant : Comment fixer les valeurs issues d'une mesure sur la cartographie ? Car un mouvement d'essai est effectué entre deux points distancés d'environ 30[mm]. Idéalement, le point de mesure devrait être défini entre ces deux points. Mais afin de faciliter la conception de la cartographie, il a été décidé d'attribuer le point de mesure sur point de départ du mouvement. C'est-à-dire que les points de mesures sont en réalité décalés vers le centre d'environ 15[mm] pour les trajectoires radiales, et de 15[mm] vers l'axe du robot le plus proche pour les trajectoires tangentielles. Ce dernier phénomène résulte de la reconstitution des secteurs, voir chapitre 8.5. Vu que l'on procède de cette manière, le diamètre du plan de transition qui sera cartographié passe à 540[mm] et de ce fait, les trajectoires tangentielles sur la dernière ligne de rayon, « K » sur le graphique ci contre, ne seront pas réalisées.

Le script « polarplot3d.m » nécessite un tableau du même nombre de colonne que de ligne, sinon il réalise automatiquement une interpolation entre les points de mesures, ce qui n'est pas optimal. Vu que l'on dispose de 10 rayons de mesures en rapport les points mesurés sur le début de la trajectoire, il a été décidé de diviser un secteur par 10 sur 10, donc 100 points de cartographie, comme le démontre la figure ci-contre.

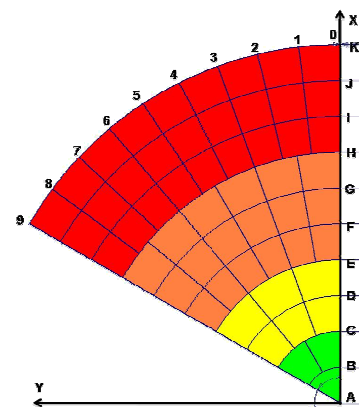
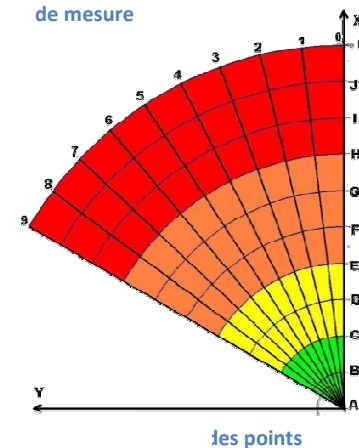


Figure 55 : Secteur des points de mesure



les points

Ne disposant que de 64 points de mesures, et non 100, ces derniers ont été organisés dans le tableau à fournir au script « polarplot3d ». Ce tableau se nomme « Tabsector ». Le tableau d'adaptation de 64 à 100 points est disponible en annexe dans le fichier « plan.xlsx » sur le CD. C'est le script « SectorTabRad64.m » qui se charge d'organiser le tableau.

8.12.4 Création secteur par secteur

Si l'on voulait créer d'un seul coup la cartographie complète sur 360 degrés, un tableau de 100 par 100 aurait été nécessaire. Afin de simplifier ceci, il a été décidé de reconstituer l'ensemble du plan de transition, secteur par secteur de 60°, avec les tableaux de 10 sur 10. C'est le script « Carto.m » qui se charge de représenter graphiquement les secteurs à partir des tableaux fournis par « SectorTabRad64.m ».

En représentant l'accélération du secteur de la série « 1 » mesuré en 3D, on obtient finalement la figure ci-contre.

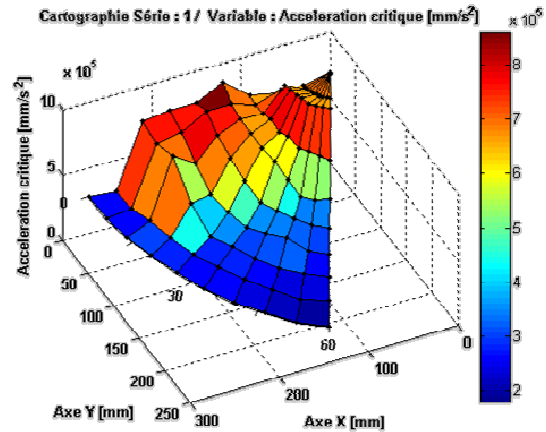


Figure 57 : Surface de la Série 1

8.13 Première cartographie : Série 1

Série N°	Trajectoire	Type d'analyse de variance	Poids [kg]	Z [mm]	Résolution
1	Radiale	2 ^{ème} dérivée	A vide	-350	6*64

La réalisation d'un graphique en 3D étant au point, la cartographie de l'accélération critique de la série « 1 » donne le graphique ci-dessous. Pour rappel, la série « 1 » est un essai sans poids supplémentaire, donc à vide, à une hauteur Z=-350[mm], et déterminée avec l'analyse de variance du maximum de la deuxième dérivée.

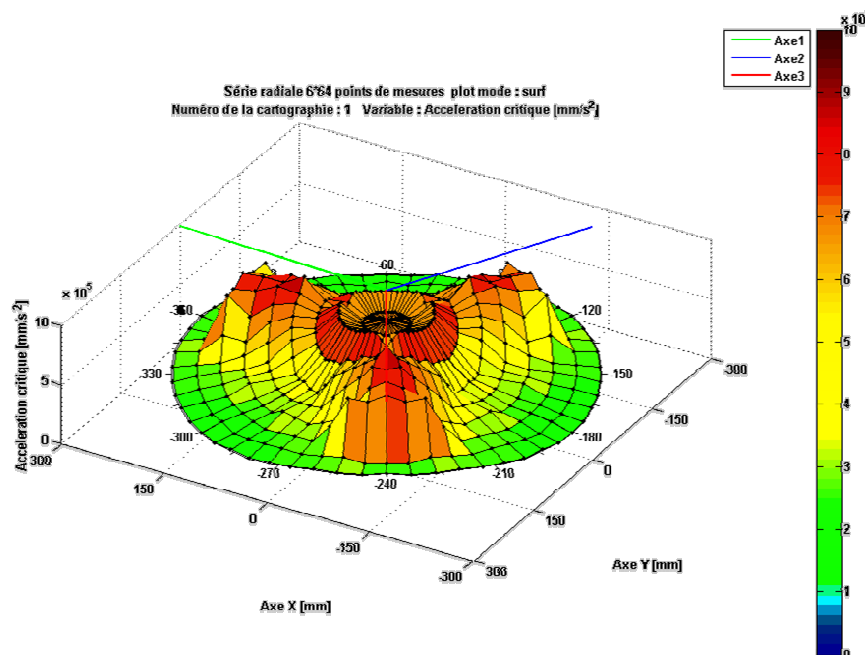


Figure 58 : Cartographie de l'accélération critique de la Série 1

Les axes du robot sont représentés par les trois lignes colorées. L'on peut aussi extraire la variance associée à ces accélérations critiques.

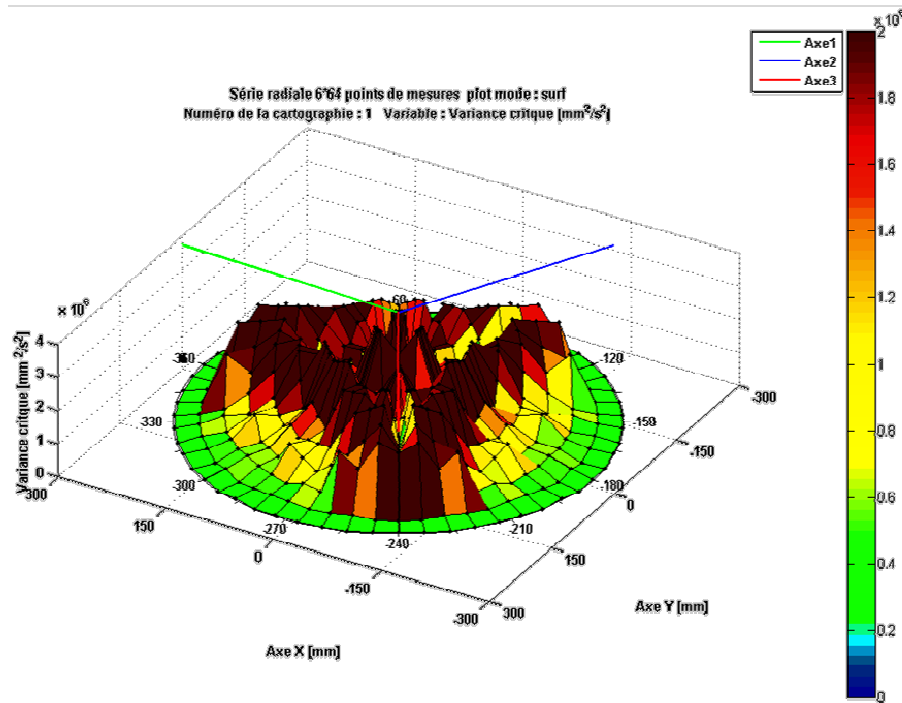


Figure 59 : Cartographie de la variance critique de la Série 1

Premières constatations de cette série « 1 », radiale, à vide, Z = -350[mm] :

- ◆ Le procédé de cartographie semble bien fonctionner
- ◆ Les accélérations du robot sont élevées, on dépasse les 800'000[mm/s²] à certains endroits ce qui correspond à 80[G].
- ◆ Pour les mouvements radiaux, les accélérations sont beaucoup plus élevées le long des axes du robot.
- ◆ Comme prévu, la méthode de la deuxième dérivée a pour conséquence des différences de variances élevées.
- ◆ Au centre du plan de transition, il subsiste une sorte de cratère, ce qui paraît étonnant.

Suite à la dernière constatation, il a été décidé d'augmenter le nombre de points de mesures au centre du plan de transition.

Remarque :

La notion de [G] est utilisée afin de simplifier la lecture des valeurs d'accélérations. 1[G] est une unité relative à l'accélération correspondant l'attraction terrestre, donc sa pesanteur, c'est-à-dire à 9,8 [m/s²]. Cette valeur a été reprise et fixée à 10 [m/s²] dans ce rapport pour faciliter la conversion des [mm/s²] en [G]

8.14 Augmentation des points de mesures

En effet, si l'on se réfère au secteur de cartographie, voir chapitre 8.7.1, et à la table Excel de remplissage du tableau de cartographie, en annexe 8, il y a peu de points de mesures au centre du plan de travail. Pour remplir les dix cases d'une ligne du tableau lorsqu'il y a moins de dix mesures, on procède à des moyennes entre les valeurs, ce qui n'est pas vraiment représentatif. Augmenter le nombre de points de mesures radiales de 64 à 100 permettrait de palier à ce problème et d'observer plus précisément ce qui se passe au centre. Les séries tangentielles n'ont pas été concernées par ce changement car la surface est déjà suffisamment quadrillée de trajectoires d'essais et son tableau de remplissage est aussi disponible en annexe 8.

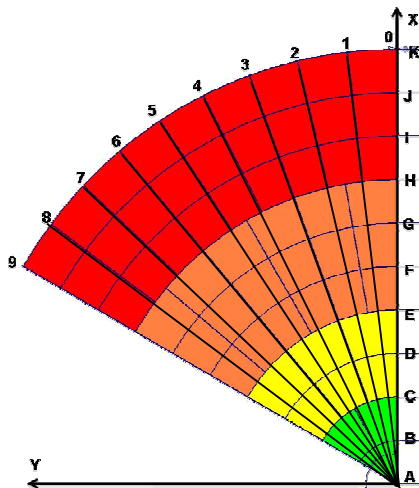
8.15 Restructuration des séries d'essais

Avec désormais 100 points de mesures, une nouvelle série de codes a été instaurée afin répertorier les trajectoires de tests radiales.

Série cartographique radiale 100 points														
Série 1			Série 2			Série 3			Série 4			Série 5		
Code	Start	End	Code	Start	End	Code	Start	End	Code	Start	End	Code	Start	End
201	A0	B0	211	B0	C0	221	C0	D0	231	D0	E0	241	E0	F0
202	A0	B1	212	B1	C1	222	C1	D1	232	D1	E1	242	E1	F1
203	A0	B2	213	B2	C2	223	C2	D2	233	D2	E2	243	E2	F2
204	A0	B3	214	B3	C3	224	C3	D3	234	D3	E3	244	E3	F3
205	A0	B4	215	B4	C4	225	C4	D4	235	D4	E4	245	E4	F4
206	A0	B5	216	B5	C5	226	C5	D5	236	D5	E5	246	E5	F5
207	A0	B6	217	B6	C6	227	C6	D6	237	D6	E6	247	E6	F6
208	A0	B7	218	B7	C7	228	C7	D7	238	D7	E7	248	E7	F7
209	A0	B8	219	B8	C8	229	C8	D8	239	D8	E8	249	E8	F8
210	A0	B9	220	B9	C9	230	C9	D9	240	D9	E9	250	E9	F9
Série 6			Série 7			Série 8			Série 9			Série 10		
Code	Start	End	Code	Start	End	Code	Start	End	Code	Start	End	Code	Start	End
251	F0	G0	261	G0	H0	271	H0	I0	281	I0	J0	291	J0	K0
252	F1	G1	262	G1	H1	272	H1	I1	282	I1	J1	292	J1	K1
253	F2	G2	263	G2	H2	273	H2	I2	283	I2	J2	293	J2	K2
254	F3	G3	264	G3	H3	274	H3	I3	284	I3	J3	294	J3	K3
255	F4	G4	265	G4	H4	275	H4	I4	285	I4	J4	295	J4	K4
256	F5	G5	266	G5	H5	276	H5	I5	286	I5	J5	296	J5	K5
257	F6	G6	267	G6	H6	277	H6	I6	287	I6	J6	297	J6	K6
258	F7	G7	268	G7	H7	278	H7	I7	288	I7	J7	298	J7	K7
259	F8	G8	269	G8	H8	279	H8	I8	289	I8	J8	299	J8	K8
260	F9	G9	270	G9	H9	280	H9	I9	290	I9	J9	300	J9	K9

Tableau 5 : Tableau des codes radiaux 100 points

Les programmes sous Matlab ont été adaptés et c'est désormais le script « SectorTabRad100.m » qui remplit le tableau de secteur pour la réalisation de la cartographie. Toutes les mesures sont stockées dans une case du tableau, il n'y a donc plus besoin de faire des interpolations pour remplir les 100 cases du tableau. Le script du remplissage de tableau avec 64 points de mesures, « SectorTabRad64.m » a été conservé et peut toujours représenter graphiquement la série de cartographie « 1 ». Le script « Carto.m » détecte automatiquement s'il s'agit d'une série radiale de 64 points, radiale de 100 points ou tangentielle de 45 points.



Pour les essais tangentiels, c'est le script « SectorTabTan45.m » qui remplit le tableau du secteur de cartographie. Ne possédant pas 100 points de mesures, des interpolations sous forme de moyennes ont été réalisées. La manière de compléter la table suivante, qui représente le remplissage d'un secteur avec le codage des mouvements, a été obtenu en superposant le graphique illustrant les points de mesures du secteur à celui divisé par 10 sur 10 comme illustré ci-contre. La dénomination « moy » signifie la moyenne des valeurs adjacentes.

9	8	7	6	5	4	3	2	1	0	
301	301	301	301	301	301	301	301	301	301	A
301	301	301	301	301	301	301	301	301	301	B
303	303	303	moy	moy	moy	302	302	302	302	C
306	306	306	305	305	305	305	304	304	304	D
309	309	309	308	308	308	308	307	307	307	E
315	315	314	moy	313	312	moy	311	310	310	F
321	321	320	moy	319	318	moy	317	316	316	G
327	327	326	moy	325	324	moy	323	322	322	H
336	335	334	333	332	332	331	330	329	328	I
345	344	343	342	341	341	340	339	338	337	J

Tableau 6 : Remplissage du tableau "SectorTab" des essais tangentiels

8.16 Révision du programme d'acquisition « EX_10.PAS »

Jusqu'à maintenant, les cycles de mouvements du robot étaient implémentés dans un programme « EX_10.PAS ». Avec le passage à 100 points de mesures, un message d'erreur apparaissait indiquant que le nombre de variable maximale avait été dépassé. Suite à cette problématique de mémoire sous DOS, la solution a été de diviser le programme en 3 parties :

1. « CARTRA1.PAS » V 1.2 : Pour les séries à trajectoires radiales de 201 à 250.
2. « CARTRA2.PAS » V 1.2 : Pour les séries à trajectoires radiales de 251 à 300.
3. « CARTTAN.PAS » V 1.2 : Pour les séries à trajectoires tangentiels de 301 à 345.

Les 3 programmes ont la possibilité de réaliser des cycles de mouvements avec paramétrage manuel des coordonnées de départ et d'arrivée. Cette révision du programme d'acquisition a été l'occasion d'alléger le code et de faire des appels de fonctions pour des commandes qui prenaient beaucoup de lignes de codes et qui se retrouvaient dans le code. Ces procédures sont :

- ◆ EmptyTab : ligne 1561 dans « CARTRA1.PAS »
Vider les tableaux contenant les mesures
- ◆ WriteParam : ligne 1633 dans « CARTRA1.PAS »
Ecrire les paramètres dans « INFOTIME.INT » et « INFOPARAM.INT »
- ◆ AssignSerie : ligne 1676 dans « CARTRA1.PAS »
Assigner les coordonnées et le nom du mouvement en fonction du code de trajectoire.
- ◆ ReadCoordPoint : ligne 2079 dans « CARTRA.PAS »
Lire les coordonnées des points du mouvement.

Les programmes en Turbopascal se trouvent en annexe sur le CD et les parties importantes du code en annexe 3. Cette dernière est issue de « CARTRA1.PAS ». Les programmes sont relativement similaires exceptés les variables de locations de points et la procédure « AssignSerie ».

8.17 Programme final de représentation cartographique

Le script « Carto.m » se charge d'effectuer le graphique en 3D. L'utilisateur entre le numéro de la série et choisit ensuite le paramètre à extraire de la série. A noter que la représentation des valeurs critiques propres à chaque axe n'est pas totalement fonctionnelle car elle cartographie le même secteur sur tout le plan de transition, alors que ce dernier doit être associé aux valeurs critiques des autres axes pour être juste. L'utilisateur a ensuite la possibilité de choisir entre 4 types de représentations graphiques.

```

PLOT DES CARTOGRAPHIES
Informations de la série
0 Retour au menu principal
1 Plot Accel Critique Déduite
2 Plot Variance Critique Déduite
3 Plot Axe Critique Déduit

4 Plot Accel Critique Axe1 (Non fonctionnel)
5 Plot Accel Critique Axe2 (Non fonctionnel)
6 Plot Accel Critique Axe3 (Non fonctionnel)
7 Plot Accel Critique Totale en 2ème Dérivée
8 Plot Accel Critique Totale en Plateau
9 Plot Variance Critique Axe1 (Non fonctionnel)
10 Plot Variance Critique Axe2 (Non fonctionnel)
11 Plot Variance Critique Axe3 (Non fonctionnel)
12 Plot Variance Critique Totale en 2ème Dérivée
13 Plot Variance Critique Totale en Plateau

votre reponse SVP : |
    
```

[Je à Cartographier](#)

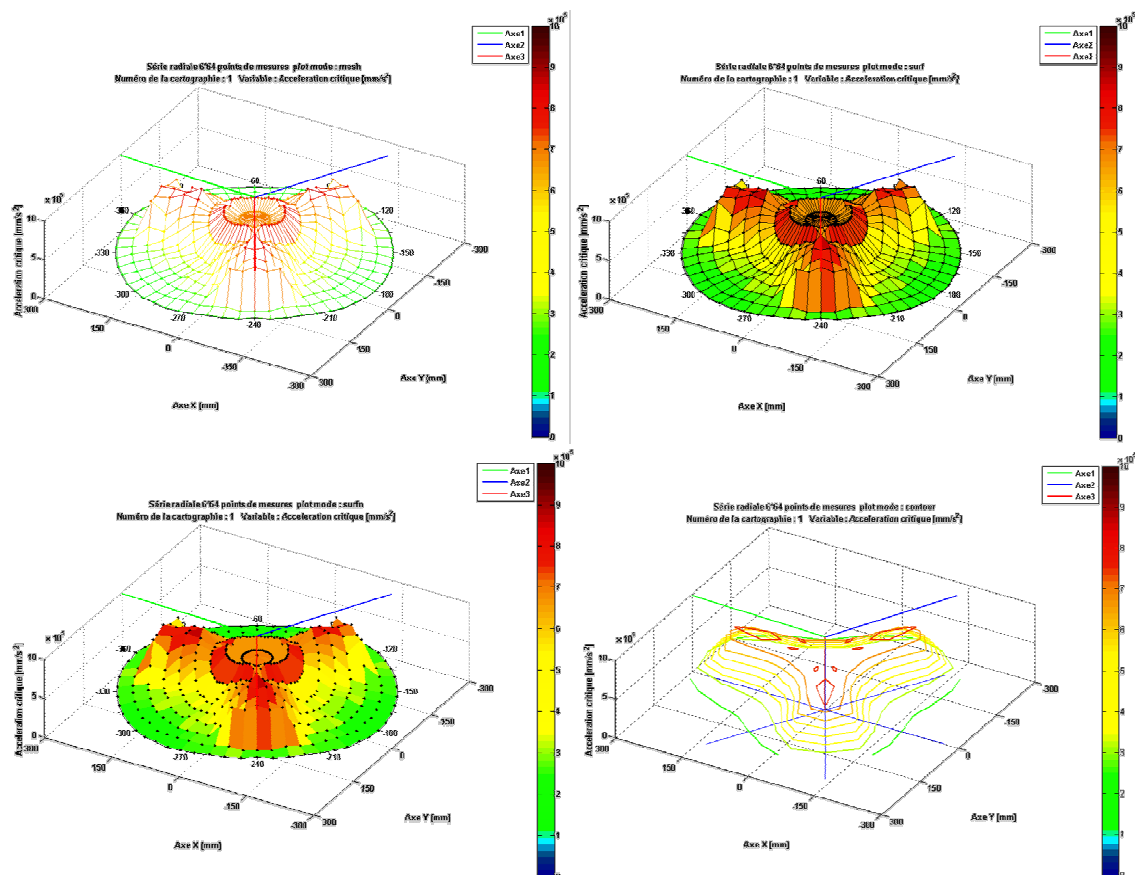


Figure 62 : Types de représentation graphique 3D : Mesh, Surf, Surf et Contour

8.18 Nouvelles cartographies

Suite à la révision du programme d'acquisition, différentes séries de cartographies on été réalisées avec des paramètres différents, comme le résume le tableau ci-dessous.

Série N°	Trajectoire	Type d'analyse de variance	Poids [kg]	Z [mm]	Résolution
1	Radiale	2 ^{ème} dérivée	A vide	-350	6*64
10	Radiale	2 ^{ème} dérivée	A vide	-350	6*100
11	Tangentielle	2 ^{ème} dérivée	A vide	-350	6*45
30	Radiale	2 ^{ème} dérivée	1	-350	6*100
31	Tangentielle	2 ^{ème} dérivée	1	-350	6*45
40	Radiale	Plateau : 150000 [mm ² /s ²]	1	-350	6*100
50	Radiale	2 ^{ème} dérivée	1	-450	6*100

Tableau 7 : Liste des cartographies

Remarque 1 :

La perspective 3D des représentations ainsi que la représentation des 3 axes moteurs du robot, sont ajustés automatiquement selon la valeur maximale des mesures. Néanmoins, l'échelle de couleur reste la même pour tout les mêmes types de graphiques. Pour les accélérations critiques, cela va de 0 à 10⁵ [mm/s]. Les variances critiques vont de 0 à 2⁶ [mm²/s²].

Remarques 2 :

La notion « A vide » n'est pas totalement exacte. En effet, la nacelle du robot et les avants bras possèdent une masse et une inertie qui influent le comportement du robot. Afin de déterminer le poids à prendre en compte, la nacelle est pesée avec une seule paire d'avant bras. Ce qui donne une masse de 514 grammes. L'atelier mécanique a rapidement réalisé un petit lest d'1 kilogramme pour réaliser les essais avec une masse supplémentaire.



Légendes

1. Avant bras
2. Bras de rotation
3. Nacelle
4. Raccord de tuyau
5. Masse d'1 [kg]
6. Buse

Figure 63 : Nacelle avec masse d'1 [kg]

8.18.1 Série 10

Série N°	Trajectoire	Type d'analyse de variance	Poids [kg]	Z [mm]	Résolution
10	Radiale	2 ^{ème} dérivée	A vide	-350	6*100

Cette série est identique à la première mais avec plus de précisions et une analyse plus pointilleuse par la méthode de la deuxième dérivée. Les accélérations critiques de cette série donnent ceci :

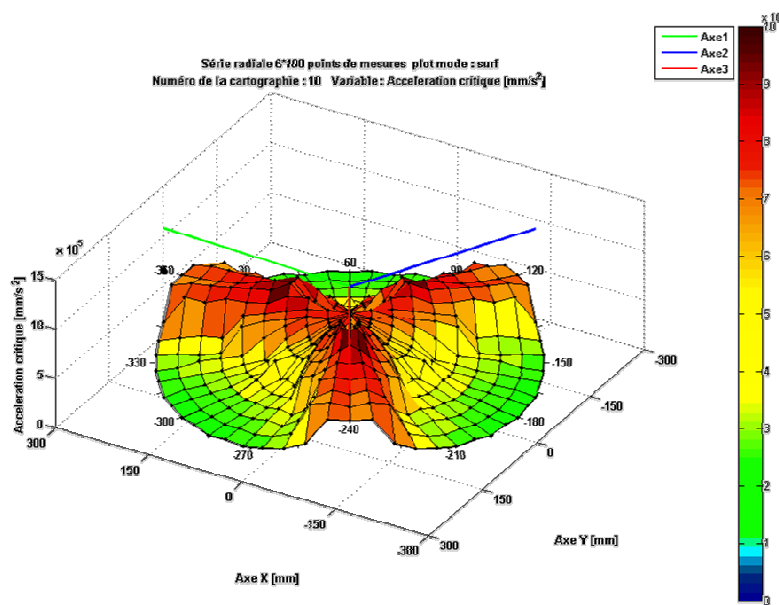


Figure 64 : Cartographie de l'accélération critique de la Série 10

Comparée à la série 1, la zone centrale est bien plus détaillée, l'on constate bel est bien un affaissement en son centre. Les accélérations sur les extrémités des axes sont plus élevées par rapport à la série 1. Les valeurs maximales ne se trouvent pas exactement au centre, mais décalées d'environ 90[mm] dessous les trois axes du robot. Les valeurs maximales dépassent les 90[G] et les minimales, en bordure entre les axes, autour de 20[G]. Les variances associées aux accélérations critiques de la série 10 donnent le graphique suivant :

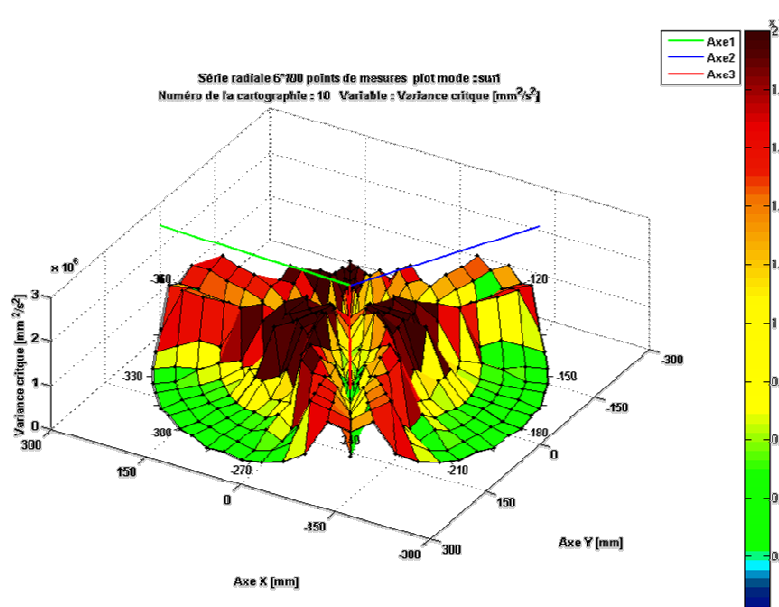


Figure 65 : Cartographie de la variance critique de la Série 10

Les variances sont mieux délimitées que sur la série 1, mais reste similaires. En effet, il y a quasiment un facteur 10 entre les variances minimales et maximales, ce qui n'est pas vraiment optimal si l'on désire une certaine précision dans le mouvement. Les variances qui vont au-delà du jaune, dépasse largement la valeur de $1'000'000[\text{mm}^2/\text{s}^2]$, ce qui correspond à une erreur de vitesse d' $1[\text{m/s}]$. C'est un chiffre relativement élevé, si l'on considère que le plan de transition cartographié représente $540[\text{mm}]$ de diamètre.

Si l'on observe les axes critiques pour les mouvements radiaux de cette série 10, on obtient le graphique suivant :

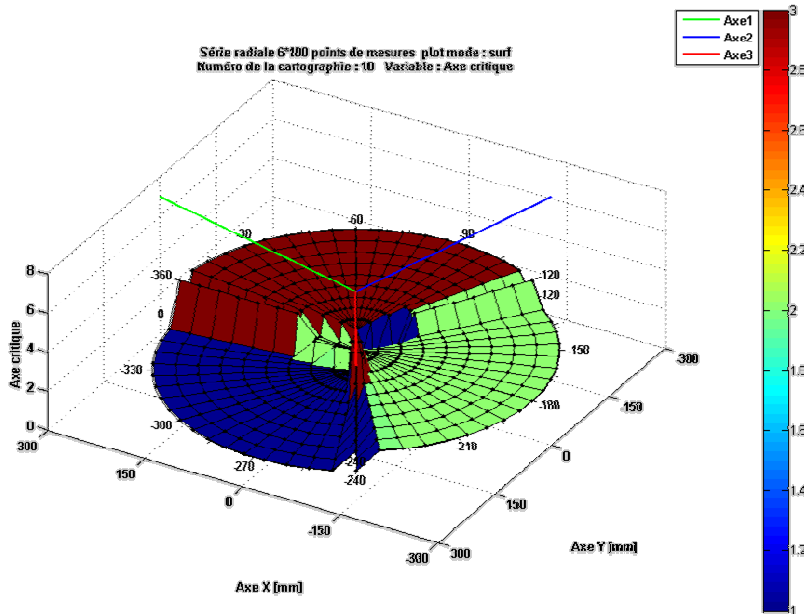


Figure 66 : Cartographie de l'axe critique de la Série 10

Mise à part au centre où le comportement du robot est particulier, l'axe critique est toujours celui qui est opposé à la zone de la trajectoire. Ce phénomène vient du fait que le moteur de l'axe en question se retrouve seul à fournir un effort pour effectuer un mouvement et que les autres axes ne bougent que faiblement, s'il l'on prend le cas extrême. Mais si une trajectoire s'effectue sous un bras d'axe, les deux autres en opposition reprennent à ensemble l'effort à effectuer pour le mouvement.

8.18.2 Série 11

Série N°	Trajectoire	Type d'analyse de variance	Poids [kg]	Z [mm]	Résolution
11	Tangentielle	2 ^{ème} dérivée	A vide	-350	6*45

Cette série cartographique a été réalisée avec des trajectoires tangentielles mais avec les mêmes paramètres de la Série 10. Les accélérations critiques donnent le graphique suivant :

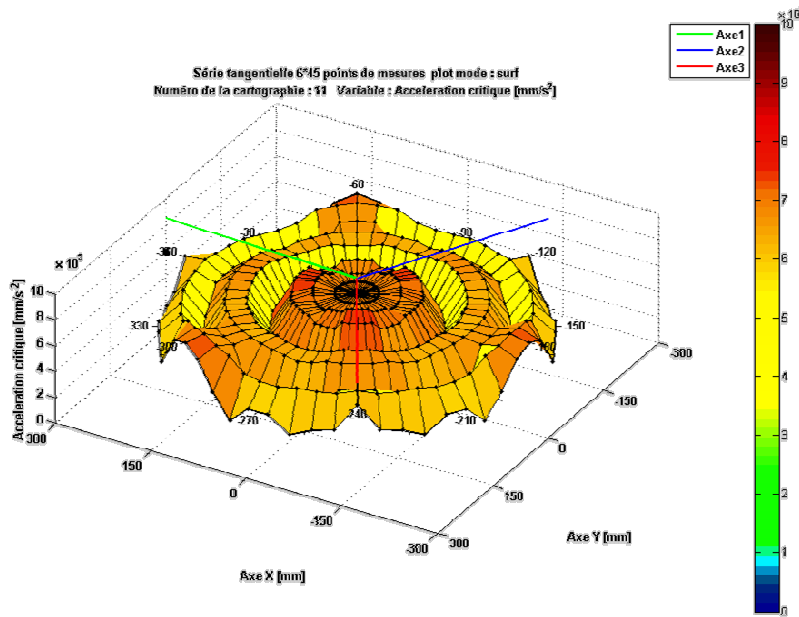


Figure 67 : Cartographie de l'accélération critique de la Série 11

Les valeurs sont beaucoup plus constantes qu'avec une série radiale. On s'aperçoit que des « creux » se forment à une certaine distance du centre. Ce phénomène a tout d'abord été traité comme une erreur de mesure. Les points correspondants ont alors été à nouveau testés avec plus d'attention. Le résultat est resté le même, ce qui pousserait à dire que le robot dispose de certaines faiblesses tangentielles selon le rayon de la trajectoire. Les variances associées donnent le graphique suivant :

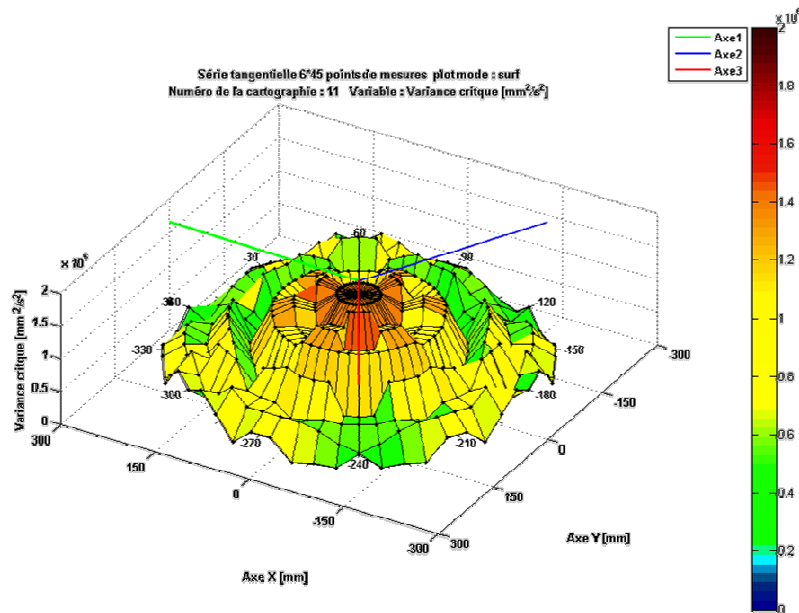


Figure 68 : Cartographie de la variance critique de la Série 11

Les erreurs de variances associées aux accélérations critiques sont beaucoup moins élevées que sur un mouvement radial. Mais entre le maximum et le minimum subsiste un facteur 4 environ.

Le graphique représentant les axes critiques est très intéressant :

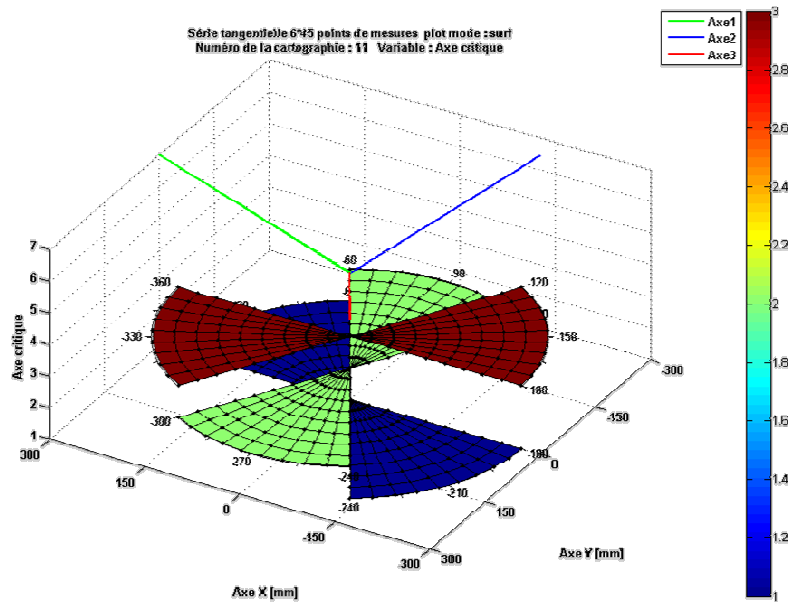


Figure 69 : Cartographie de l'axe critique de la Série 11

L'on constate clairement que ce sont les zones perpendiculaires aux axes moteurs qui sont critiques.

8.18.3 Série 30

Série N°	Trajectoire	Type d'analyse de variance	Poids [kg]	Z [mm]	Résolution
30	Radiale	2 ^{ème} dérivée	1	-350	6*100

Cette série est similaire à la 10 mais avec un poids d'un kilogramme. Les valeurs des accélérations critiques donnent le graphique ci-dessous :

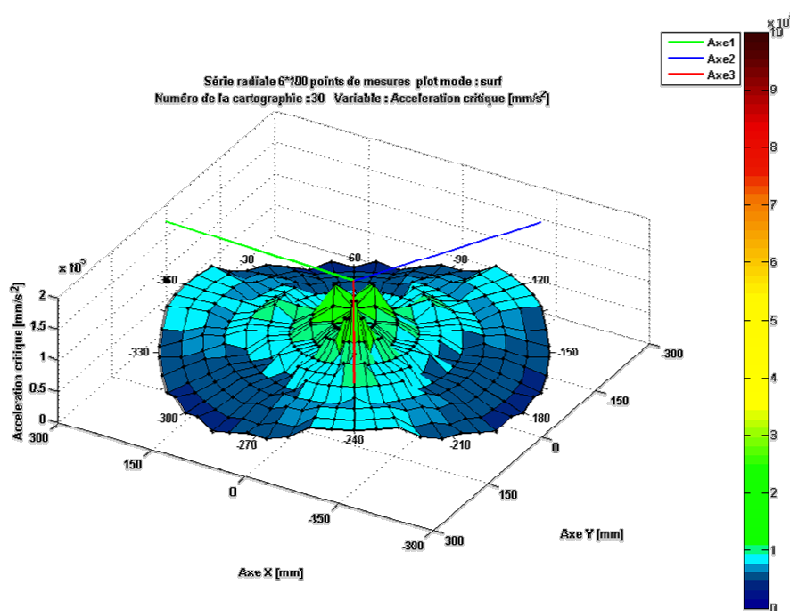


Figure 70 : Cartographie de l'accélération critique de la Série 30

Avec une masse, les accélérations critiques sont sensiblement plus petites. Comparé à la série 10 réalisé dans les même conditions mais à vide, les valeurs sont environs 5 fois plus basses. Il en résulte encore une tendance à ce que les accélérations soient plus optimales sur les trajectoires effectuées en dessous des axes du robot.

Le graphique des variances associées est lui aussi intéressant :

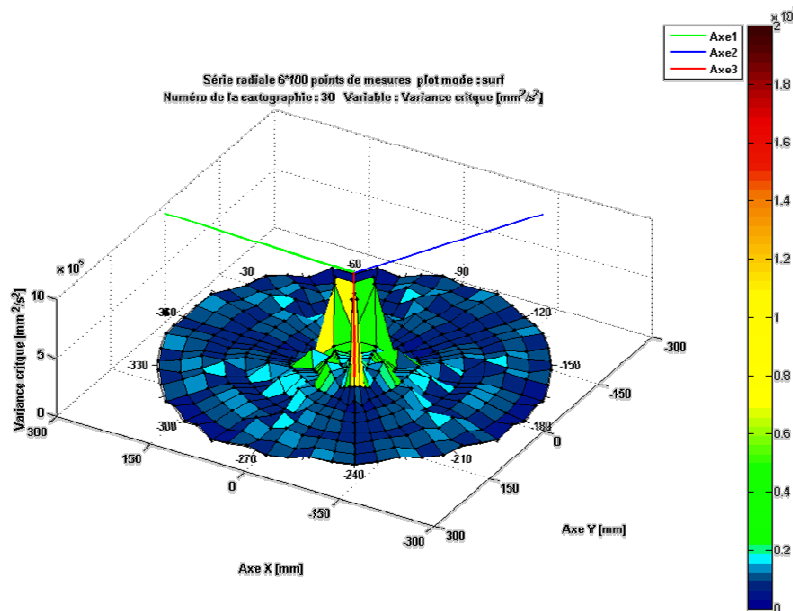


Figure 71 : Cartographie de la variance critique de la Série 30

Mise à part au centre, les valeurs sont très proches. Ce qui signifie que l'analyse du maximum de la deuxième dérivée est alors exploitable même si l'on désire une certaine régularité dans les erreurs de variances. Ce raisonnement est possible si l'on excepte les quelques trajectoires au centre du plan de transition ayant des variances plus élevées.

8.18.4 Série 31

Série N°	Trajectoire	Type d'analyse de variance	Poids [kg]	Z [mm]	Résolution
31	Tangentielle	2 ^{ème} dérivée	1	-350	6*45

Cette série est réalisée avec des trajectoires tangentielles dans les mêmes conditions que la série 30. Les accélérations critiques sont représentées ci-dessous :

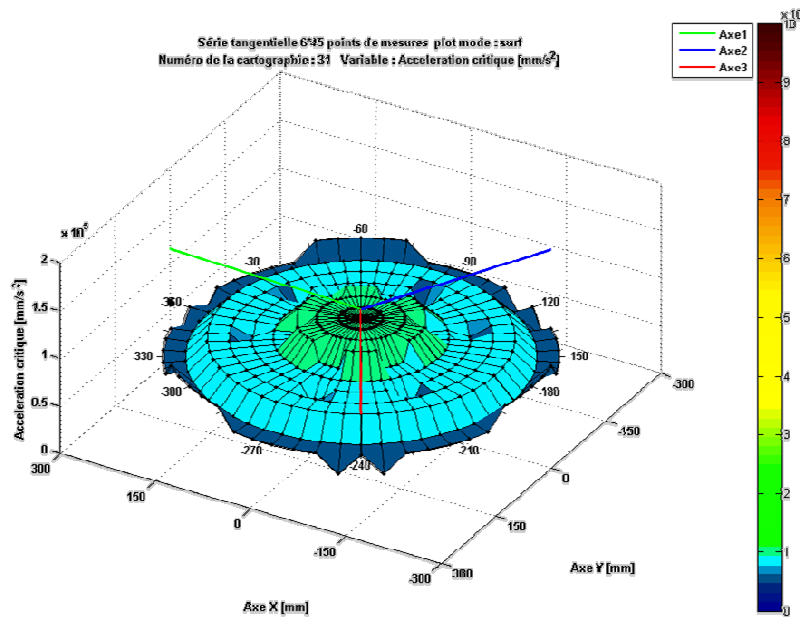


Figure 72 : Cartographie de l'accélération critique de la Série 31

Les valeurs sont relativement proches comparées aux trajectoires radiales et même plus élevées à certains endroits, ce qui est intéressant.

Pour ce qui en est des variances associés, voir le graphique ci-dessous.

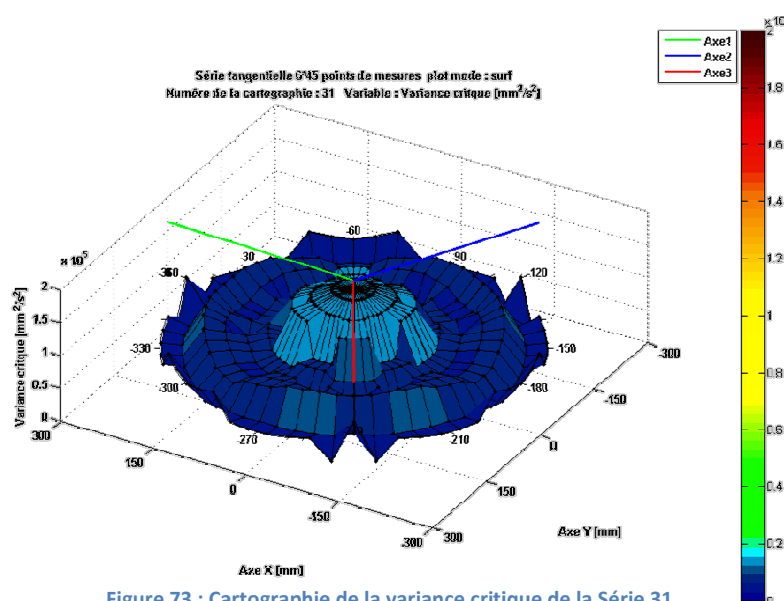


Figure 73 : Cartographie de la variance critique de la Série 31

Celles-ci sont très basses et relativement constantes si l'on se base par rapport à l'échelle.

8.18.5 Série 40

Série N°	Trajectoire	Type d'analyse de variance	Poids [kg]	Z [mm]	Résolution
40	Radiale	Plateau : 150000 [mm ² /s ²]	1	-350	6*100

Cette série a été reprise de la série 3 mais ses données ont été retraitées sous Matlab en faisant cette fois-ci, une analyse de variance avec une valeur de plateau limite. Cette dernière a été fixée à 150'000 [mm²/s²] ce qui correspond à une erreur de vitesse de 0.387 [m/s]. Les accélérations critiques donnent alors ceci :

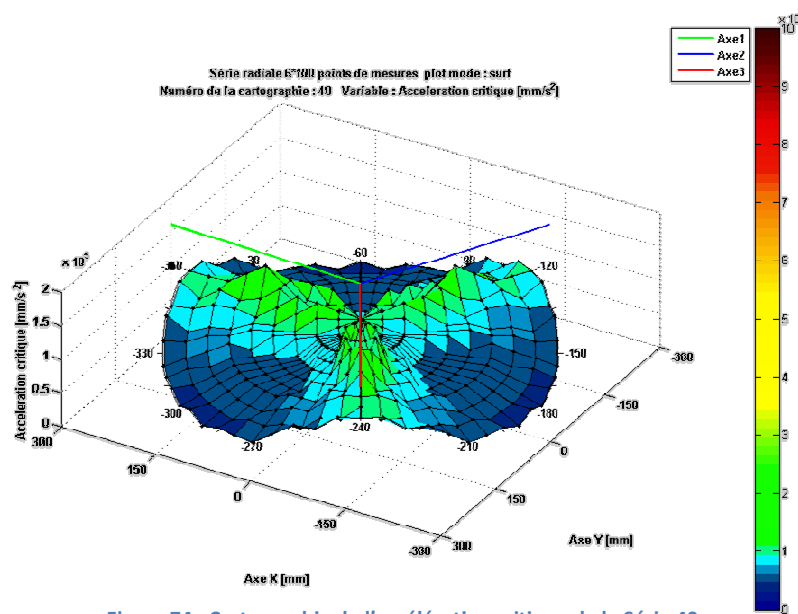


Figure 74 : Cartographie de l'accélération critique de la Série 40

Comparées à la série 30, les accélérations sont plus élevées, spécialement le long des axes. De plus, l'aspect ressemble particulièrement aux accélérations critiques de la série 10.

Les variances critiques sont représentées ci-dessous :

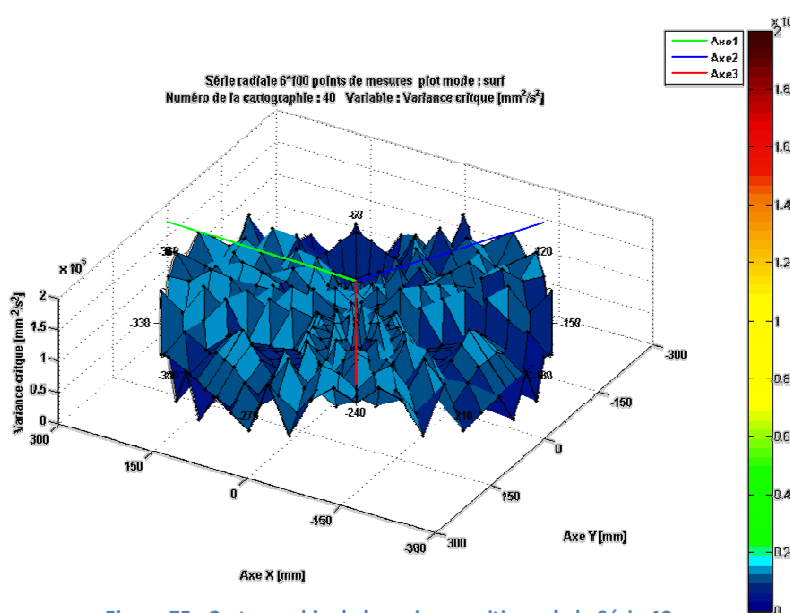


Figure 75 : Cartographie de la variance critique de la Série 40

Celles-ci se situent comme prévu juste en dessous de $150'000[\text{mm}^2/\text{s}^2]$. Les axes critiques ressortent légèrement différemment qu'avec une analyse de la deuxième dérivée, comme le montre le graphique ci-dessous :

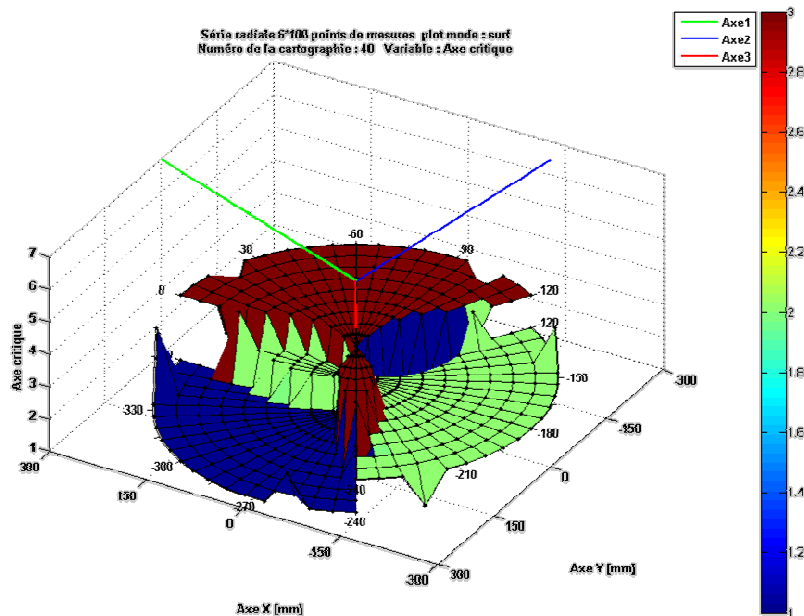


Figure 76 : Cartographie de l'axe critique de la Série 40

Mis à part quelques fluctuations sur l'extérieur du disque, qui s'apparenteraient à des erreurs de mesures, le reste du graphique correspond bien à ce qui a été trouvé avec la série 10. On voit bien que chacun des moteurs du robot peinent, au centre, sous leur propre axe. C'est sûrement dû au fait, qu'à ces endroits-ci, la trajectoire du robot est toujours reprise en grande partie par l'axe situé au dessus.

8.18.6 Série 50

Série N°	Trajectoire	Type d'analyse de variance	Poids [kg]	Z [mm]	Résolution
50	Radiale	2 ^{ème} dérivée	1	-450	6*100

Une série cartographique a été réalisée à une hauteur différente des précédentes, mais toujours avec un lest d'un kilogramme. L'analyse de la variance est cette fois-ci de nouveau le maximum de la deuxième dérivée. Avec une hauteur de Z=-450[mm], les accélérations critiques donnent le graphique suivant :

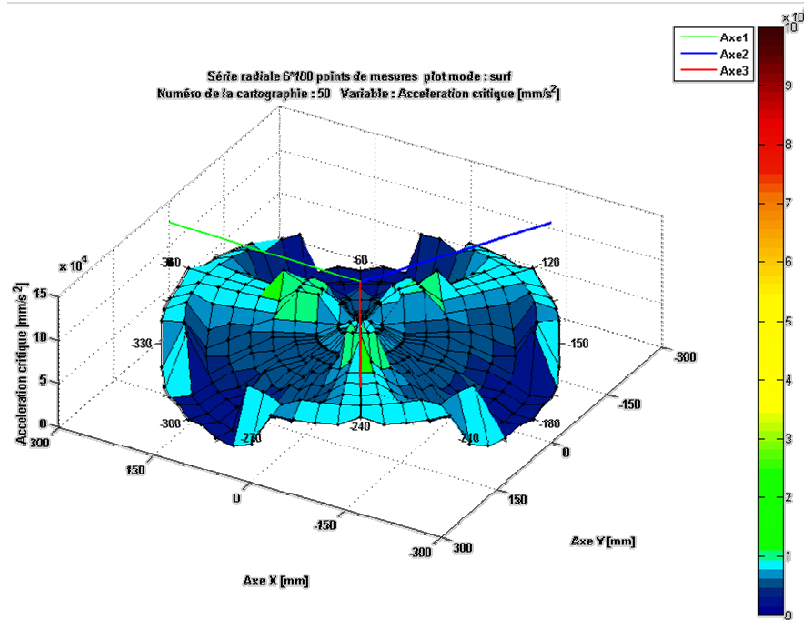


Figure 77 : Cartographie de l'accélération critique de la Série 51

Comparée à la série 30, les valeurs sont moins élevées. A cette hauteur, le robot a plus de peine à suivre pour mouvements qui ne sont pas réalisés en dessous d'un axe. Les variances associées sont exposées dans le graphique suivant :

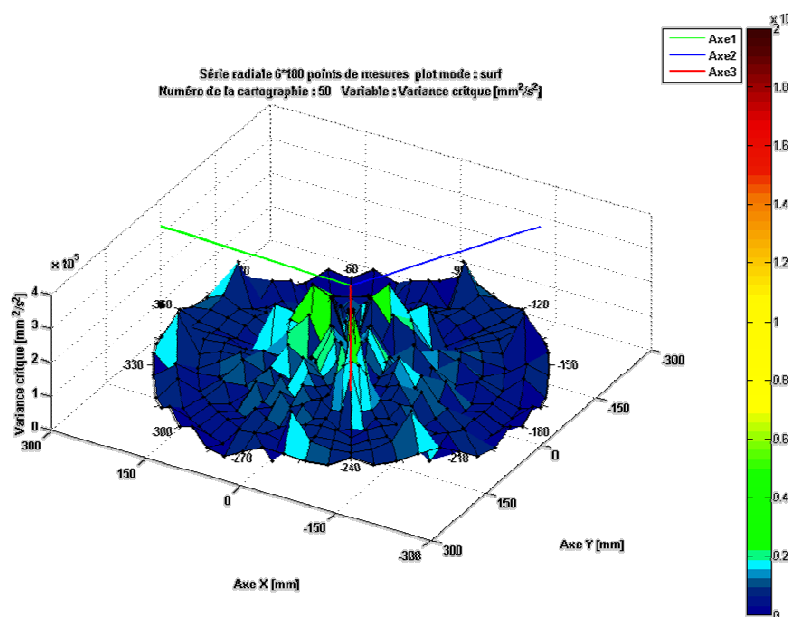


Figure 78 : Cartographie de la variance critique de la Série 40

Les valeurs sont un peu plus élevées qu'à une hauteur de $Z = -350$ [mm], mais on retrouve toujours une certaine augmentation au centre du plan de transition.

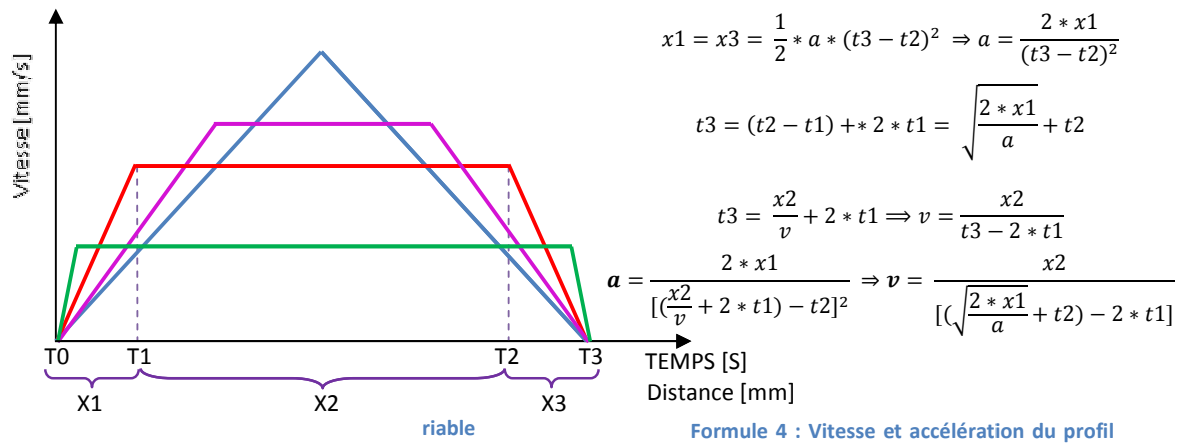
8.19 Synthèse des cartographies

Les cartographies du chapitre précédent font ressortir beaucoup d'informations sur le comportement du Robot Delta. Mais pour en réaliser une analyse pointue et interpréter les résultats parfaitement adéquats, il faudrait avoir des connaissances plus approfondies en robotique, plus spécialement sur le Robot Delta. Mais sans avoir cette expérience il est possible d'en tirer quelques conclusions intéressantes :

- ◆ Les trajectoires sont clairement avantageuses lorsqu'elles sont effectuées sur un des trois axes du robot surtout pour des mouvements radiaux.
- ◆ Pour les trajectoires tangentielles, les valeurs sont plutôt constantes.
- ◆ A vide, les valeurs des accélérations sont extrêmement élevées, il en va de même pour les erreurs de variances associées.
- ◆ L'ajout d'un poids d'un kilogramme diminue d'un facteur 5 environ les performances du robot.
- ◆ Si l'on ne se préoccupe pas de la précision du mouvement, l'analyse de variance avec le maximum de la deuxième dérivée est préférentielle à la variance plateau. Ce serait le cas si l'on estime que le robot dispose de suffisamment de temps afin de corriger les erreurs de vitesse et de position lors de la descente vers la cible à atteindre.
- ◆ La hauteur Z du plan de transition influe sur la valeur des accélérations critiques. Seulement une seule cartographie a été réalisée à une hauteur différente de $Z = -350$ [mm], mais il est certain que la hauteur du plan de transition influe sur les performances du robot. Ce comportement est spécialement ressenti sur les zones opposées aux axes moteurs.

9 ECHAUFFEMENTS MOTEURS

Ce point important a été abordé en début du rapport. Par la suite, le projet c'est plutôt concentré sur la manière de cartographier accélérations critiques. Le sujet a pourtant été régulièrement abordé avec Monsieur Jean-Daniel Marquard et plusieurs idées ont surgies. La première était bien sûr celle développée au chapitre 5.6 en reconstituant les échauffements à partir des équations du moteur à courant continu. Une direction de à explorer serait de réaliser des cycles de mouvements de longueur fixe et de durée fixe avec une accélération et une vitesse variable. Le but serait de comparer les pertes calorifiques des moteurs entre des mouvements composés d'une grande accélération avec une vitesse basse et des mouvements avec une petite accélération et une vitesse élevée comme le démontre l'image ci-dessous.



Cette analyse permettrait de savoir quel profit d'accélération et de vitesse serait le plus favorable selon la trajectoire du robot afin de limiter les échauffements des moteurs. Les paramètres sont tous connus si on se base sur les formules mathématiques ci-dessous en prenant comme exemple le profil de vitesse rouge.

Une solution proposée par Monsieur Michel Huser était d'intégrer des sondes thermiques aux moteurs et d'en ressortir une valeur de température dans la programmation. Cette dernière serait ensuite traitée par un algorithme qui déterminerait une consigne de couple. Le principe serait que le robot travaillerait initialement à des consignes élevées puis s'adapteraient en fonction de l'échauffement des moteurs, donc à son point de fonctionnement. Dans le cadre de plusieurs robots traitant un convoyeur, cela permettrait d'enclencher en début de cycle, uniquement les machines en amont de la bande transporteuse avec des consignes élevées. Puis les autres machines seraient activées progressivement dès que les premières verraient leur cadence diminuer avec l'échauffement des moteurs. La courbe suivante illustre un exemple de fixation du couple en fonction de la température mesurée sur un moteur.

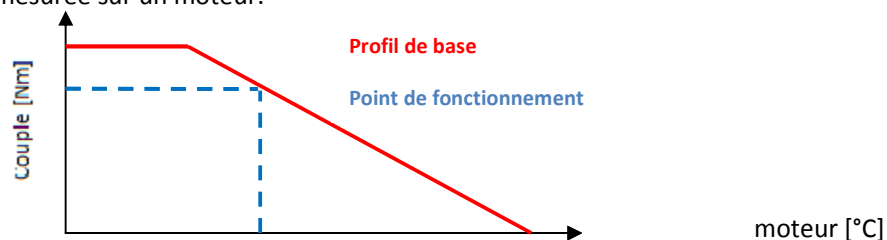


Figure 80 : Fixation du couple en fonction des échauffements

10 DISCUSSION DES RÉSULTATS

Les résultats des cartographies soulèvent un intérêt particulier et permettraient à Bosch Packaging Technology SA d'obtenir des informations utiles pour le développement du Robot Delta. La majorité des remarques ont directement été faites dans les chapitres concernés afin de concentrer les informations avec les nombreux thèmes et principes qui entourent le Robot Delta.

La première partie de ce Travail de Diplôme a été plutôt consacré à l'étude du robot ainsi qu'à sa remise en état de fonction pour travailler sans dérangements informatiques. La suite du travail a été réalisée dans l'optique d'être facilement récupérable et optimisable. Bon nombre de thèmes ont été évoqués tout au long du rapport, d'autres ont à peine été survolés voir mis de côté afin de ne pas trop surcharger ce-dernier et de se concentrer sur ce qui était vraiment essentiel. Par exemple, l'implémentation des programmes d'acquisition, en TurboPascal, n'ont pas beaucoup été expliqués, car les commentaires au sein de la programmation permettent facilement de s'y retrouver et de comprendre en grande partie le fonctionnement.

Le rapport a été volontairement retranscrit le plus chronologiquement possible, afin de représenter au mieux la manière dans laquelle le Travail de Diplôme a été réalisé et de rendre le plus accessible possible le concept en s'immergeant progressivement.

On est bien loin de l'optimisation finale du Robot Delta, mais les résultats cartographiques sont une grande avancée pour la suite du projet. Il en résulte bon nombre de perspectives.

10.1 Problèmes rencontrés

La première difficulté a été le manque d'informations liées au robot. C'est en se plongeant dans la documentation complexe de Demarex SA, en lisant d'anciens Travaux de Diplôme et en effectuant différentes recherches sur internet qu'il a été possible de rassembler les informations nécessaires. Le travail de Diplôme de Monsieur Christophe Gabbud, en 2004, m'a particulièrement aidé.

Passablement des bugs informatiques m'ont retardé dans le début de mon travail, tels que des blocages de ports de communications et des ralentissements importants. Une cause principale était l'antivirus jugé « trop actif » mais verrouillé par le service informatique de l'école.

Le logiciel Matlab permet de réaliser une quantité énorme de fonction. Mais il a nécessité un certain temps d'adaptation pour s'y retrouver parfaitement et utiliser les bonnes syntaxes lors des appels de fonctions de base. L'aide contextuelle associée est très performante mais l'on peu vite s'y perdre.

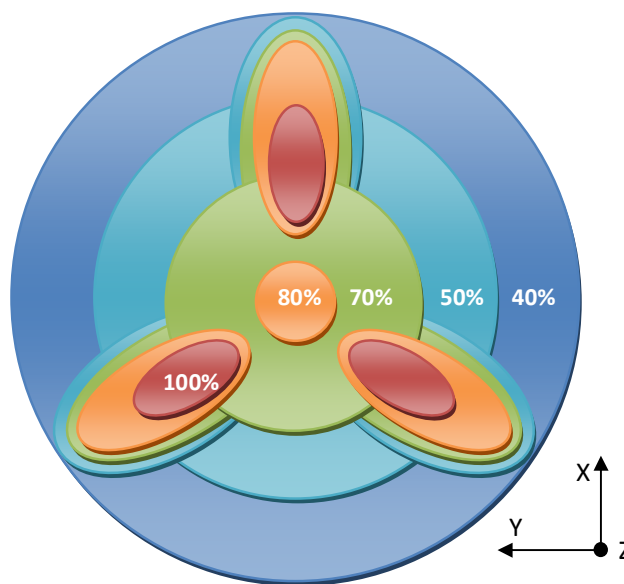
Le procédé d'échantillonnage pour isoler individuellement chaque mouvement d'un cycle d'essai, a pris un certain temps avant d'être optimal. En effet, la formule permettant de détecter un départ de mouvement n'a pas été trouvée du premier coup et c'est au fil des essais que la méthode a été affinée.

La limitation du système d'exploitation DOS, sur le PC Îlot, a eu comme conséquence une grande restructuration et la division du code commande du robot 3 parties. Le souhait initial était d'avoir un programme unique gérant l'intégralité des cycles d'essais.

10.2 Améliorations et perspectives

La première grande amélioration serait l'automatisation de l'acquisition, le traitement et la sauvegarde des données du robot. En effet, cela prend passablement de temps d'effectuer ces étapes avec le système actuel, malgré la possibilité de passer rapidement d'un cycle d'essai à un autre, ainsi que l'aide à la variance critique. L'idée serait d'instaurer une petite communication entre le PC îlot et le PC Superviseur indiquant l'état du cycle d'essais pour débiter le traitement des données et la fin de celui-ci pour passer au cycle d'essais de variance. Si les essais sont automatisés, une cartographie complète du robot pourrait être réalisée rapidement et ouvrir bon nombre de perspectives.

La manière de réaliser un plan d'assignation des consignes se verraient un peu modifiée en comparaison à ce qui a été développé dans le rapport au chapitre 8.6 En effet, avec les résultats de des cartographies, donnant une allure d'hélice à trois pâles, le plan d'assignation pourrait ressembler à ceci :



..

Figure 81 : Idée d'un futur principe d'assignation

Sous Matlab, bon nombre de fonctions pourraient être améliorées :

- ◆ Simplifier la sélection des essais sauvegardés.
- ◆ Rendre fonctionnel les cartographies des valeurs critiques propres à chaque axes.
- ◆ Simplifier le programme en réalisant plus d'appels de fonctions
- ◆ Optimiser le remplissage du tableau de secteur cartographique
- ◆ Améliorer l'algorithme de sélection de l'axe critique

Il serait peut-être plus judicieux de réaliser le même programme que Matlab mais dans un langage plus approprié à l'utilisation et à la représentation graphique. Une interface sous Labview, de Texas Instrument, serait intéressante.

11 CONCLUSIONS

L'acquisition de données issues des mouvements du Robot Delta a été simplifiée et optimisée. Il est désormais possible de réaliser des cycles d'essais, c'est à dire la répétition d'une trajectoire précise, en augmentant progressivement la consigne d'accélération, afin d'en définir sa valeur critique selon l'axe du robot le plus défavorable. Cette valeur critique est déterminée par le traitement de la variance de l'erreur de vitesse du robot. Deux méthodes permettent cette analyse, soit par la détection de l'augmentation brutale de la variance par la méthode de la deuxième dérivée, soit en fixant une variance maximale à ne pas dépasser. L'utilisateur est aidé dans le traitement des données par une aide d'analyse automatique.

Les essais peuvent être sauvegardés et retraités par la suite. La conjugaison de plusieurs essais permet de réaliser une cartographie en trois dimensions d'un plan de transition du robot selon un paramètre précis, par exemple l'accélération critique. Les cartographies, réalisées dans le cadre de ce Travail de Diplôme, fournissent un grand nombre d'informations sur le fonctionnement du Robot Delta et ouvrent une multitude de perspectives pour l'approfondissement de celles-ci.

« L'optimisation des trajectoires d'un Robot Delta », représentait l'objectif de ce projet, mais compte tenu du peu de temps à disposition et du travail considérable à exécuter, le résultat final ne se trouve pas au sein de ce rapport. Néanmoins, un grand pas en avant a été réalisé notamment par la représentation du comportement du robot sur une cartographie en trois dimensions. Même s'il reste encore un effort important à fournir pour développer et valider une optimisation concrète du Robot, la problématique a vigoureusement été abordée. Ce rapport recueille un nombre appréciable de développements et d'idées, principalement la logique de correction des consignes d'accélération et de vitesse ainsi que la limitation des échauffements moteurs, donnant en définitive diverses directions à explorer.

Malgré le fait que le cahier des charges initial n'a pas été comblé, j'estime que ce Travail de Diplôme est une réussite, car des résultats concrets ont été produits et les problématiques futures abordées. Il n'a pas été facile de s'immerger dans la recherche et le développement d'une machine qui m'était totalement inconnue, mais la conception et le fonctionnement du Robot Delta a progressivement éveillé ma curiosité. J'ai ainsi pris beaucoup de plaisir à chercher les solutions et optimiser celles qui me paraissaient les meilleures.

Je souhaite aux développeurs de Bosch Packaging Technology SA de trouver dans ce rapport des pistes pour l'amélioration du Robot Delta provenant d'un point de vue extérieur. Je souhaite aussi aux futurs utilisateurs que ce document fournisse le plus d'informations possibles afin de leur permettre une compréhension et un apprentissage rapide de la machine. J'espère aussi qu'il aura éveillé la curiosité du lecteur pour le domaine de la robotique, tout comme il a éveillé la mienne.

Au final, le Robot Delta, par les cartographies, dévoile désormais un peu plus de ses attitudes, de ses comportements, un peu plus de son visage...

Sion, le 11 Juillet 2010

Michael Briguet

12 BIBLIOGRAPHIES

- [1] Documentation Demarex SA pour Delta95 – V3.3 (2002) par JFS/MHU.
- [2] Thèse N° 925 de l'EPFL 1991 « Conception d'un robot parallèle rapide à 4 degrés de liberté » par Reymond Clavel.
- [3] Livre « Programme en TurboPascal » Par Frédéric Blanc et Pierre Brandeis Editions PSI (1989).
- [4] Travail de Diplôme 2004 « Application de démonstration pour robot Delta » par Christophe Gabbud.
- [5] Travail de Diplôme 2009 « Ajustement automatique des consignes de vitesse et d'accélération d'un robot » par Pierre Fournier.
- [6] Site web de Bosch Packaging Technology SA.
<http://www.boschpackaging.com/>
- [7] Site web du club des développeurs francophones.
<http://www.developpez.com/>
- [8] Site web de Matlab.
<http://www.mathworks.com/>
- [9] Site web d'informations sur les robots du type parallèle.
<http://www.parallemic.org/>
- [10] Site web d'informations sur les notions de statistiques.
<http://biblioxtrn.uqar.qc.ca/stat/Default.htm>

13 REMERCIEMENTS

Je tiens particulièrement à remercier les personnes citées ci-dessous pour le temps précieux qu'ils m'ont accordés, leur soutien et leurs encouragements tout au long de ce Travail de Diplôme.

- ◆ Monsieur Marcuard Jean-Daniel, professeur à l'HES-SO Valais.
Pour ses connaissances en robotique et en automatisation. Ses précieux conseils m'ont permis de ni m'emballer ni me décourager lors des diverses étapes du travail et de choisir toujours la solution la plus constructive possible.
- ◆ Messieurs Michel Huser et Alain Teklits de Bosch Packaging Technology.
Pour leur gentillesse, leur disponibilité et les réponses très précises lors de mes nombreuses interrogations
- ◆ Monsieur Paul-Henri Emery, responsable de l'atelier mécanique de l'HES-SO Valais.
Pour sa gentillesse et la réalisation rapide du lest du robot.
- ◆ Ma classe, volée 2007, particulièrement Johan, Valentin, Jérôme et Jonathan.
Pour les nombreux encouragements et échangeant d'idées entourés d'une ambiance géniale.
- ◆ Ariane et Olivier Felley.
Pour la relecture assidue de mon rapport et les corrections apportées.
- ◆ Conception et Bertrand.
Pour le soutien et les encouragements qu'ils m'ont apportés durant ces années d'HES.
- ◆ Ma famille et surtout mes parents.
Pour leur soutien infaillible depuis le début de mes études et leur omniprésence lors des moments de doutes.
- ◆ Et tout particulièrement à Alexandra. Outre la relecture de mon rapport et tes précieux conseils, merci de ta présence de chaque instant et d'avoir sans cesse cru en moi.

« Il y a, au cours de l'existence, des rencontres imprévues et singulières, où bien des faits se produisent, en apparence anodins, et qui auront été les dons du destin. »

Hector Bianciotti

14 TABLE DES FIGURES

Figure 1 : Robot Delta de l'HES-SO	2
Figure 2 : Robot Delta vu de face avec son volume de travail	3
Figure 3 : Robot Delta vu de dessus avec son plan de travail	3
Figure 4 : Axes cartésiens du robot	3
Figure 5 : Cycle "Pick and Place"	4
Figure 6 : Principe des "Overlaps"	4
Figure 7 : Profil théorique de vitesse d'un mouvement	5
Figure 8 : Résumé de l'environnement de travail	6
Figure 9 : Environnement de Travail	7
Figure 10 : Exemple de paramètres Hardware	8
Figure 11 : Architecture du programme	8
Figure 12 : Noyau multitâche	8
Figure 13 : Principe de la boucle de régulation PIDF	9
Figure 14 : Accueil EX_10.PAS	10
Figure 15 : Dossier partagé	10
Figure 16 : Graphique Matlab initial	11
Figure 17 : Décalage de la courbe théorique, en rouge, et réelle, en vert	11
Figure 18 : D95Conf	12
Figure 19 : Principe des modèles géométriques	12
Figure 20 : Comparaison des valeurs	13
Figure 21 : Relais de sécurité du robot (en rouge)	13
Figure 22 : Câblage de la pompe à vide	14
Figure 23 : Moteurs et pompe à vide	14
Figure 24 : Idée de carte de valeurs	14
Figure 25 : Idée d'assignation	15
Figure 26 : Assignation d'un volume de travail	15
Figure 27 : Variante de fonctionnement 1	17
Figure 28 : Variante de fonctionnement 2	18
Figure 29 : Zones symétriques du robot	19
Figure 30 : Types de trajectoires	20
Figure 31 : Zones circulaires	21
Figure 32 : Zones triangulaires	21
Figure 33 : Point de départs de plusieurs trajectoires	23
Figure 34 : Ecart types	24
Figure 35 : Allure d'une erreur de vitesse	24
Figure 36 : Allure de l'erreur de vitesse sous "D95Conf"	25
Figure 37 : Augmentation de la variance	25
Figure 38 : Graphique Matlab des erreurs de vitesse	26
Figure 39 : Délimitation de l'échantillonnage	27
Figure 40 : Graphique Matlab de la variance	28
Figure 41 : Principe des microsamples	30
Figure 42 : Variance déterminant l'incrément L	31
Figure 43 : Tableau des codes radiaux 64 points	32
Figure 44 : Tableau des codes tangentiels 45 points	32

Figure 45 : Accueil du programme "Delta" sous Matlab	34
Figure 46 : Nouvelle acquisition sous Matlab	35
Figure 47 : Type de variance sous Matlab	35
Figure 48 : Analyse plateau théorique	36
Figure 49 : Traitement de l'acquisition sous Matlab	37
Figure 50 : Principe de sauvegarde d'un essai unique	38
Figure 51 : Principe de la sauvegarde des essais d'une sérié cartographique	38
Figure 52 : Début d'un fichier TXT	38
Figure 53 : Recherche d'un ancien essai	39
Figure 54 : Graphique Matlab de base et polarplot3d.m	40
Figure 55 : Secteur des points de mesure	41
Figure 56 : Secteur des points de cartographie.....	41
Figure 58 : Cartographie de l'accélération critique de la Série 1	42
Figure 57 : Cartographie d'un secteur de la Série 1	42
Figure 59 : Cartographie de la variance critique de la Série 1.....	43
Figure 60 : Superposition des secteurs	45
Figure 62 : Types de représentation graphique 3D : Mesh, Surf, Surfn et Contour.....	46
Figure 61 : Sélection de la variable à Cartographier	46
Figure 63 : Nacelle avec masse d'1 [kg].....	47
Figure 64 : Cartographie de l'accélération critique de la Série 10	48
Figure 65 : Cartographie de la variance critique de la Série 10.....	48
Figure 66 : Cartographie de l'axe critique de la Série 10.....	49
Figure 67 : Cartographie de l'accélération critique de la Série 11	50
Figure 68 : Cartographie de la variance critique de la Série 11.....	50
Figure 69 : Cartographie de l'axe critique de la Série 11.....	51
Figure 70 : Cartographie de l'accélération critique de la Série 30	51
Figure 71 : Cartographie de la variance critique de la Série 30.....	52
Figure 72 : Cartographie de l'accélération critique de la Série 31	53
Figure 73 : Cartographie de la variance critique de la Série 31.....	53
Figure 74 : Cartographie de l'accélération critique de la Série 40	54
Figure 75 : Cartographie de la variance critique de la Série 40.....	54
Figure 76 : Cartographie de l'axe critique de la Série 40.....	55
Figure 77 : Cartographie de l'accélération critique de la Série 51	56
Figure 78 : Cartographie de la variance critique de la Série 40.....	56
Figure 79 : Profil de vitesse variable.....	58
Figure 80 : Fixation du couple en fonction des échauffements	58
Figure 81 : Idée d'un futur principe d'assignation.....	60

15 TABLE DES TABLEAUX

Tableau 1 : Correspondances symétriques	20
Tableau 2 : Exemple de référence	20
Tableau 3 : Relations des zones circulaires	21
Tableau 4 : Relations des zones triangulaires	22
Tableau 5 : Tableau des codes radiaux 100 points.....	44
Tableau 6 : Remplissage du tableau "SectorTab" des essais tangentiels.....	45
Tableau 7 : Liste des cartographies	47

16 TABLE DES CODES

Code 1 : Cycle de mouvement.....	26
Code 2 : Boucle d'échantillonnage	27
Code 3 : Condition de détection de mouvement	27
Code 4: Echantillonnage	27
Code 5: Procédure "AssignSerie"	33
Code 6 : Variance plateau sous Matlab	36

17 TABLE DES ÉQUATIONS

Formule 1 : Equations du moteurs DC	15
Formule 2 : Variance	24
Formule 3 : Ecart type	24
Formule 4 : Vitesse et accélération du profil variable.....	58

18 ANNEXES

1. Bilan intermédiaire.
2. Tableaux des points de mesures.
3. Aperçu du code TurboPascal « Cartra1.PAS ».
4. Aperçu des codes sous Matlab.
5. Mode d'emploi.
6. Structogrammes.
7. Exemple d'un fichier TXT.
8. Cartographies.

ANNEXE 1

Bilan intermédiaire

Rapport intermédiaire du travail de diplôme

Diplômant Michaël Briguet

Optimisation des trajectoires d'un robot Delta

Rappel des principaux objectifs

Le robot parallèle de type Delta, fabriqué par Bosch Packaging à Romanel, dispose de caractéristiques très intéressantes pour déplacer rapidement de petits objets ou comestibles dans des « blisters ».

Les consignes d'accélération et de vitesse du robot sont fixées en fonction du segment de trajectoire le plus défavorable de la zone de travail et ne varient plus. Les performances du robot se retrouvent ainsi restreintes alors que celui-ci pourrait effectuer des mouvements plus rapides dans certaines parties de son volume de travail. En optimisant, durant le cycle de travail, les consignes d'accélération et de vitesse selon sa trajectoire, le robot verrait sa cadence et ainsi sa productivité augmenter en tenant compte des échauffements moteurs.

Afin d'évaluer et d'interpréter les mouvements des différents axes du robot, un programme acquiert diverses informations (position, vitesse, erreur, couple) et permet son traitement avec le logiciel Matlab sous forme de graphes. A noter que le robot est commandé depuis un ordinateur fonctionnant sous DOS (PC îlot) et que le traitement des informations se fait via un autre ordinateur (PC superviseur) évoluant avec XP. Le transfert des informations s'effectue par un dossier partagé.

Pour créer un algorithme permettant d'améliorer la productivité du robot en modifiant ses consignes d'accélération et de vitesse tout en limitant l'échauffement des moteurs, différents essais seront programmés pour être ensuite analysés à l'aide des informations recueillies sous Matlab. Des tests finaux valideront cet algorithme en le comparant au fonctionnement actuel du robot.

Déroulement du Travail de Diplôme :

- Apprentissage de l'environnement du robot Delta (Pascal, DOS, Matlab, Manipulation du Robot).
- Analyse théorique : sélection des variables à étudier.
- Mise au point d'un cycle de test permettant de recueillir les données significatives.
- Analyse des données (Logiciel Matlab).
- Développement d'une stratégie attribuant une accélération et une vitesse idéales à chaque mouvement.
- Programmation de la stratégie dans la commande du robot et validation par des essais.

Tâches réalisées

L'apprentissage de la machine ainsi que son environnement était l'objectif premier de ce début de Travail de Diplôme. L'architecture spécifique associée au robot, la manière de le mettre en route, de le programmer, de le faire fonctionner correctement et finalement d'acquérir les données désirées, ont nécessité une intention particulière.

Le manque d'informations sur certains aspects de la machine a exigé de longues recherches dans la documentation officielle du robot ainsi qu'à la médiathèque. Le rapport du Travail de Diplôme de mon prédécesseur, Monsieur Pierre Fournier, en 2009, contenait très peu d'éléments concernant la mise en route, la manipulation et la commande du robot. Tandis que le rapport d'un ancien Travail de Diplôme, celui de Monsieur Christophe Gabbud en 2004. Il avait partiellement traité de la programmation du robot Delta et son rapport possédait certaines informations essentielles à la manipulation et la compréhension de celui-ci.

Premièrement, il a fallu résoudre des problèmes informatiques avant de pouvoir utiliser correctement le robot. Il s'est avéré, en outre, que l'antivirus du PC superviseur empêchait la liaison avec le PC îlot sur le dossier de partage en bloquant des ports de communication.

La suite de l'apprentissage s'est déroulée en manipulant le robot à l'aide du logiciel « D95Conf ». Ce soft a permis, en pilotant manuellement le robot, de tester ses différents paramètres et d'évaluer ses limites dans son volume de travail. La pompe à vide, qui permet de créer un vacuum au bout de la nacelle du robot, ne s'enclenchait pas. Une légère modification du câblage a donc été effectuée pour rendre opérationnelle cette fonctionnalité.

Afin de réaliser des cycles de test, il était primordial de se plonger dans le programme permettant de lancer automatiquement des procédures au robot. Le fichier « EX_10 .PAS » de mon prédécesseur a servi de base pour comprendre et mettre en pratique les différentes fonctions du robot en s'aidant d'une légère interface graphique sous DOS, le tout programmé en langage Pascal.

Initialement, les données du couple des axes du robot n'étaient graphiquement pas disponibles sous Matlab. Il a fallu modifier et tester l'acquisition de ces données sans pouvoir totalement comprendre la fonctionnalité des « microsamples », à cause du manque d'informations à ce sujet.

Actuellement, la programmation d'une première séquence de test est en cours. Elle a pour but d'évaluer les erreurs de positions et de vitesses sur un plan de travail horizontal en fonction de la trajectoire effectuée à différentes valeurs de consignes.

■ Problèmes identifiés

Un détail agaçant subsiste encore dans l'ergonomie de travail. En effet, l'accès au dossier partagé, pour transmettre les fichiers de programmation, se bloque régulièrement et empêche ainsi le fonctionnement du PC îlot sous DOS. Son redémarrage est donc souvent nécessaire, afin de travailler à nouveau de manière correcte.

En augmentant les consignes d'accélération et de vitesse, les moteurs fourniront plus rapidement une quantité de couple plus importante, induisant ainsi des déperditions thermiques supplémentaires. Il est alors important de les quantifier. A l'heure actuelle, il reste tout de même un doute sur le type de commande des moteurs (tension ou courant). L'acquisition de la tension et du courant via un oscilloscope permettra de reconstituer mathématiquement les pertes calorifiques des moteurs et de garantir que le futur algorithme ne fasse pas surchauffer les moteurs après un certain temps de travail.

Il reste encore quelques points à éclaircir, comme la manière d'interpréter les résultats des essais sur les différents axes et comment les cartographier (symétrie des axes). Et aussi de déterminer une façon optimale de modifier les consignes dans la programmation.

■ Tâches à réaliser

Pour la suite, il faudra tout d'abord évaluer et implémenter une stratégie d'optimisation de consigne sur une surface de travail horizontale en forme de cercle (2D). Donc uniquement pour des mouvements à une hauteur fixe. L'étape suivante sera de transposer cette stratégie sur un volume de travail en forme de cylindre (3D). Finalement, les déperditions thermiques sur les moteurs seront considérées et la manière de les limiter sera implémentée dans l'algorithme d'optimisation.

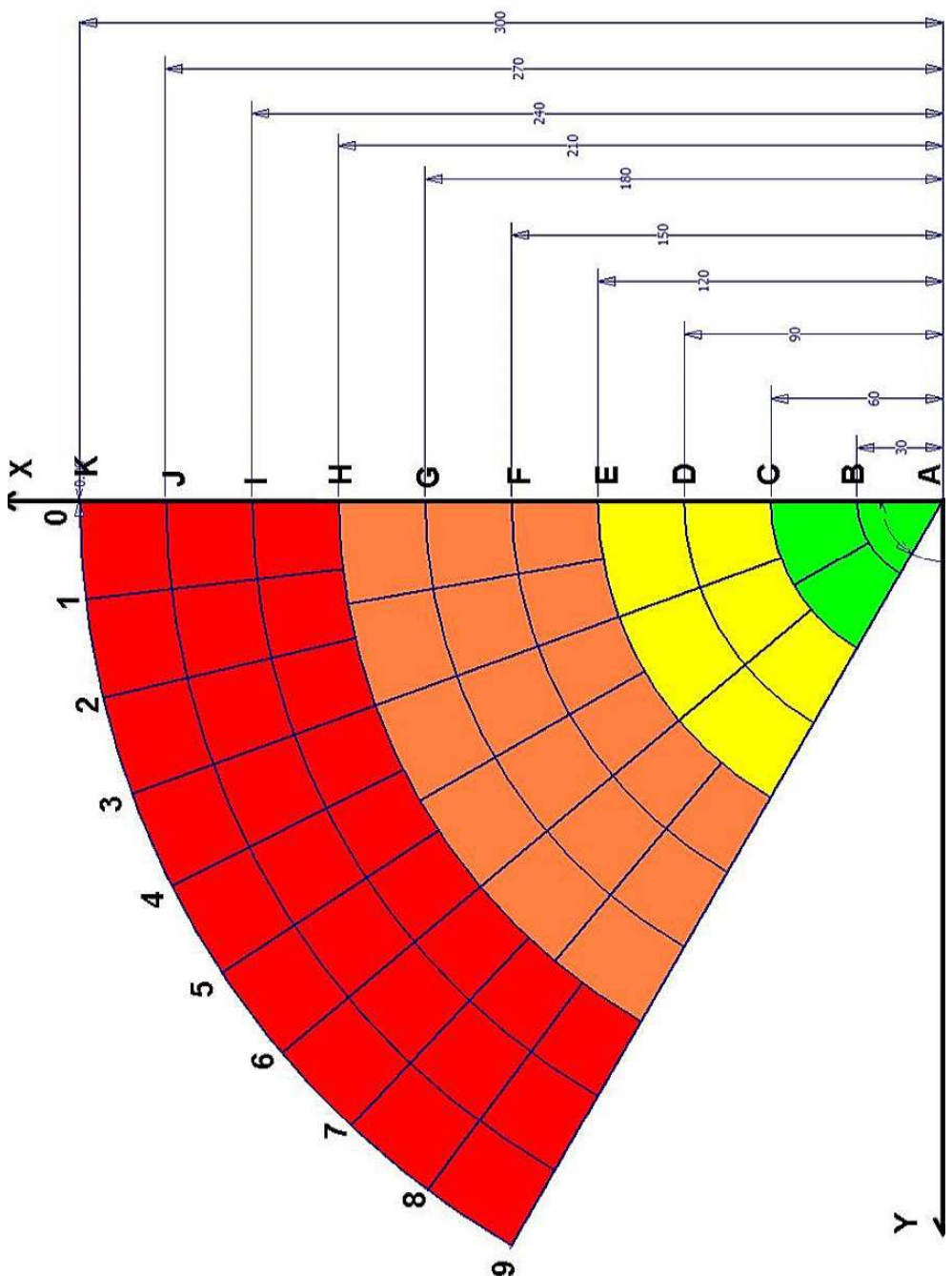
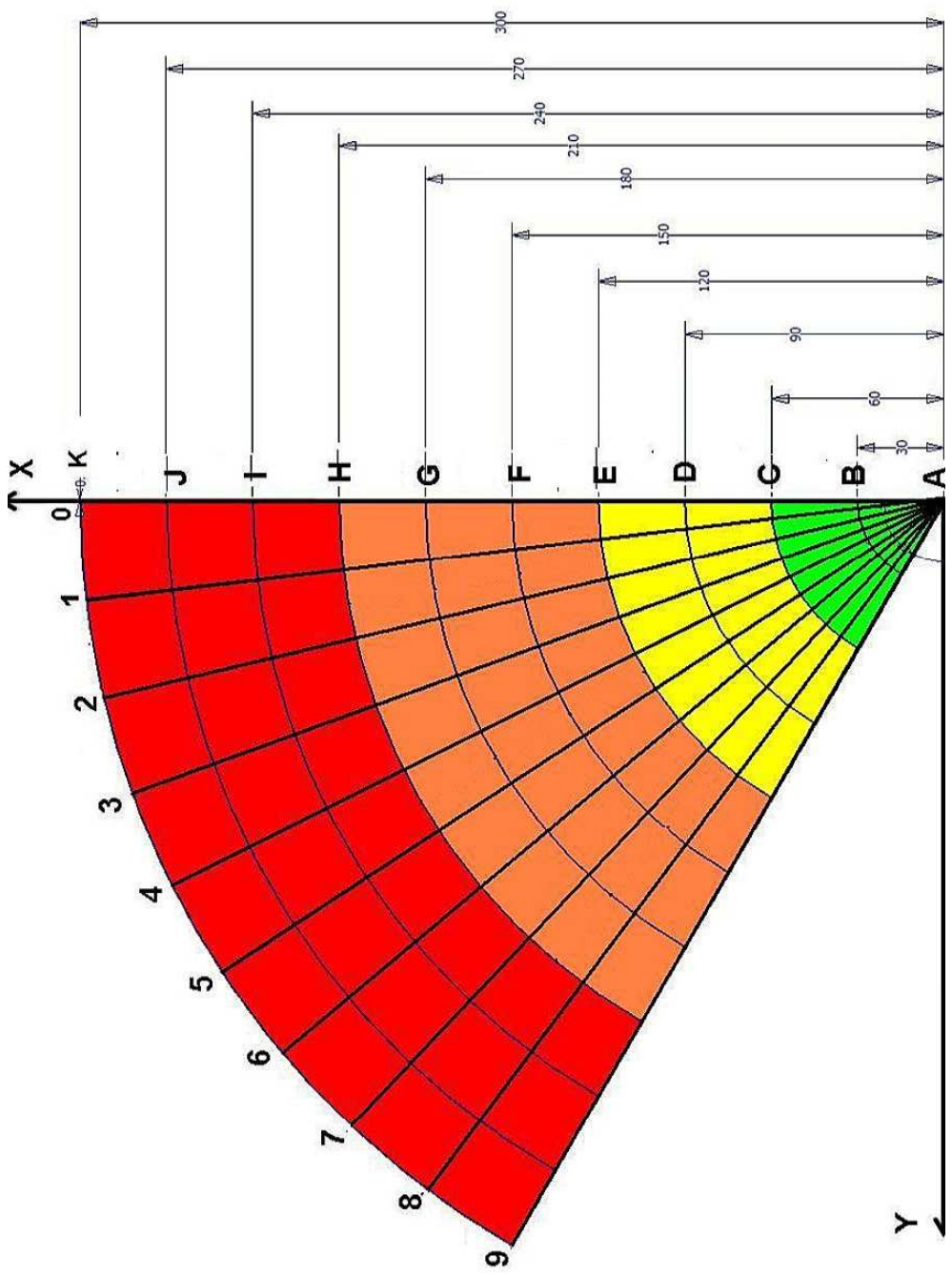
Déroulement des prochaines activités :

- Comparer les valeurs des acquisitions sous Matlab si elles correspondent bien à celles du D95Conf.
- Déterminer le type de commande des moteurs (tension ou courant).
- Finaliser la première séquence de test (mouvements simples sur un plan 2D) et interpréter les résultats.
- Définir une séquence d'essai permettant de "cartographier" les erreurs de chaque trajectoire sur un plan de travail 2D et la programmer.
- Elaborer un algorithme qui modifie les consignes d'accélération et de vitesse selon la trajectoire du robot sur un plan de travail 2D.
- Programmer une séquence de travail sur un plan en 2D, la tester avec et sans optimisation des consignes puis comparer les résultats.
- Définir une séquence d'essai permettant de "cartographier" dans un volume de travail 3D.
- Modifier l'algorithme faisant varier les consignes d'accélération et de vitesse pour un volume de travail 3D.
- Programmer une séquence de travail sur un plan en 3D, la tester avec et sans optimisation des consignes puis comparer les résultats.
- Reconstituer les pertes calorifiques des moteurs
- Etablir une stratégie afin de limiter ces échauffements et l'implémenter dans la programmation.
- Procéder à des essais finaux puis comparer les résultats avec et sans optimisation.

ANNEXE 2

Points de mesures

- **TABLEAU DES COORDONNÉES (VERSION 64 POINTS).**
- **TABLEAU DES COORDONNÉES (VERSION 100 POINTS).**
- **IMAGE DES SECTEURS DE MESURES.**



ANNEXE 3

Programme d'acquisition sous DOS

- **CARTRA1.PAS**

PAR LA LONGUEUR CONSÉQUENTE DES TROIS PROGRAMMES, PLUS DE 3800 LIGNES CHACUN, SEUL UN APERÇU DES LIGNES ESSENTIELLES DE «CARTRA1.PAS » ONT ÉTÉ IMPRIMÉES. LES CODES COMPLETS SONT DISPONIBLES EN ANNEXE SUR LE CD.

Aperçu du code d'acquisition du Robot Delta

« CARTRA1.PAS » Version 1.2 : 3801 Lignes

ENTÊTE : Ligne 1

```
{*****}
{ Programme de cartographie du plan de travail }
{ HES_S O S I O N }
{ Project: Optimisation Robot Delta }
{ File name : CartRa1.pas }
{ }
{ Designed by : Briguet Michael }
{ Version V 1.2 Date : 05.07.2010 }
{ }
{ DEMAUREX Robotique & Microtechnique SA TEST }
{ Chemin des M,sanges 3 }
{ 1032 Romanel sur Lausanne / Switzerland }
{ }
{ Project : DELTA95 system software }
{ File name : CARTRA1.pas }
{ }
{ Designed by : MHU }
{ Version V 1.00 Date : 31-jan-1997 }
{ }
{ Modified by : MHU }
{ Version V 3.03 Date : 23-may-1997 }
{*****}
```

Variables importantes : Ligne 133

VAR

```
{global variables defined for the record}
Objet : Fich; {Type de fichier de sauvegarde}
InfoDateTime : FichierDateTime; {Fichier des parametres temporels}
InfoParam : FichierParam; {Fichier des parametres du mouvement}
Tab : Tableau; {Tableau de sauvegarde 1}
Tab2 : Tableau; {Tableau de sauegarde 2}
i : Integer; {Index remplissage de tableau}
ManCount:Integer; {Index choix des coordonnees manuel}
Reponse : Char; {Variable de reponse utilisateur}
AccelVal : c_realType; {Valeur d'acceleration par default}
SpeedVal : c_realType; {Valeur de vitesse par default}
newAccel : c_realType; {Nouvelle valeur d'acceleration}
newSpeed : c_realType; {Nouvelle valeur de vitesse}
modifAccel : Boolean; {Bit de modification d'acceleration}
modifSpeed : Boolean; {Bit de modification de vitesse}
parametres : Boolean; {Bit de modification des parametres}
continue : Boolean; {Bit de suivi de Serie}
trajectoire : word; {Code de trajectoire}
loc : locationType ; {Coordonne de mouvement}
locName1 : String; {Nom du point de depart}
```

```
locName2 : String; {Nom du point d'arrivee}
{Variables temporelles}
Annee : word;
Mois : word;
Jour : word;
JourDeSemaine : word;
HeureDebut : word;
MinuteDebut : word;
SecondeDebut : word;
Seconde100Debut : word;
HeureFin : word;
MinuteFin : word;
SecondeFin : word;
Seconde100Fin : word;
{Bit de controle des tableaux}
Tableau1Plein : Boolean;
Tableau2Plein : Boolean;
Tab1enCours : Boolean;
AccelStepInt: longint; {Step d'acceleration}
AccelMaxInt: longint; {Acceleration de fin de dycle}
AccelMinInt: longint; {Acceleration de depart de cycle}
NbStepReal :longint; {Nombre de Step}
RunStep :integer; {Index de step}
nextmove: boolean; {Bit de cycle suivant}
restartmove : boolean; {Bit de repetition du cycle}
Serie8 : integer; {Controle de nombre de cycles}
Start_X: c_RealType; {Point de depart en X}
Start_Y: c_RealType; {Point de depart en Y}
Start_Z: c_RealType; {Point de depart en Z}
End_X: c_RealType; {Point d'arrivee en X}
End_Y: c_RealType; {Point d'arrivee en Y}
End_Z: c_RealType; {Point d'arrivee en Z}
Man_Start_X: c_RealType; {Point de depart manuel en X}
Man_Start_Y: c_RealType; {Point de depart manuel en Y}
Man_Start_Z: c_RealType; {Point de depart manuel en Z}
Man_End_X: c_RealType; {Point d'arrivee manuel en X}
Man_End_Y: c_RealType; {Point d'arrivee manuel en Y}
Man_End_Z: c_RealType; {Point d'arrivee manuel en Z}
MoveCode: c_RealType; {Code de mouvement}
StepSerie : integer; {Index de step de serie}
AccelStep: c_RealType; {Step d'acceleration REAL}
AccelMax : c_RealType; {Acceleration de fin de cycle REAL}
AccelMin : c_RealType; {Acceleration de depart de cycle REAL}
NbStep : c_RealType; {Nombre de Step REAL}
RunStepReal : c_RealType; {Index de step REAL}
TargetTime : c_RealType; {Temps de mouvement REAL}
PrepareTargetTime : c_RealType; {Preparation de mouvement REAL}
run : Integer; {Index de nombre de cycles}
nbrCycles: word; {Nombre de cycles}
```


Déclaration des points de locations : Ligne 486

```
{Variables de points de location}  
PointA0 : locationType ;  
PointB0 : locationType ;  
PointB1 : locationType ;  
PointB2 : locationType ;  
etc..
```

Procédure de récupération des coordonnées des points dans le SOFT_REC.CFG : Ligne 579

```
{-----}  
PROCEDURE DecodeLocation_soft_rec (name: STRING; VAR value: locationType; read: BOOLEAN) ; FAR ;  
{-----}  
BEGIN  
    IF FitStrings (name, 'pickPosition') THEN BEGIN  
        IF read THEN pickPosition := value ELSE value := pickPosition  
    END  
    {Controle des point de location de cartographie}  
    ELSE IF FitStrings (name, 'PointA0') THEN BEGIN  
        IF read THEN PointA0 := value ELSE value := PointA0  
    END  
    ELSE IF FitStrings (name, 'PointB0') THEN BEGIN  
        IF read THEN PointB0 := value ELSE value := PointB0  
    END  
    ELSE IF FitStrings (name, 'PointB1') THEN BEGIN  
        IF read THEN PointB1 := value ELSE value := PointB1  
    END  
    ELSE IF FitStrings (name, 'PointB2') THEN BEGIN  
        IF read THEN PointB2 := value ELSE value := PointB2  
    END  
    Etc..  
END
```

Procédure d'initialisation des points de locations : Ligne 987

```
{-----}  
PROCEDURE Read_soft_rec (readOnly: BOOLEAN) ;  
{-----}  
BEGIN  
    {Initialisation des points de location}  
    InitLocation (PointA0) ;  
    InitLocation (PointB0) ;  
    InitLocation (PointB1) ;  
    InitLocation (PointB2) ;  
    InitLocation (PointB3) ;  
    Etc..  
END
```

Initialisation des tableaux d'acquisition : Ligne 1552

```
{-----}
PROCEDURE EmptyTab;
{Cette procedure vide les tableaux d'enregistrement des mesures      }
{-----}
BEGIN
  FOR i:= 1 TO 100 DO
    BEGIN
      Tab[1,i] := 0;
      Tab[2,i] := 0;
      Etc..
      Tab2[17,i] := 0;
      Tab2[18,i] := 0;

    END;
    Tab1enCours := true;
    i:=1;
    Tableau1Plein := false;
    Tableau2Plein := false;
    Ax_CaptureEnable (1,true);
    Ax_CaptureEnable (2,true);
    Ax_CaptureEnable (3,true);

    {Emplacement des fichiers de sauvegarde}
    Assign(Objet,'T:\DATSAVED\DATA.INT');
    ReWrite(Objet);
    writeln('***** ICI *****');
    Assign(InfoDateTime,'T:\DATSAVED\INFOTIME.INT');
    ReWrite(InfoDateTime);
    Assign(InfoParam,'T:\DATSAVED\INFOPARA.INT');
    ReWrite(InfoParam);
    {Assignation de la date et de l'heure}
    GetDate(Annee, Mois, Jour, JourDeSemaine);
    GetTime(HeureDebut, MinuteDebut, SecondeDebut, Seconde100Debut);
  END;
```

Ecriture des paramètres du mouvement : Ligne 1624

```
{-----}
PROCEDURE WriteParam;
{Cette proc,dure ecrit les parametre lies au mouvement}
{-----}
BEGIN
  GetTime(HeureFin, MinuteFin, SecondeFin, Seconde100Fin);
  write(InfoDateTime,Jour );
  write(InfoDateTime,Mois);
  write(InfoDateTime,Annee);
  write(InfoDateTime,HeureDebut);
  write(InfoDateTime,MinuteDebut);
  write(InfoDateTime,SecondeDebut);
  write(InfoDateTime,HeureFin);
  write(InfoDateTime,MinuteFin);
  write(InfoDateTime,SecondeFin);
  write(InfoDateTime,nbrCycles);
  write(InfoDateTime,trajectoire);
```

```

AccelVal := D3n_Q_Accel(left);
SpeedVal := D3n_Q_Speed(left);
write(InfoParam,AccelVal);
write(InfoParam,SpeedVal);
write(InfoParam,Start_X);
write(InfoParam,Start_Y);
write(InfoParam,Start_Z);
write(InfoParam,End_X);
write(InfoParam,End_Y);
write(InfoParam,End_Z);
write(InfoParam,MoveCode);
write(InfoParam,RunStepReal);
write(InfoParam,AccelMin);
write(InfoParam,AccelMax);
write(InfoParam,AccelStep);
write(InfoParam,PrepareTargetTime);
Close(Objet);
Close(InfoDateTime);
Close(InfoParam);
END;
```

Procédure d'assignation des coordonnées avec le code de trajectoire : Ligne 1667

```

{-----}
PROCEDURE AssignSerie;
{Cette procedure assigne les coordonees du mouvement en fonction du code de trajectoire }
{-----}
BEGIN
{Serie Radiale 1}
IF (trajectoire >= 201) AND (trajectoire <= 210) THEN
BEGIN
CASE Trajectoire OF
201: BEGIN
point1:= pointA0;
point2:= pointB0;
locName1:='A0';
locName2:='B0';
END;
202: BEGIN
point1:= pointA0;
point2:= pointB1;
locName1:='A0';
locName2:='B1';
END;
203: BEGIN
point1:= pointA0;
point2:= pointB2;
locName1:='A0';
locName2:='B2';
END;
Etc..
END ;
Etc...
END ;
```


Remplissage des tableaux de données : Ligne 2671

```
{-----}  
PROCEDURE IrqTrace ; FAR ;  
{-----}  
BEGIN  
  {Controle de la variable d'activation}  
  IF startMoving THEN  
    BEGIN  
    {Controle du tableau ouvert}  
    IF Tab1enCours THEN  
      BEGIN  
        {Sauvegarde des donnees del'axe 1}  
        Tab[1,i] := Ax_Q_TheoreticalPositionMicroSample (1,0);  
        Tab[1,i+1] := Ax_Q_TheoreticalPositionMicroSample (1,1);  
        Etc..  
        Tab[1,i+9] := Ax_Q_TheoreticalPositionMicroSample (1,9);  
  
        Tab[2,i] := Ax_Q_PositionMicroSample (1,2);  
        Tab[2,i+1] := Ax_Q_PositionMicroSample (1,3);  
        Etc..  
        Tab[2,i+9] := Ax_Q_PositionMicroSample (1,1);  
  
        Tab[3,i] := Ax_Q_TheoreticalSpeedMicroSample (1,0);  
        Tab[3,i+1] := Ax_Q_TheoreticalSpeedMicroSample (1,1);  
        Etc..  
        Tab[3,i+9] := Ax_Q_TheoreticalSpeedMicroSample (1,9);  
  
        Tab[4,i] := Ax_Q_SpeedMicroSample (1,2);  
        Tab[4,i+1] := Ax_Q_SpeedMicroSample (1,3);  
        Etc..  
        Tab[4,i+9] := Ax_Q_SpeedMicroSample (1,1);  
  
        Tab[5,i] := Ax_Q_TheoreticalTorqueMicroSample (1,0);  
        Tab[5,i+1] := Ax_Q_TheoreticalTorqueMicroSample (1,1);  
        Etc..  
        Tab[5,i+9] := Ax_Q_TheoreticalTorqueMicroSample (1,9);  
  
        Tab[6,i] := Ax_Q_TorqueMicroSample (1,2);  
        Tab[6,i+1] := Ax_Q_TorqueMicroSample (1,3);  
        Etc..  
        Tab[6,i+9] := Ax_Q_TorqueMicroSample (1,1);  
        Etc..  
  
      END ;  
      Etc.. Pour les autres axes et le 2ème tableau  
    END ;  
  END ;  
END ;
```

ANNEXE 4

Programme de traitement Matlab

- DELTA.M
- NEW_ACQUI.M
- OLD_ACQUI.M
- CARTO.M
- SECTORTABRAD100.M
- SEARCHPARAM.M
- TRAJNAME.M

**CERTAINS PROGRAMMES NE SONT PAS REPRÉSENTÉS EN ENTIER DE PAR LEUR TAILLE.
LES CODES COMPLETS SONT DISPONIBLES EN ANNEXE SUR LE CD.**

Aperçu du code d'acquisition du Robot Delta

« Delta.m » Version 2.0 : 88 Lignes

ENTÊTE : Ligne 1

```
%Accueil programme
clc;
disp(' _____');
disp(' | GESTION DE DONNEES DU ROBOT DELTA |');
disp(' | Logiciel integre basé sur MATLAB |');
disp(' | _____');
disp(' | Auteurs : Briguët Michael |');
disp(' | Version : 2.0 |');
disp(' | Sion, le 05 Juillet 2010 |');
disp(' | _____');
disp(' ');
disp(' VEUILLEZ SIGNALER TOUT DEFAUT AUX AUTEURS SVP !');
disp(' ');
%Boucle du pogramme
choix1 = 100;
while choix1 > 0,

%affichage du menu
disp(' _____');
disp(' MENU PRINCIPAL |');
disp(' _____');
disp(' 0 : Quitter |');
disp(' -----');
disp(' 1 : Nouvelle acquisition |');
disp(' 2 : Ancienne acquisition |');
disp(' -----');
disp(' 3 : Cartographie d'une série de test |');
disp(' -----');
disp(' 4 : Informations sur le programme |');
disp(' _____');

%boucle d'interrogation
choix1 = 100;
while ((choix1 < 0) || (choix1 > 4)),
    choix1 = input(' Votre reponse SVP : ');
    if isempty(choix1),
        choix1=100;
    end
end; %fin du while d'interrogation
if choix1 == 0,
    clc;
    disp('SORTIE du Programme Delta...');
    disp('Pour le relancer introduisez : Delta "RETURN"');
%Nouvelle acquisistion
elseif choix1 == 1,
    choix2 = 1;
    New_Acqui;
Etc...
```


Aperçu du code d'acquisition du Robot Delta

« New_Acqui.m » Version 2.0 : 1328

ENTÊTE : Ligne 1

```
%-----  
% New_Acqui.m  
%  
% Nouvelle Acquisition de données du Robot Delta  
% -----  
% Version : 2.0  
% Auteurs : Briguet Michael(V2.0)  
% Sion, le 05 Juillet 2010  
%-----  
  
%Boucle du programme  
choix2=100;  
while choix2 > 0,  
  
    Affich=100;  
    %Effacement de l'écran  
    clc;  
    %interrogation de l'utilisateur  
    %affichage du menu du choix de l'affichage  
    disp('_____');  
    disp('          NOUVELLE ACQUISITION          ');  
    disp('_____choix de l''affichage_____');  
    disp('1 : Axes 1,2,3 : Position, Vitesse, Erreurs      : 1');  
    disp('2 : Axes 1,2,3 : Position, Vitesse, Erreurs, Variances : 2');  
    disp('3 : Axes 1,2,3 : Variances et Erreurs de vitesse (echant.) : 3');  
    disp('4 : Axes 1,2,3 : Variances uniquement (+ rapide)      : 4');  
    disp('_____')  
    while ((Affich < 1) | (Affich > 4)),  
        Affich = input('votre reponse SVP : ');  
    end;  
  
    VarType = 100;  
    %Choix du type de variance  
    if Affich ~= 1,  
        clc;  
        disp('_____');  
        disp('          NOUVELLE ACQUISITION          ');  
        disp('_____Type de Variance_____');  
        disp('1 : Maximum de la 2ème dérivée de la Variance la plus élevée ');  
        disp(' ');  
        disp('2 : Plateau de Variance limite par défaut (150000 [mm2/s2]) ');  
        disp(' ');  
        disp('3 : Plateau de Variance limite personnalisée');  
        disp(' ');  
        disp('_____');  
  
        while ((VarType < 1) | (VarType > 3)),
```

```
VarType = input('votre reponse SVP : ');
end;

%Variance plateau par défaut
VarPlat=150000;

%Choix à l'utilisateur de la variange plateau
if VarType == 3,
    disp('_____Plateau de Variance limite personnalisé_____');
    disp('Veuillez entrer une valeur de variance limite : [mm2/s2]');
    disp('Par exemple: 150000');
    VarPlatStr = input('VarPlat: ');
    Varplat = str2double(VarPlatStr);
end;
end;
Etc..
```

Récupération de données des fichiers « DATA.INT », « INFOTIME.INT » et « INFOPARA.INT » Ligne 70

```
%Debut de l'acquisition des données
disp('_____');
disp(' ACQUISITION DES DONNEES EN COURS..... ');
disp('_____');

%Variable de choix de Delta.m pour retour à la page principale
choix1=100 ;

%Effacement des données d'un éventuel essai précédent en cache
clear tab_final startendstep TabVar1 TabVar2 TabVar3 TabAccel1 TabAccel2 TabAccel3
clear x_exp x_mant x_neg x_out deriv11 deriv12 deriv21 deriv22 deriv31 deriv32
clear derivtot1 derivtot2 ThSpeedMSample1 ThSpeedMSample2 ThSpeedMSample3
clear TabVartot SpeedMSample1 SpeedMSample2 SpeedMSample3 Diff1 Diff2 Diff3 x NbStep

%Décodage des données c_realtype
fid = fopen('D:\TRANS\DATSAVED\DATA.INT');
x = fread(fid,Inf,'uint16');
x = flipr(reshape(x,5,numel(x)/5));
x_neg = x(:,1)>32767;
x_exp = bitand(x(:,1),hex2dec('7FFF'))-16383;
Q = 1.0/65536.0;
x_mant = 2.0*Q*(x(:,2)+Q*(x(:,3)+Q*(x(:,4)+Q*(x(:,5)))));
fclose(fid);

x_out = x_mant.*2.^x_exp;
x_out(x_neg) = -x_out(x_neg);

%Décodage des données de temps
fid2 = fopen('D:\TRANS\DATSAVED\INFOTIME.INT');
InfoDateTime = fread(fid2,Inf,'uint16');
fclose(fid2);

%Decodage des données c_realtype des paramètres
fid3 = fopen('D:\TRANS\DATSAVED\INFOPARA.INT');
```

```
InfoParam = fread(fid3,Inf,'uint16');
InfoParam = fliplr(reshape(InfoParam,5,numel(InfoParam)/5));
InfoParam_neg = InfoParam(:,1)>32767;
InfoParam_exp = bitand(InfoParam(:,1),hex2dec('7FFF'))-16383;
Q = 1.0/65536.0;
InfoParam_mant = 2.0*Q*(InfoParam(:,2)+Q*(InfoParam(:,3)+Q*(InfoParam(:,4)+Q*(InfoParam(:,5))));
fclose(fid3);

InfoParam_out = InfoParam_mant.*2.^InfoParam_exp;
InfoParam_out(InfoParam_neg) = -InfoParam_out(InfoParam_neg);

%Mettre les données dans le vecteur 'x_out'
%dans un tableau de 18 colonnes
for p=1:numel(x_out)
    %Détermination de l'index des colonnes
    %Correction si p est un multiple de 100
    if (p/100 - fix(p/100) == 0.0)
        correc = 1;
    else
        correc = 0;
    end
    %correction si pe est un multiple de 1800
    if (p/1800 - fix(p/1800) == 0.0)
        correc2 = 1;
    else
        correc2 = 0;
    end
    c = fix(p/100)+(1-correc)-(fix(p/1800)-correc2)*18; % 'c' is the column's index

    %Détermination de l'index des lignes
    l_prim = p - fix(p/100)*100;
    if (l_prim == 0)
        l_prim = 100;
    end
    l = l_prim + fix(p/1800)*100;

    %correction afin que les valeurs théoriques concordes
    %avec les valeurs mesurées
    if l < 18
        if c==2 | c==4 | c==6 | c==8 | c==10 | c==12 | c==14 | c==16 | c==18
            l = l+18;
        end
    elseif l >= 18
        if c==2 | c==4 | c==6 | c==8 | c==10 | c==12 | c==14 | c==16 | c==18
            l = l-17;
        end
    end

    %Les valeurs sont mise dans un tableau de 18 colonnes
    tab_final(l,c) = x_out(p,1);
end

%Assignement des variables de temps et paramètres du mouvement
Jour = int2str(InfoDateTime(1));
Mois = int2str(InfoDateTime(2));
```

```
Annee = int2str(InfoDateTime(3));
Heure = int2str(InfoDateTime(4));
Minute = int2str(InfoDateTime(5));
Seconde = int2str(InfoDateTime(6));
HeureFin = int2str(InfoDateTime(7));
MinuteFin = int2str(InfoDateTime(8));
SecondeFin = int2str(InfoDateTime(9));
nbreCycles = int2str(InfoDateTime(10));
trajectoire = (InfoDateTime(11));

AccelVal = int2str(InfoParam_out(1));
SpeedVal = int2str(InfoParam_out(2));
Start_X = int2str(InfoParam_out(3));
Start_Y = int2str(InfoParam_out(4));
Start_Z = int2str(InfoParam_out(5));
End_X = int2str(InfoParam_out(6));
End_Y = int2str(InfoParam_out(7));
End_Z = int2str(InfoParam_out(8));
NbStep = (InfoParam_out(10));
AccelMin = (InfoParam_out(11));
AccelMax = (InfoParam_out(12));
AccelStep = (InfoParam_out(13));
PrepareTargetTime = (InfoParam_out(14));

%Conversion des entier en string
trajectoireStr = int2Str(trajectoire);
AccelMinStr = int2str(AccelMin);
AccelMaxStr = int2str(AccelMax);
AccelStepStr = int2str(AccelStep);

%Interprétation du code de mouvement avec la fonction "trajName"
[locName1 locName2] = trajName(trajectoire);
```

Etc...

Analyse de la variance de l'axe 1 : Ligne 198

```
%-----
% Détermination de la variance de l'axe 1
%-----
if Affich == 2 || Affich == 3 || Affich == 4,
    %Titre de la fenêtre
    figure('numbertitle','off','name','Var SpdE','OuterPosition',[100,100, 1100, 900]);
    %Recherche des valeurs dans le tableau
    ThSpeedMSample1 = tab_final(:,3);
    SpeedMSample1 = tab_final(:,4);
    %Erreur de vitesse
    Diff1 = (ThSpeedMSample1 - SpeedMSample1);
    t = 1 ;
    %Boucle d'échantillonnage
    for Step=0 : NbStep-1,
        next=0;
        %Boucle de recherche de début de mouvement
        while next==0;
            %Condition du début de l'échantillonnage
```

```

if ( (t == 1) || (((abs(Diff1(t-3)))+(abs(Diff1(t-2)))+(abs(Diff1(t-1)))<=5) && Diff1(t)~=0 && ((abs(Diff1(t+1)) +
abs(Diff1(t+2)) + abs(Diff1(t+3)) + abs(Diff1(t+4)) + abs(Diff1(t+5)) + abs(Diff1(t+6)) + abs(Diff1(t+7)))>150) ))
    %Début de l'échantillonnage
    startendstep(1,(Step+1)) = t;
    %Fin de l'échantillonnage
    startendstep(2,(Step+1)) = t+350;
    %Condition si l'échantillonnage dépasse la valeur max du tableau
    if startendstep(2,(Step+1)) > l,
        endstep = l ;
    end;
    %Définition de la variance
    Variance1 = var(Diff1(startendstep(1,(Step+1)):startendstep(2,(Step+1))));
    next=1;
end;
%Incrémentement de la boucle d'échantillonnage
t = t+1;
if t >= (l-2)
    t = l-2;
end;
end;
%Assignment dans un tableau de la variance correspondante
TabVar1(Step+1)=Variance1;
%Assignment dans un tableau de l'accélération liée à la variance
TabAccel1(Step+1)=(AccelMin+AccelStep*Step);
%Recherche du mouvement suivant
t=t+350;
end;
%Plot de la courbe d'échantillonnage
plot(TabAccel1,TabVar1,'-g');
%Maintien du plot pour les autres courbes
hold
%Variance critique avec la méthode de la 2ème dérivée
if VarType > 0,
    %Dérivée première et seconde du tableau de variance
    deriv11 = gradient(TabVar1);
    deriv12 = gradient(deriv11);
    %Détermination du maximum de la deuxième dérivée
    [yd1,vd1]=max(deriv12);
    %Assignment des valeurs correspondantes
    AccelCritAxe1=TabAccel1(vd1-1);
    VarCritAxe1=TabVar1(vd1-1);
end;
%Variance critique avec la méthode de plateau
if VarType == 2 || VarType ==3,
    v=1;
    okout=0;
    %Boucle de recherche de variance maximum
    while okout == 0
        if TabVar1(v) > VarPlat,
            AccelCritAxe1 = TabAccel1(v-1);
            VarCritAxe1 = TabVar1(v-1);
            okout = 1;
        end;
        %Si la variance maximum n'a pas atteint la valeur plateau
        if v == NbStep && okout == 0;

```

```
    AccelCritAxe1 = TabAccel1(v);
    VarCritAxe1 = TabVar1(v);
    okout = 1;
end;
v = v+1;
end;
end;

%Plot de lignes qui correspondent aux valeurs critiques trouvées
plot([AccelCritAxe1 AccelCritAxe1],[0 VarCritAxe1],'-g')
plot([AccelMin AccelCritAxe1],[VarCritAxe1 VarCritAxe1],'-g')

%Labels sur les axes
ylabel('e Variance [mm^2/s^2]');
xlabel('Acceleration [mm/s^2]');

%Titre du graphique
title(['VARIANCE' ' Le ',Jour,'.',Mois,'.',Annee,' de ',Heure,'h',Minute,'min',Seconde,'s',' à '
HeureFin,'h',MinuteFin,'min',SecondeFin,'s']);
['Accél.Min/Max =',AccelMinStr,'/',AccelMaxStr,'mm/s^2 Step=',AccelStepStr,'mm/s^2 Vit.=',SpeedVal,'mm/s']
['Code de mouvement ',trajectoireStr,' Start:',locName1,' X:',Start_X,' Y:',Start_Y,' Z:',Start_Z,' / End ; ',locName2,'
X:',End_X,' Y:',End_Y,' Z:',End_Z,]};
```

Axe critique Ligne 524

```
%-----
% Détermination de l'axe critique
%-----
if ((VarCritAxe1>= VarCritAxe2) && (VarCritAxe1>= VarCritAxe3)),
    AccelCrit = AccelCritAxe1;
    AxeCrit = 1;
    VarCrit = VarCritAxe1;
elseif ((VarCritAxe2>= VarCritAxe1)&& (VarCritAxe2>= VarCritAxe3)),
    AccelCrit = AccelCritAxe2;
    AxeCrit = 2;
    VarCrit = VarCritAxe2;
else AccelCrit = AccelCritAxe3;
    AxeCrit = 3;
    VarCrit = VarCritAxe3;
end;
%Fin de l'affichage du graphique de variance
end;
```

Erreur de Vitesse de l'axe 1 : Ligne 625

```
%-----
%Erreur de vitesse de l'axe 1
%-----
if Affich ==1 || Affich ==2 || Affich == 3,
    %Titre de la fenêtr
    figure('numbertitle','off','name','1 SpdE','OuterPosition',[100,100, 1100, 900]);
    %Recherche des valeurs dans le tableau
    ThSpeedMSample1 = tab_final(:,3);
    SpeedMSample1 = tab_final(:,4);
    %Plot de l'erreur de vitesse
    plot(ThSpeedMSample1 - SpeedMSample1);
    %Maintien du plot
    hold;
    %Plot de repères indiquant le début et la fin de l'échantillonnage
    %de la variance
    if Affich ==2 || Affich == 3,
        for n=1: NbStep,
            plot([startendstep(1,n) startendstep(1,n)],[-500 500],':g')
            plot([startendstep(2,n) startendstep(2,n)],[-500 500],':r')
        end;
    end;

    %Labels sur les axes
    xlabel('Time [ms]');
    ylabel('vitesse [mm/s]');

    %Titre du graphique
    title(['AXE 1 ERREUR VITESSE' ' Le ',Jour,' ',Mois,' ',Annee,...
        ' de ',Heure,'h',Minute,'min',Seconde,'s', ' à ' HeureFin,...
        'h',MinuteFin,'min',SecondeFin,'s'];
        ['Accél.Min/Max =',AccelMinStr,'/',AccelMaxStr,'mm/s^2 Step=',...
        AccelStepStr,'mm/s^2 Vit.=',SpeedVal,'mm/s']
        ['Code de mouvement : ',trajectoireStr,' Start : ',locName1,...
        ' X:',Start_X, ' Y:',Start_Y, ' Z:',Start_Z, ' / End : ',locName2,...
        ' X:',End_X, ' Y:',End_Y, ' Z:',End_Z,]);
    %Legendes associées au graphique
    legend('Axe1 Erreur Vit.','Location','NorthWest');
end;
```

Sauvegarde d'un fichier TXT : Ligne 1120

```
%Condition de sauvegarde de l'analyse
if choix2 == 2 || choix2 == 3 || choix2 == 4 || choix2 == 5,
    %Effacement de l'écran
    clc;
    %Variable résumant la date et l'heure
    ID = [Annee,'-',Mois,'-',Jour,'-',Heure,'h',Minute,'m',Seconde,'s'];
    %Gestion de donnée d'un essai unique
    if choix2==2 || choix2 ==3,
        %Creation du dossier propre a l'essai
        %s'il n'est pas deja existant
        mkdir('D:\Robot_Delta_INT_History',ID);
        %Création du fichier de résumé de l'essai
```

```
fichier=fullfile('D:\Robot_Delta_INT_History\',ID,'\',[ID,'.txt']);  
fid = fopen(fichier,'wt');  
SerieName = 'essai unique';  
end;
```

Sauvegarde des données : Ligne 1275

```
disp('_____');  
disp(' SAUVEGARDE DES DONNEES EN COURS..... (niveau 2)');  
disp('_____');  
%Transfert des fichiers de données sur le mouvement vers  
%vers le dossier adéquats si c'est un essai unique  
if choix2 == 2 || choix2 == 3;  
    %Copie des fichier.INT  
    copyfile('D:\TRANS\DATSAVED\DATA.INT',['D:\Robot_Delta_INT_History\',ID]);  
    copyfile('D:\TRANS\DATSAVED\INFOTIME.INT',['D:\Robot_Delta_INT_History\',ID]);  
    copyfile('D:\TRANS\DATSAVED\INFOPARA.INT',['D:\Robot_Delta_INT_History\',ID]);  
    %Changement de nom  
    movefile(['D:\Robot_Delta_INT_History\',Annee,'-',Mois,'-',Jour,'-  
,Heure,'h',Minute,'m',Seconde,'s\DATA.INT'],['D:\Robot_Delta_INT_History\',Annee,'-',Mois,'-',Jour,'-  
,Heure,'h',Minute,'m',Seconde,'s\,Annee,'-',Mois,'-',Jour,'-',Heure,'h',Minute,'m',Seconde,'s_DATA.INT']);  
    movefile(['D:\Robot_Delta_INT_History\',Annee,'-',Mois,'-',Jour,'-  
,Heure,'h',Minute,'m',Seconde,'s\INFOTIME.INT'],['D:\Robot_Delta_INT_History\',Annee,'-',Mois,'-',Jour,'-  
,Heure,'h',Minute,'m',Seconde,'s\,Annee,'-',Mois,'-',Jour,'-',Heure,'h',Minute,'m',Seconde,'s_INFOTIME.INT']);  
    movefile(['D:\Robot_Delta_INT_History\',Annee,'-',Mois,'-',Jour,'-  
,Heure,'h',Minute,'m',Seconde,'s\INFOPARA.INT'],['D:\Robot_Delta_INT_History\',Annee,'-',Mois,'-',Jour,'-  
,Heure,'h',Minute,'m',Seconde,'s\,Annee,'-',Mois,'-',Jour,'-',Heure,'h',Minute,'m',Seconde,'s_INFOPARA.INT']);  
    end;  
    %Si c'est un essais issu d'une série  
    if choix2 == 4 || choix2 == 5;  
        %Copie des fichier.INT  
        copyfile('D:\TRANS\DATSAVED\DATA.INT',['D:\Robot_Delta_INT_History\',SerieName]);  
        copyfile('D:\TRANS\DATSAVED\INFOTIME.INT',['D:\Robot_Delta_INT_History\',SerieName]);  
        copyfile('D:\TRANS\DATSAVED\INFOPARA.INT',['D:\Robot_Delta_INT_History\',SerieName]);  
        %Changement de nom  
        movefile(['D:\Robot_Delta_INT_History\',SerieName,\DATA.INT'],  
['D:\Robot_Delta_INT_History\',SerieName,'\',[trajectoireStr,'_DATA.INT']]);  
  
        movefile(['D:\Robot_Delta_INT_History\',SerieName,\INFOTIME.INT'],['D:\Robot_Delta_INT_History\',SerieName,'\',[trajec  
toireStr,'_INFOTIME.INT']]);  
  
        movefile(['D:\Robot_Delta_INT_History\',SerieName,\INFOPARA.INT'],['D:\Robot_Delta_INT_History\',SerieName,'\',[trajec  
toireStr,'_INFOPARA.INT']]);  
    end;
```


Aperçu du code d'acquisition du Robot Delta

« Old_Acqui.m » Version 2.0 : 1410 Lignes

Début de programme différent de New_Acqui : Ligne 1

```
%-----  
% Old_Acqui.m  
%  
% Retraitement d'une acquisition de données du Robot Delta  
% -----  
% Version : 2.0  
% Auteurs : Briguet Michael(V2.0)  
% Sion, le 05 Juillet 2010  
%-----  
  
choix3 = 100;  
  
%interrogation de l'utilisateur sur le type d'essai à retraiter  
clc;  
disp('_____');  
disp(' ACQUISITION D UNE ARCHIVE (niveau 2)');  
disp('_____');  
disp('0 : Revenir au menu principal ');  
disp('1 : Essai unique ');  
disp('2 : Essai de variance unique ');  
disp('3 : Essai de variance issue d"une série ');  
disp('_____');  
while ((choix3 < 0) || (choix3 > 3)),  
choix3 = input('votre reponse SVP : ');  
end;  
%Effacement de l'écran  
clc;  
  
%Traitement d'un essai unique  
if choix3 == 1 || choix3 == 2,  
clc;  
%Affichage des essais trouvés  
disp('_____');  
disp('Lecture d"un modèle sauvé au préalable :');  
disp('Liste des modèles trouvés dans le répertoire de travail :');  
com='dir '  
commande=strcat('dir D:\Robot_Delta_INT_History\');  
eval(commande);  
disp(' ');  
%Demande à l'utilisateur le nom de l'essai à retraiter  
dossier=input('Nom de la mesure ? : ','s');  
%Contrôle si l'essai existe ou non  
if exist(['D:\Robot_Delta_INT_History\',dossier]) == 7,  
traj = dossier;  
clc;  
disp('FICHER TROUVE !')  
%Affichage du fichier .TXT lié a l'essai par l'intermédiaire
```

```
%d'un message popup
disp('Affichage des commentaires associés au fichier');
fichier=fullfile('D:\Robot_Delta_INT_History\',dossier,'\',[traj,'.txt']);
popupmessage(fichier,dossier);

else
    %Retour au début du programme si le fichier n'est pas trouvé
    disp('FICHIER NON TROUVE !');
    pause(1);
    Old_Acqui;
end;
end;

%Traitement d'un essai de variance unique
if choix3 == 3,
    clc;
    %Affichage des essais trouvés
    disp('_____');
    disp('Lecture d'un fichier d'une Série :');
    disp('Liste des modèles trouvés dans le répertoire de travail :');
    com='dir ';
    commande=strcat('dir D:\Robot_Delta_INT_History\');
    eval(commande);
    disp(' ');
    %Demande à l'utilisateur le nom de la série à retraiter
    dossier=input('Nom de la Série ? : ','s');
    %Contrôle si la série existe ou non
    if exist(['D:\Robot_Delta_INT_History\',dossier]) == 7,
        clc;
        disp('SERIE TROUVEE !');
    else
        %si la série n'existe pas, retour au début du programme
        disp('SERIE NON TROUVE !');
        pause(1);
        Old_Acqui;
    end;

    clc;
    %Si la série existe on demande quel essai de la série faut-t-il traiter
    disp('_____');
    disp('Lecture d'un fichier de série :');
    disp('Liste des trajectoires trouvés dans le répertoire de la série :');
    com='dir ';
    commande=strcat('dir D:\Robot_Delta_INT_History\',dossier);
    eval(commande);
    disp(' ');
    %Demande à l'utilisateur le nom de l'essai issu
    %de la série à retraiter
    traj=input('Numéro de trajectoire ? : ','s');

    if exist(['D:\Robot_Delta_INT_History\',dossier,'\',[traj,'.txt']]) == 2,
        clc;
        disp('FICHIER TROUVE !');
        %Contrôle si l'essai existe
        disp('Affichage des commentaires associés au fichier');
```

```
fichier=fullfile('D:\Robot_Delta_INT_History\',dossier,'\',[traj,'.txt']);  
popupmessage(fichier,traj);  
else  
%Si l'essai n'existe pas, retour au début du programme  
disp('FICHER NON TROUVE !');  
pause(1);  
Old_Acqui;  
end;  
end;
```

[ETC .. Idem New_Acqui.m](#)

Aperçu du code d'acquisition du Robot Delta

« Carto.m » Version 2.0 : 705 Lignes

ENTÊTE : Ligne 1

```
%-----  
% Carto.m  
%  
% Création de cartographie  
% -----  
% Version : 2.0  
% Auteurs : Briguët Michael(V2.0)  
% Sion, le 05 Juillet 2010  
%-----  
%  
%Effacement de l'écran  
clc;  
%Demande à l'utilisateur la série à cartographier  
disp('Veuillez entrer le nom de la Série à représenter');  
disp('_____')  
SerieNameStr=input('Numéro de la Série : ' , 's');  
%Conversion du numéro de la série en entier  
SerieNameNum = str2num(SerieNameStr);  
  
choixC=100;  
choixP=100;  
  
%test automatique du type de série  
test264 = exist(['D:\Robot_Delta_INT_History\',SerieNameStr,'\264.txt']);  
test265 = exist(['D:\Robot_Delta_INT_History\',SerieNameStr,'\265.txt']);  
test345 = exist(['D:\Robot_Delta_INT_History\',SerieNameStr,'\345.txt']);  
SerieType = 0;  
%Série radiale ancien format (64 points de mesures)  
if test265 == 0 && test345 == 0 && test264 == 2,  
    SerieType = 1;  
    SerieTypeStr = 'Série radiale 6*64 points de mesures';  
%Série radiale nouveau format (100points de mesures)  
elseif test265 == 2,  
    SerieType = 2;  
    SerieTypeStr = 'Série radiale 6*100 points de mesures';  
%Série Tangentielle standard (45 points de mesures)  
elseif test345 == 2,  
    SerieType = 3;  
    SerieTypeStr = 'Série tangentielle 6*45 points de mesures';  
%Retour en début de programme si la série est introuvable  
else  
    clc;  
    disp('Lecture de la série impossible');  
    pause(1);  
    clc;  
    choixC=0;  
end
```

Analyse de l'accélération critique et Récupération du tableau de secteur « TabSector » : Ligne 76

```
%Choix du type de graphique
if choixP > 0,
    clc;
    disp('_____');
    disp('      PLOT DES CARTOGRAPHIES      ');
    disp('_____Type de Cartographie_____');
    switch choixP

        %Analyse de l'accélération critique déduite
        case 1,
            param = 'ACCELERATION CRITIQUE DEDUITE = [mm/s2] ';
            labelZ = 'Acceleration critique [mm/s^2]';
            %Retour du secteur analysé en fonction du type de série
            switch SerieType
                case 1
                    TabSector = SectorTabRad64(SerieNameNum,param);
                case 2
                    TabSector = SectorTabRad100(SerieNameNum,param);
                case 3
                    TabSector = SectorTabTan45(SerieNameNum,param);
            end
            AnalyzeName = 'AccelCrit';
            %Calcul et affichage de la moyenne
            meanStr = int2str(mean(mean(TabSector)));
            disp(' ');
            disp(sprintf('%s%s%s','L'accélération moyenne critique est de ',meanStr,' [mm/s^2]'));
            disp('_____');
```

ETC...

Choix du type de cartographie : Ligne 351

```
Suite du menu de choix de type de cartographie
disp('0 Retour au menu principal ');
disp('1 Plot mesh ');
disp('2 Plot surf ');
disp('3 Plot surfn ');
if choixP ~=3,
    disp('4 Plot contour ');
end;
disp('_____');
while ((choixC < 0) | (choixC > 4)),
    choixC = input('votre reponse SVP : ');
end;
%Attribution du type de cartographie
switch choixC
    case 1,
        type = 'mesh';
    case 2,
        type = 'surf';
    case 3,
        type = 'surfn';
    case 4,
```

```
    type ='contour';  
end
```

Affichage secteur par secteur de la cartographie : Ligne 377

```
%Plot des cartographie en mesh, surf et surfn  
if choixC >0 && choixC <=3,  
  
    %Titre de la cartographie  
    figure('numbertitle','off','name',['Carto Série:',SerieNameStr,' ',AnalyzeName,' ',type,], 'OuterPosition',[100,100, 1100,  
    900]);  
    %Calcul de la valeur maximum de la cartographie  
    MaxZ=max(max(TabSector));  
  
    %Représentation des axes du robot au dessus de la cartographie  
    %Pour l'affichage de l'axe défavorable  
    if choixP ==3,  
        drawLine([[300;0;7] [0;0;5]], '-g2');  
        drawLine([[ -150;-259;7] [0;0;5]], '-b2');  
        drawLine([[ -150;259;7] [0;0;5]], '-r2');  
    %Pour les autres plots  
    else  
        drawLine([[300;0;MaxZ+MaxZ*0.15] [0;0;MaxZ+MaxZ*0.1]], '-g2');  
        drawLine([[ -150;-259;MaxZ+MaxZ*0.15] [0;0;MaxZ+MaxZ*0.1]], '-b2');  
        drawLine([[ -150;259;MaxZ+MaxZ*0.15] [0;0;MaxZ+MaxZ*0.1]], '-r2');  
    end  
    %Maintien du plot  
    hold on  
    %Conversion pour le plot suivant du numéro des axes  
    %Pour l'affichage de l'axe le plus défavorable  
    if choixP ==3 ,  
        for a=1:10  
            for b=1:10  
                switch TabSector(a,b),  
                    case 1,  
                        TabSector(a,b) = 2;  
                    case 2,  
                        TabSector(a,b) = 3;  
                    case 3,  
                        TabSector(a,b) = 1;  
                    otherwise  
                        TabSector(a,b) = 3;  
                end;  
            end;  
        end;  
    end;  
  
    %Plot 3D du secteur de 0 à 60°  
    polarplot3d(TabSector,'plottype',type,...  
    'polargrid',[0 0],'tickspacing',15,...  
    'angularrange',-[0 60]*pi/180,'radialrange',270,...  
    'InterpMethod','linear');  
    %Maintien du plot
```

ETC.. Idem pour les autres secteurs

Paramètre de la perspective du graphique : Ligne 659

```
%Mise a l'échelle automatique selon la valeur maximum
%Avec limitation des axes et repères
set(gca,'DataAspectRatio',[1 1 2*MaxZ/300],'View',[-150 30],...
'Xlim', [-300 300], 'Ylim', [-300 300],...
'XTick',[-300 -150 0 150 300],'YTick',[-300 -150 0 150 300]);

%Taille des lignes
h = findobj('Type','patch');
set(h,'LineWidth',2)

%Grille activée
grid on

%Barre de couleur activée
colorbar;

%Affichage des légendes
legend('Axe1','Axe2','Axe3');

%Affichage des labels
xlabel('Axe X [mm]');
ylabel('Axe Y [mm]');
zlabel(labelZ);
end;

%Titre affiché sur la cartographie
title(['',SerieTypeStr,' plot mode : ',type];
['Numéro de la cartographie : ',SerieNameStr,' Variable : ',labelZ,]);
%couleurs de la map adaptées à l'accélération ou la variance
if choixP == 1 || (choixP >= 4 && choixP <= 8),
    caxis([0 1e+006]);
    load('ColorMapAccel','mycmap');
    set(gcf,'Colormap',mycmap);
elseif choixP == 2 || (choixP >= 9 && choixP <= 13),
    caxis([0 2e+006]);
    load('ColorMapVar','mycmap');
    set(gcf,'Colormap',mycmap);
end
```

Aperçu du code d'acquisition du Robot Delta

« SectorTabRad100.m » Version 2.0 : 189 Lignes

ENTÊTE : Ligne 1

```
%-----  
% SectorTabRad100.m  
%  
% Création d'un tableau de 10 par 10 a partir de la série  
% de mesure Radiale de 100points.  
% -----  
% Version : 2.0  
% Auteurs : Briguet Michael(V2.0)  
% Sion, le 05 Juillet 2010  
%-----  
  
function [TabCarto] = SectorTabRad100(SerieNameNum,param)  
%SerieNameNum correspond au numéro de la série à analyser  
%Param correspond au paramètre recherché dans les fichiers TXT  
%des esasis de la série,par ex. : 'VARIANCE CRITIQUE DEDUITE = [mm2/s2] '
```

Recherche dans les fichiers TXT du paramètre à analyse : Ligne 17

```
%Conversion du numéro de série entier en string  
SerieNameStr = int2str(SerieNameNum);  
  
%Recherche du paramètre a analyser dans les fichiers TXT  
%et stockage de celui-ci dans le tableau Tabline.  
  
%Boucle de recherche correspondant au nombre de fichiers  
for a=201:300,  
  
    %Conversion de l'index numérique en string  
    aStr = int2str(a);  
    %ouverture du fichier txt correspondant  
    fichierCarto=fullfile('D:\Robot_Delta_INT_History\',SerieNameStr,'\',[aStr,'.txt']);  
    fid = fopen(fichierCarto,'r');  
  
    %Recherche du nom de paramètre correspondant  
    %avec le compteur de ligne k  
    k=0;  
    x=0;  
    while x==0  
        tline = fgetl(fid);  
        if strcmp(tline ,param)==1,  
            x=1;  
        end  
    k=k+1;  
end  
%Fermeture du fichier TXT  
fclose(fid);
```



```
%Réouverture du fichier TXT
fichierCarto=fullfile('D:\Robot_Delta_INT_History\',SerieNameStr,'\',[aStr,'.txt']);
fid = fopen(fichierCarto,'r');

%Recherche de la valeur du paramètre situé après le nom (après k)
for b=0:k
    tline = fgetl(fid);
end

%Remplissage du tableau tabline et conversion de la valeur
%du paramètre en entier
b=a-200;
tlineNum = str2num(tline);
Tabline(b)=tlineNum;
fclose(fid);

end;
```

Remplissage du Tableau TabSector : Ligne 65

```
%Remplissage de la première ligne du tableau (valeurs au
%centre du plan de travail)
TabCarto(1,1)=Tabline(1);
TabCarto(1,2)=Tabline(2);
TabCarto(1,3)=Tabline(3);
TabCarto(1,4)=Tabline(4);
TabCarto(1,5)=Tabline(5);
TabCarto(1,6)=Tabline(6);
TabCarto(1,7)=Tabline(7);
TabCarto(1,8)=Tabline(8);
TabCarto(1,9)=Tabline(9);
TabCarto(1,10)=Tabline(10);
```

Etc...

```
%Remplissage de la dixième ligne du tableau
%situé en bordure du plan de travail
TabCarto(10,1)=Tabline(91);
TabCarto(10,2)=Tabline(92);
TabCarto(10,3)=Tabline(93);
TabCarto(10,4)=Tabline(94);
TabCarto(10,5)=Tabline(95);
TabCarto(10,6)=Tabline(96);
TabCarto(10,7)=Tabline(97);
TabCarto(10,8)=Tabline(98);
TabCarto(10,9)=Tabline(99);
TabCarto(10,10)=Tabline(100);

end
```

Aperçu du code d'acquisition du Robot Delta

« SearchParam.m » Version 2.0 : 65 Lignes

Code Complet

```
%-----  
% NOM DU PARAMETRE1 -  
% valeur du parametre1 -  
% -  
% NOM DU PARAMETRE2 -  
% valeur du parametre2 -  
% -  
%----- etc...  
% la valeur du paramètre est inscrite dans la variable tline.  
% Ce script recherche d'abord le nom du paramètre et  
% enregistre la ligne où il a été trouvée car a chaque  
% lecture, l'index interne a Matlab saute d'une ligne  
% la réouverture du fichier avec l'index de la ligne  
% enregistre automatiquement la valeur situé en desous  
% du nom de paramètre.  
% -----  
% Version : 2.0  
% Auteurs : Briguët Michael(V2.0)  
% Sion, le 05 Juillet 2010  
%-----  
  
% Adresse du fichier .TXT à analyser  
fichier=fullfile('D:\Robot_Delta_INT_History\2\201.txt');  
fid = fopen(fichier,'r');  
  
% variables de la boucle de recherche  
b=0;  
k=0;  
% boucle de recherche permettant d'analyser ligne par ligne  
while b==0  
  
    % acquisition de la ligne  
    tline = fgetl(fid);  
    % comparaison avec le nom du paramètre recherché  
    if strcmp(tline, 'ACCELERATION CRITIQUE DEDUITE = [mm/s2]')==1,  
        % variable de sortie de boucle  
        b=1;  
    end;  
  
    % variable de comptage de lignes  
    k=k+1;  
end  
fclose(fid);
```

```
% reouverture du fichier
fichier=fullfile('D:\Robot_Delta_INT_History\2\201.txt');
fid = fopen(fichier,'r');

% Récupération de la valeur
for b=0:k
    tline = fgetl(fid);
end
% Affichage de la valeur
disp(tline)
% Affichage du numéro de ligne
disp(k);

% fermeture du fichier.
fclose(fid);
```

Aperçu du code d'acquisition du Robot Delta

« TrajName.m » Version 2.0 : 65 Lignes

Entête et début du code

```
%-----  
% TrajName.m  
%  
% Retour des noms de début et de fin de mouvement  
% en fonction du code de trajectoire  
% -----  
% Version : 2.0  
% Auteurs : Briguet Michael(V2.0)  
% Sion, le 05 Juillet 2010  
%-----  
  
function [locName1 locName2] = TrajName( trajectoire )  
  
switch trajectoire  
  
    case 200,  
        locName1='Manuel';  
        locName2='Manuel';  
  
    case 201,  
        locName1='A0';  
        locName2='B0';  
  
    case 202,  
        locName1='A0';  
        locName2='B1';  
  
    case 203,  
        locName1='A0';  
        locName2='B2';  
  
    case 204,  
        locName1='A0';  
        locName2='B3';  
  
    Etc...  
  
    case 354,  
        locName1='K8';  
        locName2='K9';  
  
    otherwise  
        locName1='NULL';  
        locName2='NULL';  
  
end
```

ANNEXE 5

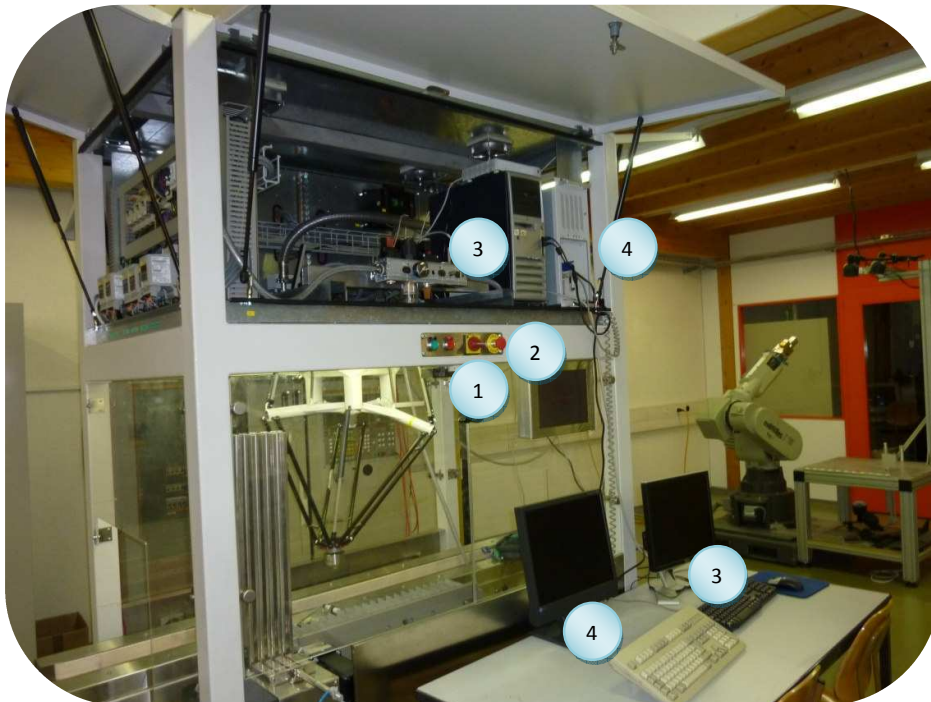
Mode d'emploi d'une acquisition

CE DOCUMENT EXPLIQUE LA MISE EN ROUTE DU ROBOT AINSI QUE LES PROCÉDURES À EFFECTUER POUR RÉALISER UNE ACQUISITION.

Mode d'emploi d'une acquisition sur le Robot Delta

Ce document résume rapidement la marche à suivre pour mettre le robot en fonction et réaliser une acquisition de données.

1. Mise en route du robot



- Activer l'interrupteur général (1).
- vérifier que le bouton d'arrêt d'urgence soit en position de repos (2).
- Allumer le PC Superviseur (3).
- Allumer le PC îlot (4).



2. Commandes DOS

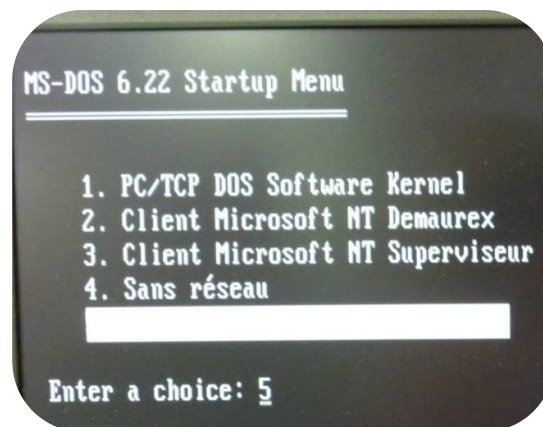
L'environnement DOS nécessite quelques commandes pour y naviguer :

- « cd nomdudossier » : Entrer dans un dossier.
- « cd.. » : Sortir d'un dossier
- « dir » : Afficher la contenance du dossier.
- « dir /p » : Affiche la contenance du dossier écran par écran.
- « Edit nomdufichier » : Ouvrir le fichier avec l'éditeur de texte
- « Del nomdufichier » : Effacer un fichier
- « copy nomdufichier destination » Copier un fichier ou un dossier
- « bp » : Lancement de Borland 7.0.

A noter que pour changer de clavier depuis le PC Îlot ou d'écran, il faut d'appuyer 3 fois sur CTRL.

3. Démarrage du PC Îlot

Ce PC fonctionnant sous Dos, le démarrage est un peu plus délicat. Dès l'enclenchement de la machine, il faut rester devant l'écran et atteindre le menu ci-dessous et sélectionner le choix 5 avant la fin du décompte du temps.



Le PC charge les données de la partition choisie. Il faut ensuite simplement appuyer sur « ENTER » et entrer le code « 1234 » puis « ENTER »

```
Microsoft DOS TCP/IP Protocol Driver 1.0a
Copyright (c) Microsoft Corporation, 1991. All rights reserved.
Copyright (c) Hewlett-Packard Corporation, 1985-1991. All rights reserved.
Copyright (c) 3Com Corporation, 1985-1991. All rights reserved.
Microsoft DOS TCP/IP NEMM Driver 1.0
The command completed successfully.
MS-DOS LAN Manager v2.1 Netbind
Fast Ethernet board's Ethernet Address:0048545A7784
Microsoft DOS TCP/IP 1.0a
MOT DE PASSE : 1234
The command completed successfully.
Type your user name, or press ENTER if it is TRANS:
Type your password:****_
```

Le programme arrive ensuite sur « C:\BRI\JOB\ », le dossier par défaut. Le dossier d'accueil est paramétrable dans le fichier « CONFIG.SYS »

4. Lancement de Borland et du programme d'acquisition

Il suffit de taper « bp » dans le dossier où les fichiers « CARTRA1.PAS », « CARTRA2.PAS » ou « CARTAN.PAS » se trouvent ainsi que les bibliothèques nécessaires au fonctionnement

```
Error 85: The local device name is already in use.
C:\BRI\JOB>bp_
```

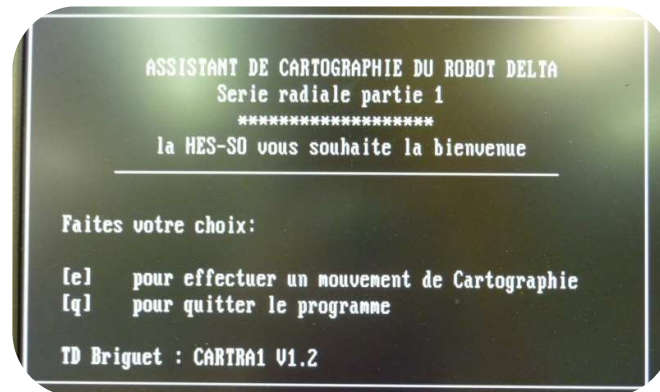
Borland 7.0 s'affiche alors, la touche « ALT+ nomdelamajuscule » permet de naviguer sur le menu principal au dessus et de choisir le fichier à ouvrir.

```
File Edit Search Run Compile Debug Tools Options Window Help
[.] CARTRA1.PAS
{S[A+,B+,D-,E-,X-,F-,G+,I+,L-,N+,P-,Q-,O-,R-,K-,S-,V-,T-,U+,W-}
{S[IFDEF DPMI] {S[C FIXED PRELOAD PERMANENT] {SENDIF}
{*****}
{ Programme de cartographie du plan de travail
{ HES_S0 SION
{ Project: Optimisation Robot Delta
{ File name : CartRal.pas
{
{ Designed by : Briguet Michael Date : 05.07.2010
{ Version U 1.2
```

Les touches « CTRL+F9 » Permet de compiler le programme et la touche « ALT+F9 » permet de le lancer.

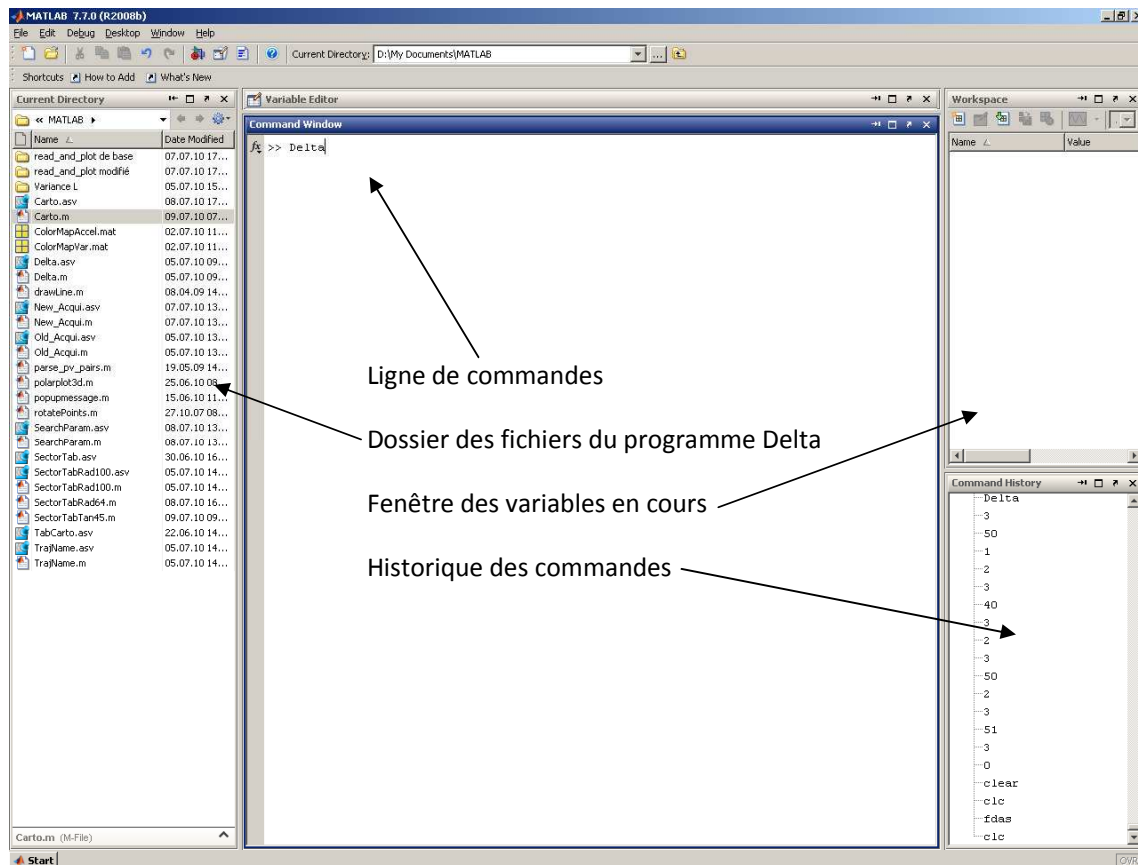
```
IP Compiling
Main file: CARTRA1.PAS
Done.
Target: Real Line number: 0
Free memory: 10837K Total lines: 3830
Compile successful:
```


Le programme se lance et on arrive sur l'écran d'accueil du programme d'acquisition. L'exemple sera développé avec « CARTRA1.PAS »



5. Ouverture de Matlab

Sur le PC Superviseur, il faut se loguer avec son nom d'utilisateur. Sinon en administrateur avec le log « Administrator » et le passe « GeminiSup ». Il suffit ensuite de lancer Matlab et son écran d'accueil apparaît. Il faut taper « Delta » pour lancer le programme



L'écran d'accueil de « Delta.m » apparaît, l'on peut procéder à un essai.

```

+-----+
| GESTION DE DONNEES DU ROBOT DELTA |
| Logiciel integre basé sur MATLAB |
+-----+
| Auteurs : Briguet Michael         |
| Version : 2.0                     |
| Sion, le 05 Juillet 2010          |
+-----+

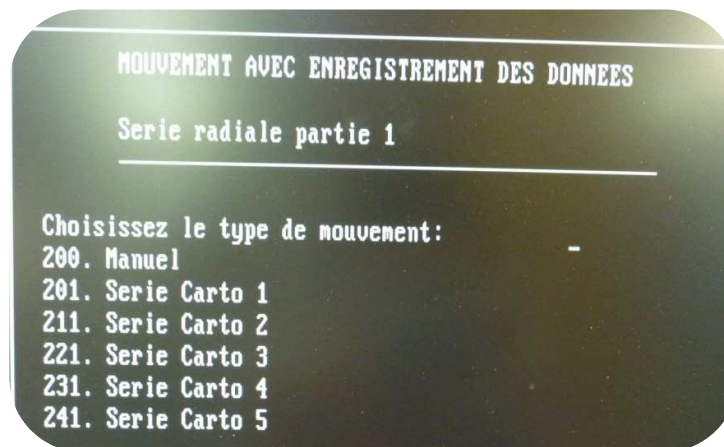
VEUILLEZ SIGNALER TOUT DEFAUT AUX AUTEURS SVP !

-----
MENU PRINCIPAL
-----
0 : Quitter
-----
1 : Nouvelle acquisition
2 : Ancienne acquisition
-----
3 : Cartographie d'une série de test
-----
4 : Informations sur le programme
-----
Votre reponse SVP : |

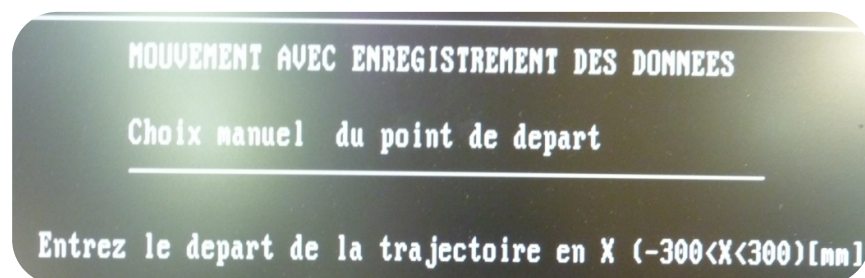
```

6. Paramétrage d'un essai sur le PC Îlot

Retour sur l'écran du PC Îlot, l'on peut choisir si l'on désire effectuer un mouvement avec paramétrage manuel des coordonnées du début et de la fin de la trajectoire, ou des mouvements avec l'assignation des coordonnées automatiques pour faire une cartographie.



Si l'on choisit le type manuel, les écrans suivant s'affichent.



Entrez le depart de la trajectoire en Y (-300<Y<300)[mm]

Entrez le depart de la trajectoire en Z (-550<X<-312)[mm]

Idem pour la fin de la trajectoire.

On arrive ensuite sur le menu de paramétrage du mouvement. Si l'on avait précédemment choisit un mouvement d'une série, l'on arriverait directement sur celui-ci.

```
NOUVEMENT POUR CARTOGRAPHIE
Mouvement Axial de A0 a B0
Code de trajectoire 201
Coordonnee exactes
Point de depart      Point d arrivee
0.000000000000000E+0000 : 3.000000000000000E+0001
0.000000000000000E+0000 : 0.000000000000000E+0000
-3.500000000000000E+0002 : -3.500000000000000E+0002

** Parametres actuels ****

Accel. de Depart : 10000 [mm/s2]
Accel. d Arrivee : 100000 [mm/s2]
Accel. Step      : 10000 [mm/s2]

*****

[p] pour modifier les parametres
[s] pour valider et debuter le mouvement
[q] pour annuler et quitter le programme
```

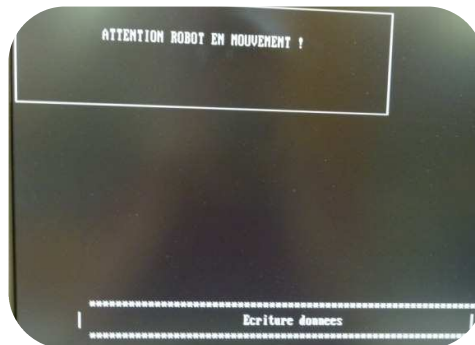
L'on peut modifier l'accélération de départ, d'arrivée et le step entre eux. Une demande à l'utilisateur.

```
NOUVEMENT AVEC ENREGISTREMENT DES DONNEES
Accel. de depart actuelle : 10000 [mm/s2]

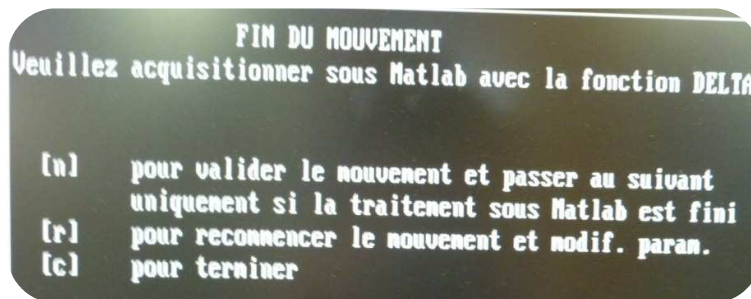
-----
Nouvelle Accel. de depart : _ [mm/s2]
```

7. Lancement du cycle de mouvements

Le cycle est lancé en appuyant sur « s ».



Le menu suivant indique la fin du mouvement, c'est là que le traitement de données commence sous Matlab.



8. Nouvelle acquisition sous Matlab

En choisissant sur le menu principal d'effectuer une nouvelle acquisition, l'on peut choisir les graphiques à afficher. Plus il y a de graphiques, plus le PC met de temps à les afficher. Le temps qu'ils apparaissent peut prendre 3-4 secondes.

```

NOUVELLE ACQUISITION
choix de l'affichage
1 : Axes 1,2,3 : Position, Vitesse, Erreurs           : 1
2 : Axes 1,2,3 : Position, Vitesse, Erreurs, Variances : 2
3 : Axes 1,2,3 : Variances et Erreurs de vitesse (echant.) : 3
4 : Axes 1,2,3 : Variances uniquement (+ rapide)       : 4
    
```

votre reponse SVP :

Si la variance fait partie de la sélection, il faut choisir en le type d'analyse de variance.

```

NOUVELLE ACQUISITION
Type de Variance
1 : Maximum de la 2ème dérivée de la Variance la plus élevée
2 : Plateau de Variance limite par défaut (150000 [mm2/s2])
3 : Plateau de Variance limite personnalisé
    
```

votre reponse SVP :

Ensuite, les graphiques s'affichent.



9. Menu de traitement

Après les graphiques, le menu de traitement des données apparaît. S'il n'y a pas d'analyse de variance, l'on peut retourner au menu principal ou sauvegarder les données. C'est seulement à cet instant qu'on peut effectuer un nouveau mouvement sur le PC îlot.

S'il y a une analyse de variance, le menu de traitement est plus conséquent. Il affiche toutes les informations issues de l'analyse.

```
NOUVELLE ACQUISITION TERMINEE
Informations sur le mouvement
Code de trajectoire = 231
Code de début de trajectoire = D0
Coordonnées START : X=90 Y=0 Z=-350
Code de fin de trajectoire = E0
Coordonnées END : X=120 Y=0 Z=-350
Criticité automatique
Axe Crit auto = 2
Accel Crit auto = 90000 [mm/s2]
Variance Auto = 65096 [mm2/s2]
Paramètres généraux
Accel Crit Axe1 = 90000 [mm/s2]
Accel Crit Axe2 = 90000 [mm/s2]
Accel Crit Axe3 = 90000 [mm/s2]
Accel Crit Tot 2ème dérivée = 90000 [mm/s2]
Accel Crit Tot Plateau = 0 [mm/s2]
Var Crit Axe1 = 52109 [mm2/s2]
Var Crit Axe2 = 65096 [mm2/s2]
Var Crit Axe3 = 50667 [mm2/s2]
Var Crit Tot 2ème dérivée = 167872 [mm2/s2]
Var Crit Tot Plateau = 0 [mm2/s2]

0 : Revenir au menu principal
1 : Effectuer une nouvelle acquisition

Test unique
2 : Sauvegarder l acquisition AVEC l'accél. critique auto
3 : Sauvegarder l acquisition SANS l'accél. critique auto

Serie de test répertoriée
4 : Sauvegarder l acquisition AVEC l'accél. critique auto
5 : Sauvegarder l acquisition SANS l'accél. critique auto

votre reponse SVP :
```

Si les déductions automatiques des paramètres critiques correspondent bien au graphique, l'on peut effectuer une sauvegarde avec ces valeurs. Sinon l'utilisateur devra entrer manuellement ces valeurs. Ces dernières sont sauvegardées dans un fichier TXT, avec les données, à l'adresse :

« D:\Robot_Delta_INT_History \».

Si c'est un essai unique, les fichiers sont renommés selon la date de l'essai, mais s'il s'agit d'un essai d'une série cartographique, un dossier au numéro de la série est créé et les fichiers renommés avec le numéro de code de la trajectoire. Le mouvement suivant peut être effectué et l'on recommence.

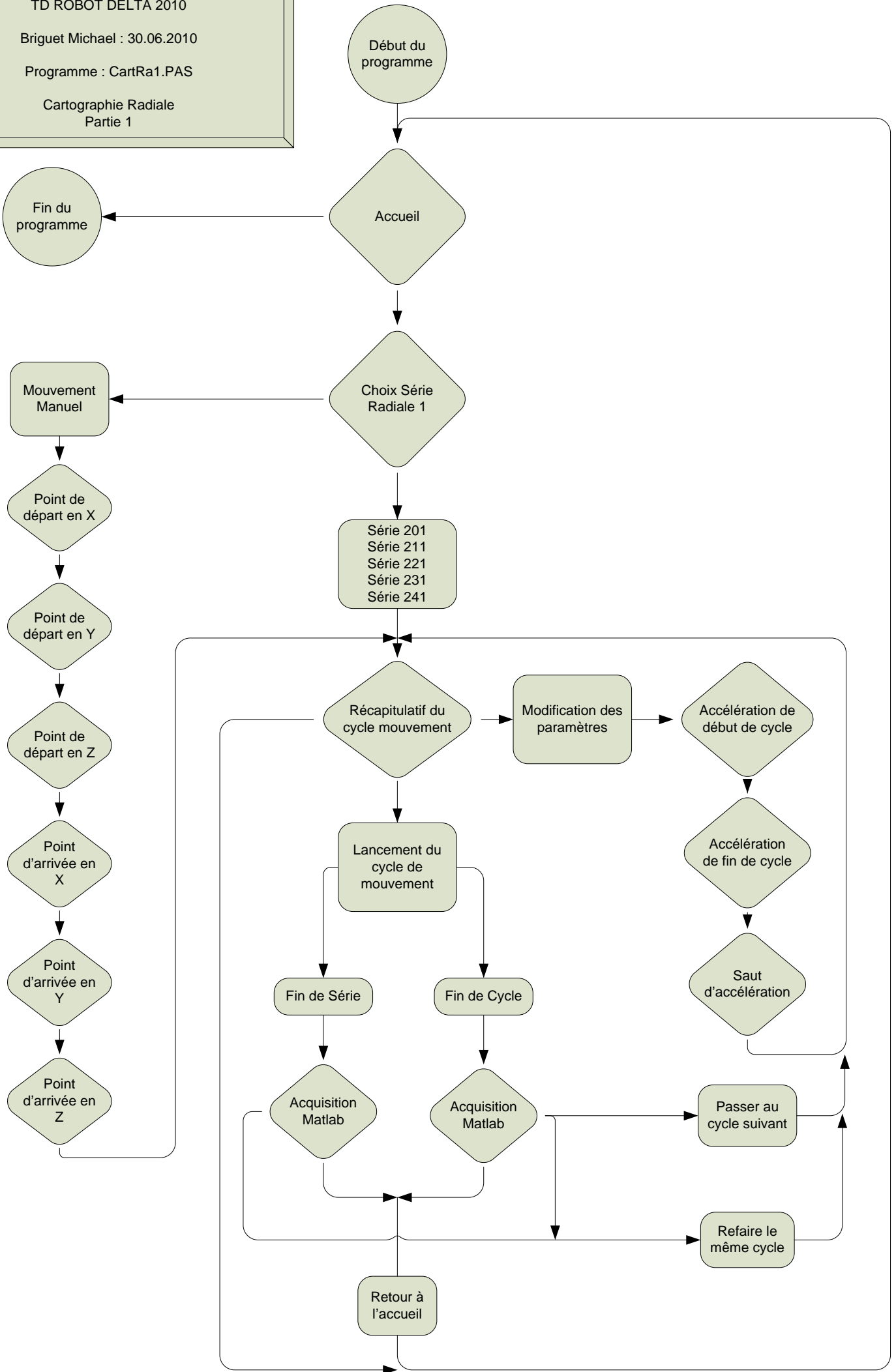
C'est uniquement après la sauvegarde qu'un nouveau mouvement peut-être démarré sur le PC îlot !

ANNEXE 6

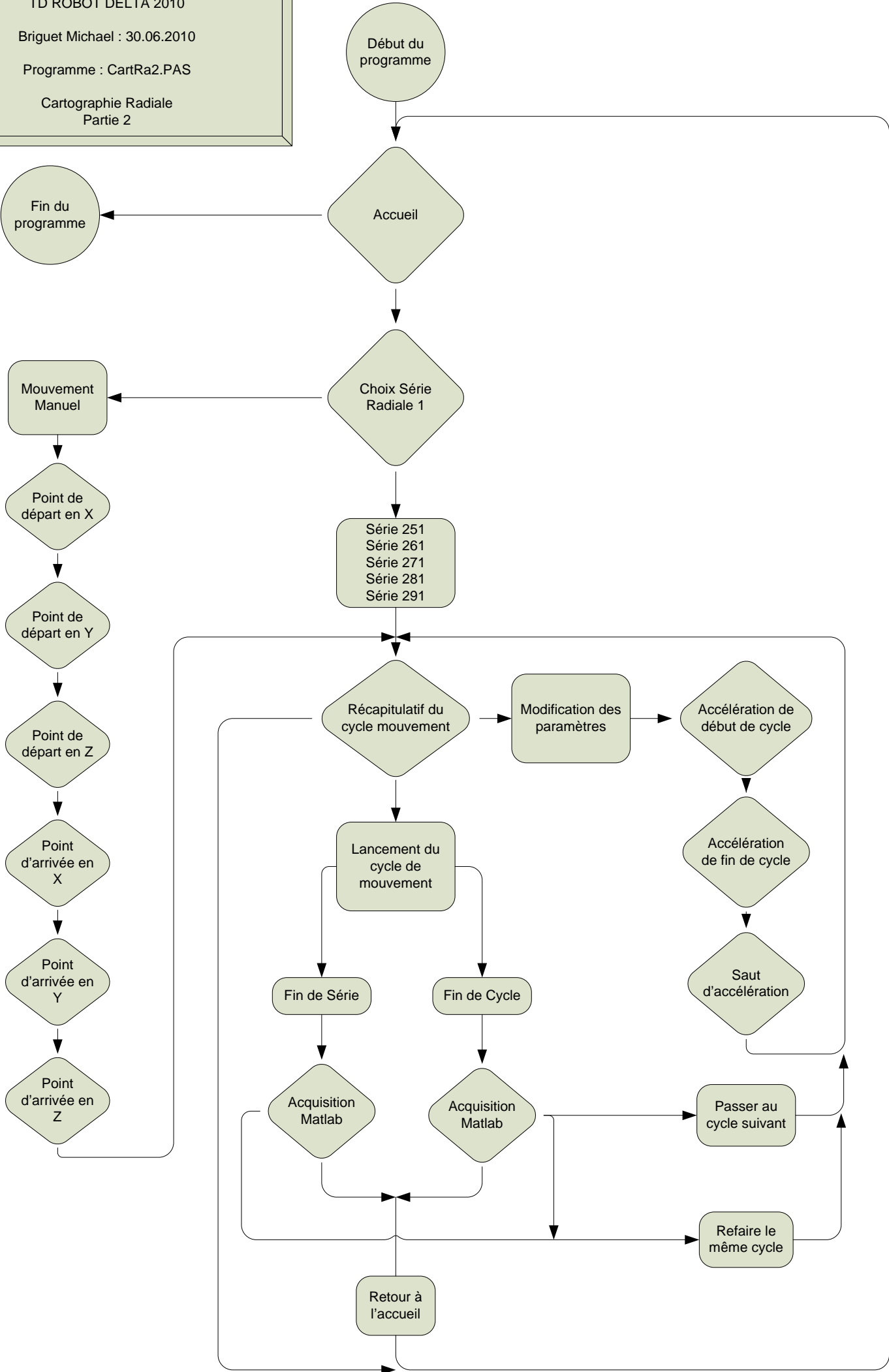
Structogrammes

- **CARTRA1.PAS**
- **CARTRA2.PAS**
- **CARTTAN.PAS**
- **DELTA.M**
- **NEW_ACQUI.M**
- **OLD_ACQUI.M**
- **CARTO.M**

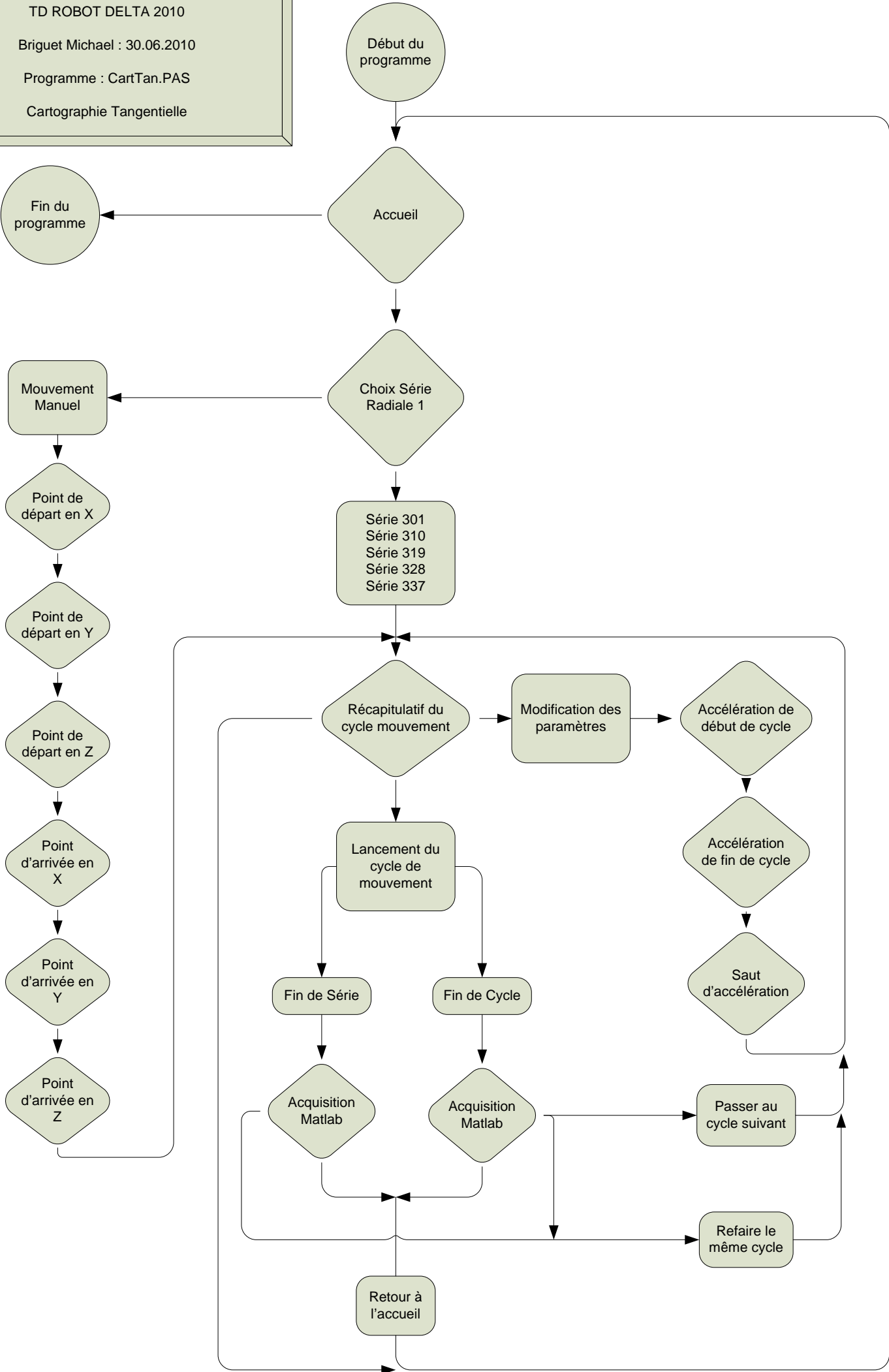
TD ROBOT DELTA 2010
Briguet Michael : 30.06.2010
Programme : CartRa1.PAS
Cartographie Radiale
Partie 1



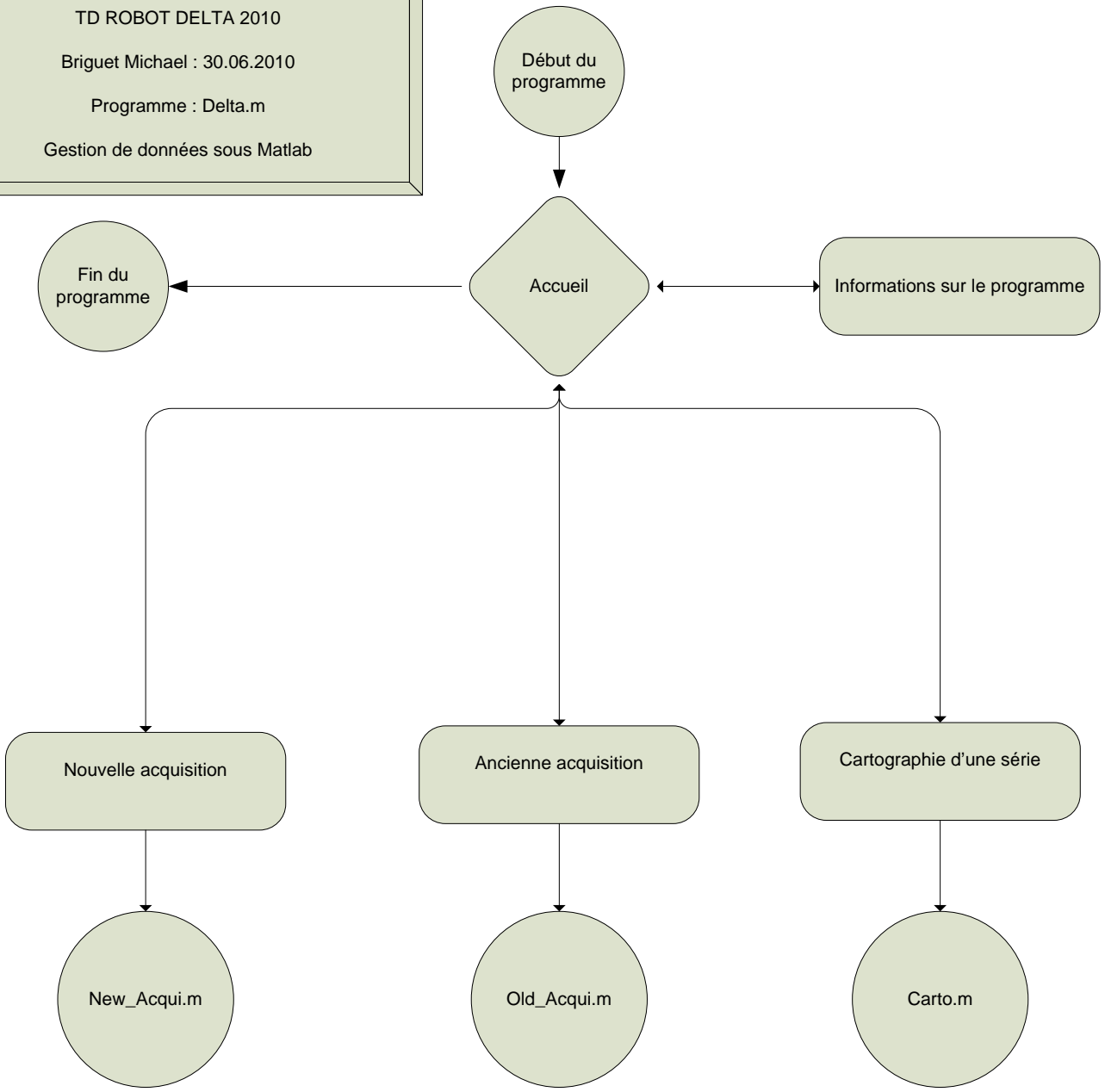
TD ROBOT DELTA 2010
Briguet Michael : 30.06.2010
Programme : CartRa2.PAS
Cartographie Radiale
Partie 2



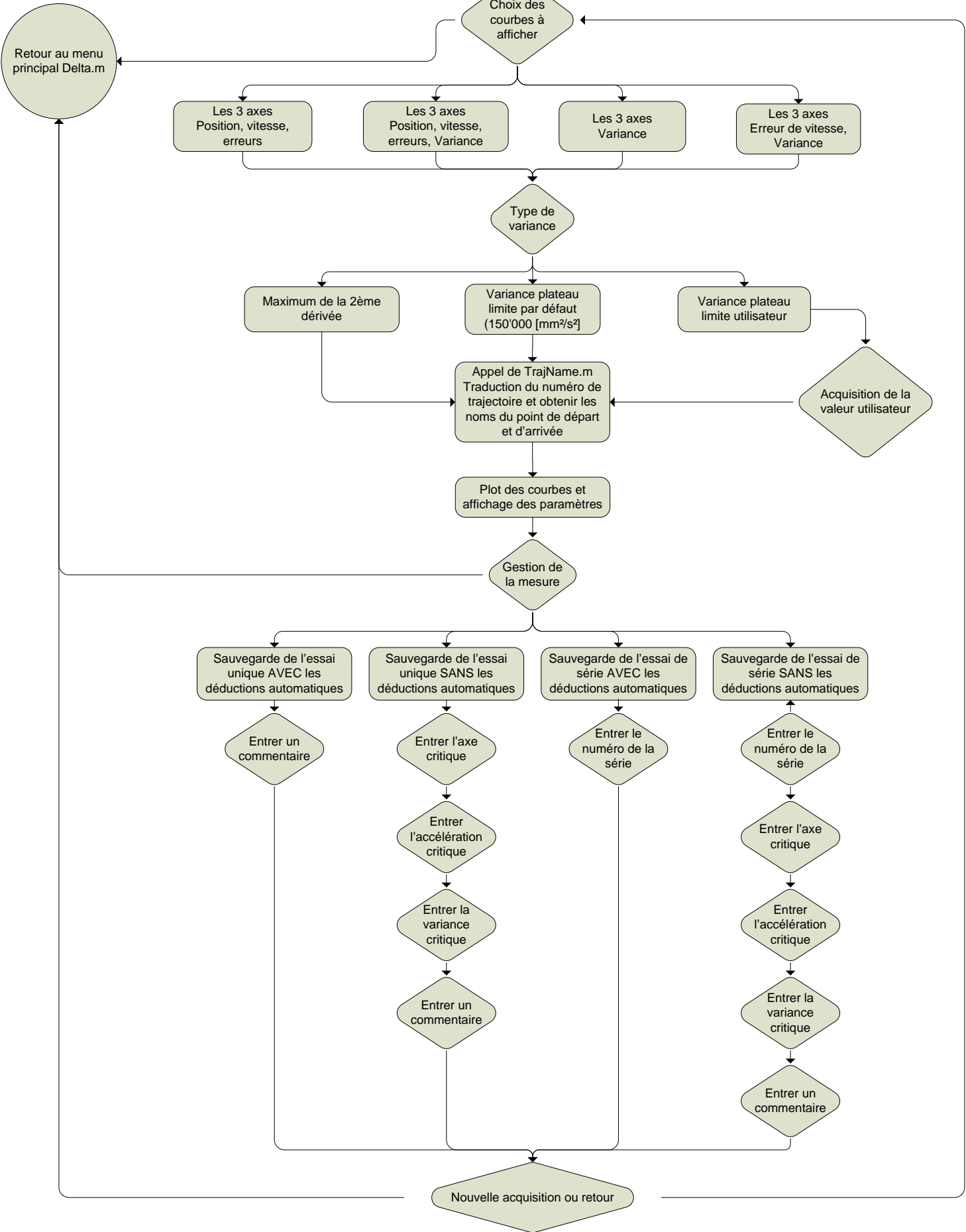
TD ROBOT DELTA 2010
Briguet Michael : 30.06.2010
Programme : CartTan.PAS
Cartographie Tangentielle



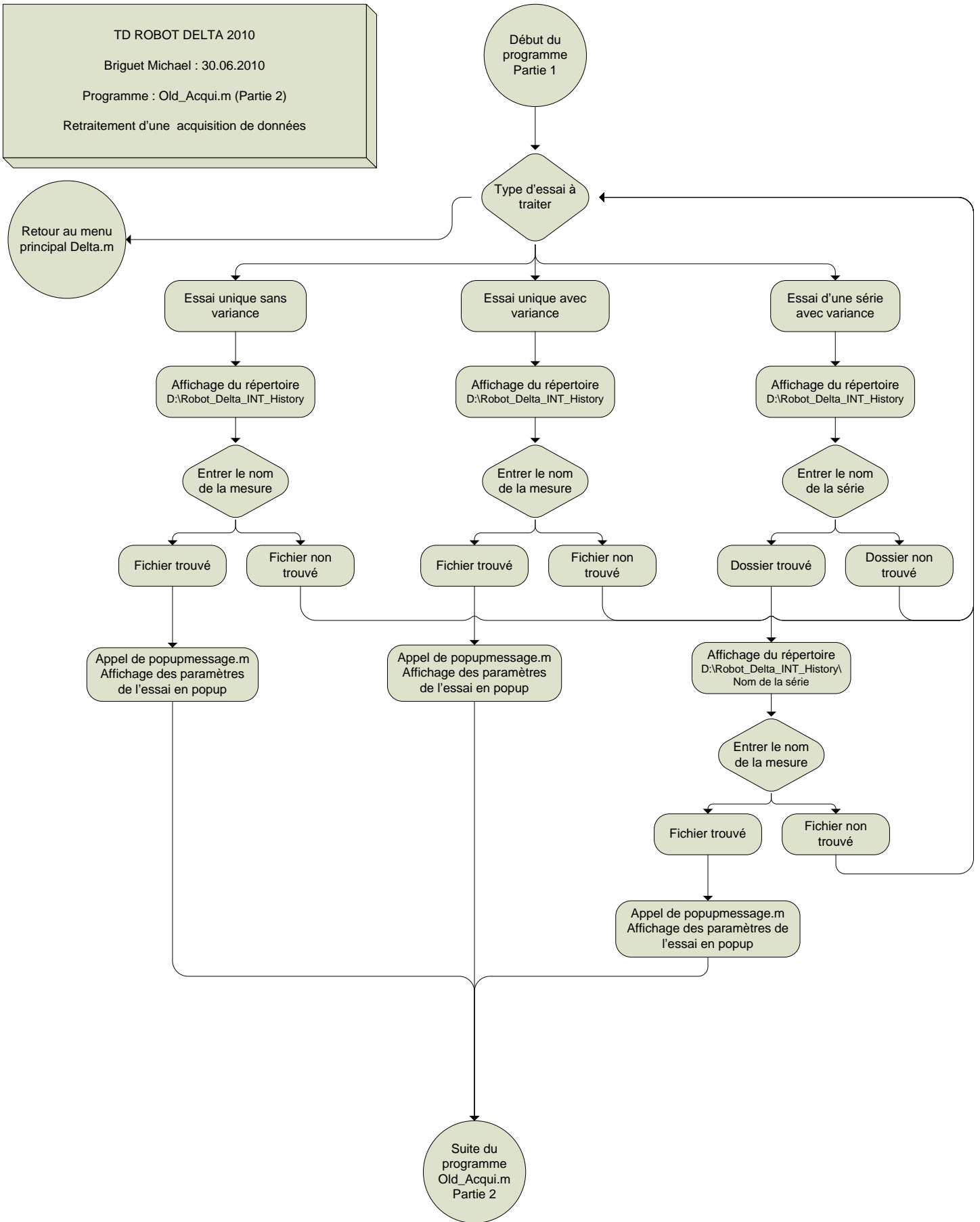
TD ROBOT DELTA 2010
Briguet Michael : 30.06.2010
Programme : Delta.m
Gestion de données sous Matlab

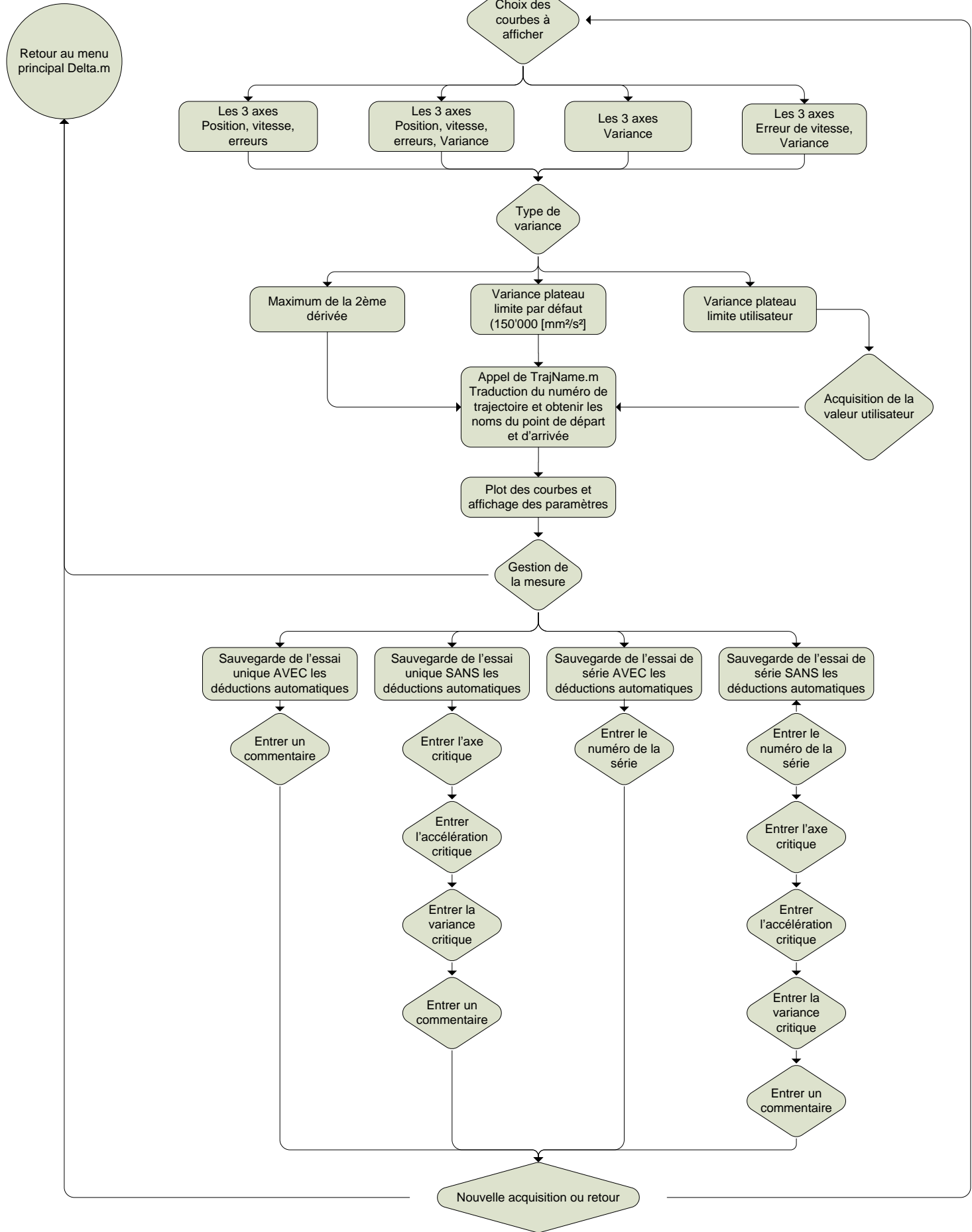


TD ROBOT DELTA 2010
 Briguet Michael : 30.06.2010
 Programme : New_Acqui.m
 Nouvelle acquisition de données



TD ROBOT DELTA 2010
Briguet Michael : 30.06.2010
Programme : Old_Acqui.m (Partie 2)
Retraitement d'une acquisition de données





TD ROBOT DELTA 2010
 Briguet Michael : 30.06.2010
 Programme : Carto.m
 Représentation de la cartographie en 3D

Début du programme

Entrer le nom de la Série

Dossier non trouvé

Dossier trouvé

Choix du paramètre à cartographier

Axe critique déduit

Accélération critique déduite

Accélération critique axe 2 (Non fonctionnel)

Accélération critique totale (2ème dérivée)

Variance critique axe 1 (Non fonctionnel)

Variance critique axe 3 (Non fonctionnel)

Variance critique totale (Plateau)

Variance critique déduite

Accélération critique axe 1 (Non fonctionnel)

Accélération critique axe 3 (Non fonctionnel)

Accélération critique totale (Plateau)

Variance critique axe 2 (Non fonctionnel)

Variance critique totale (2ème dérivée)

Appel de SearchParam.m
 Pour rechercher les paramètres désirés

Création du tableau de cartographie de 10 sur 10 représentant un secteur de 0° à 60° de 270mm de rayon.
 Détection automatique du type de Série

Appel de SectorTabRad100.m
 Pour une série Radiale à 100 points de mesures

Appel de SectorTabTan45.m
 Pour une série Tangentielle à 45 points de mesures

Appel de SectorTabRad64.m
 Pour une série Radiale à 64 points de mesures

Affichage de la moyenne

Choix du type de plot

Remarque contour:
 Pas pour l'axe critique déduit

Mesh

Surf

Surfn

Contour

Appel de polarplot3d.m
 Plot du secteur de 0° à 60°

Appel de polarplot3d.m
 Plot du secteur de 120° à 180°

Appel de polarplot3d.m
 Plot du secteur de 240° à 300°

Appel de ColorMapAccel.mat
 Echelle fixe pour les accélérations

Appel de polarplot3d.m
 Plot du secteur de 60° à 120°

Appel de polarplot3d.m
 Plot du secteur de 180° à 240°

Appel de polarplot3d.m
 Plot du secteur de 300° à 360°

Appel de ColorMapVar.mat
 Echelle fixe pour les Variances

Plot de la cartographie en 3D

Appel de drawLine.m
 Plot de la représentation des axes

Remarque polarplot3d.m
 Ce script fait appel à:
 rotatePoints.m
 parse_pv_pairs.m

Retour au menu principal Delta.m

Retour au menu principal Delta.m

Suite du programme
 Old_Acqui.m
 Partie 2

Choix des courbes à afficher

- Les 3 axes Position, vitesse, erreurs
- Les 3 axes Position, vitesse, erreurs, Variance
- Les 3 axes Variance
- Les 3 axes Erreur de vitesse, Variance

Type de variance

- Maximum de la 2ème dérivée
- Variance plateau limite par défaut (150'000 [mm²/s²])
- Variance plateau limite utilisateur

Acquisition de la valeur utilisateur

Plot des courbes et affichage des paramètres

Gestion de la mesure

- Sauvegarde de l'essai unique AVEC les déductions automatiques
- Sauvegarde de l'essai unique SANS les déductions automatiques
- Sauvegarde de l'essai de série AVEC les déductions automatiques
- Sauvegarde de l'essai de série SANS les déductions automatiques

Entrer un commentaire

Entrer l'axe critique

Entrer l'accélération critique

Entrer la variance critique

Entrer un commentaire

Entrer le numéro de la série

Entrer le numéro de la série

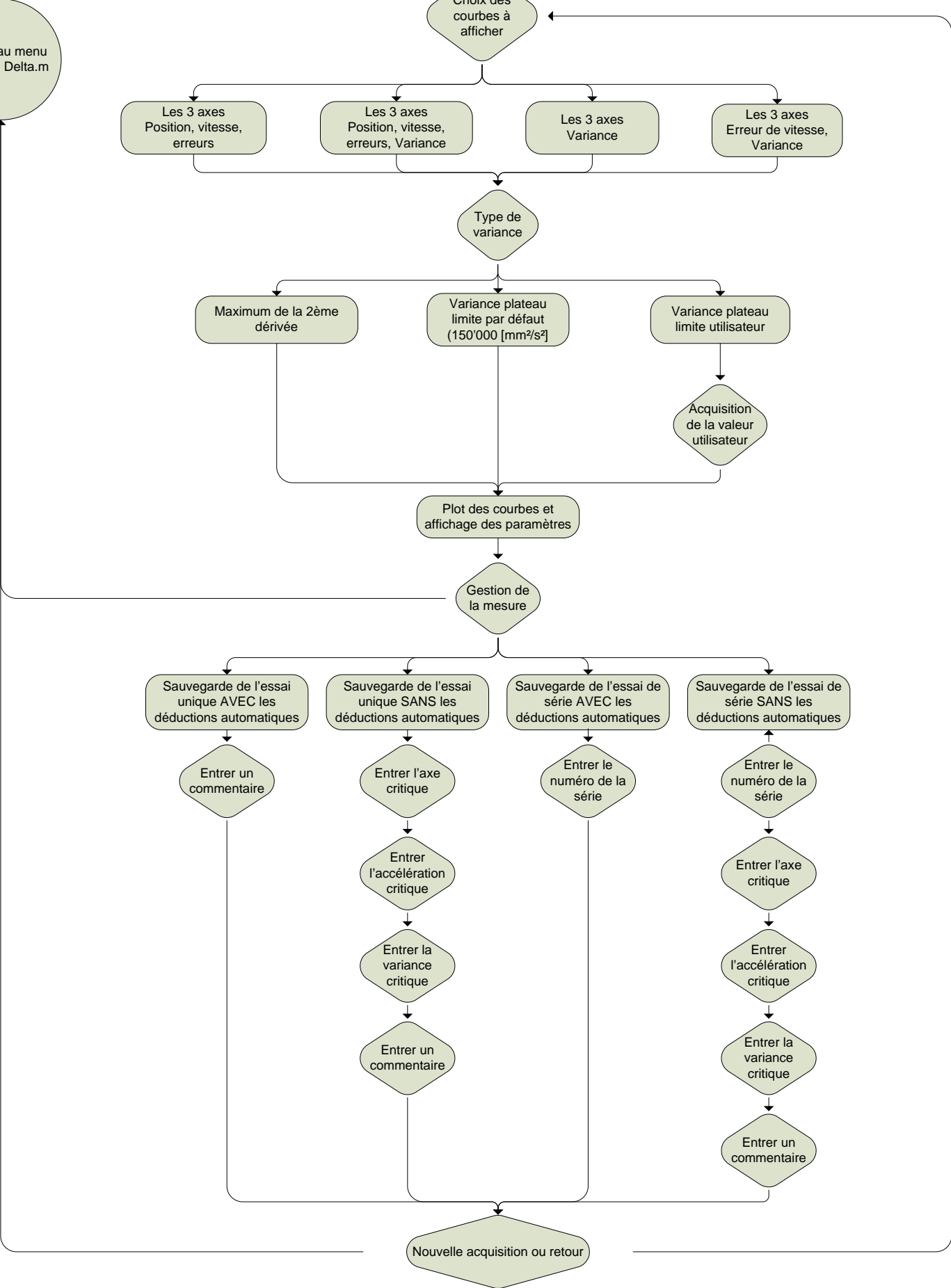
Entrer l'axe critique

Entrer l'accélération critique

Entrer la variance critique

Entrer un commentaire

Nouvelle acquisition ou retour



ANNEXE 7

Exemple d'un fichier de données .TXT

DATE ET HEURE :
2010-6-30-8h7m4s

Nom de la série :
10

Code de mouvement:
213

Point de départ:
B2

Point d'arrivée:
C2

Coordonnées de Départ [mm]
X=29 Y=7 Z=-350

Coordonnées de d Arrivée [mm]
X=58 Y=14 Z=-350

Accel Min = [mm/s²]
200000

Accel Max = [mm/s²]
800000

Step Accel = [mm/s²]
20000

Nombre de Step=
31

Type de Variance automatique =
2ème dérivée uniquement

Accel Crit Axe1 = [mm/s²]
740000

Accel Crit Axe2 = [mm/s²]
280000

Accel Crit Axe3 = [mm/s²]
260000

Accel Crit Tot 2ème dérivée = [mm/s²]
760000

Accel Crit Tot Plateau = [mm/s²]
0

Var Crit Axe1 = [mm²/s²]
4.810216e+005

Var Crit Axe2 = [mm²/s²]
2.071443e+005

Var Crit Axe3 = [mm²/s²]
4.020315e+004

Var Crit Tot 2ème dérivée = [mm²/s²]
1.228570e+006

Var Crit Tot Plateau = [mm²/s²]
0

ACCELERATION CRITIQUE DEDUITE = [mm/s²]
740000

AXE CRITIQUE =
1

VARIANCE CRITIQUE DEDUITE = [mm²/s²]
4.810216e+005

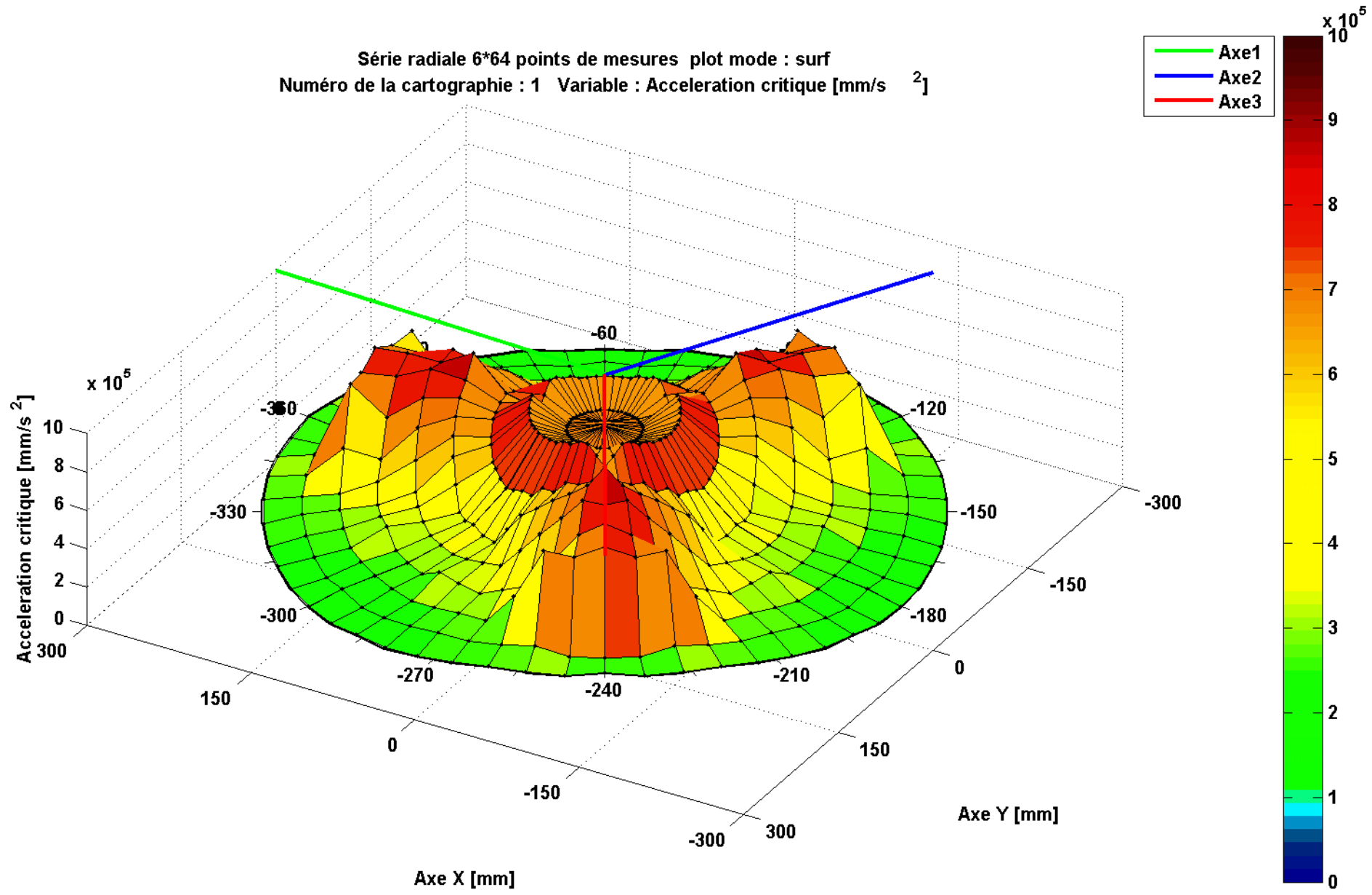
Commentaires :
OK!

ANNEXE 8

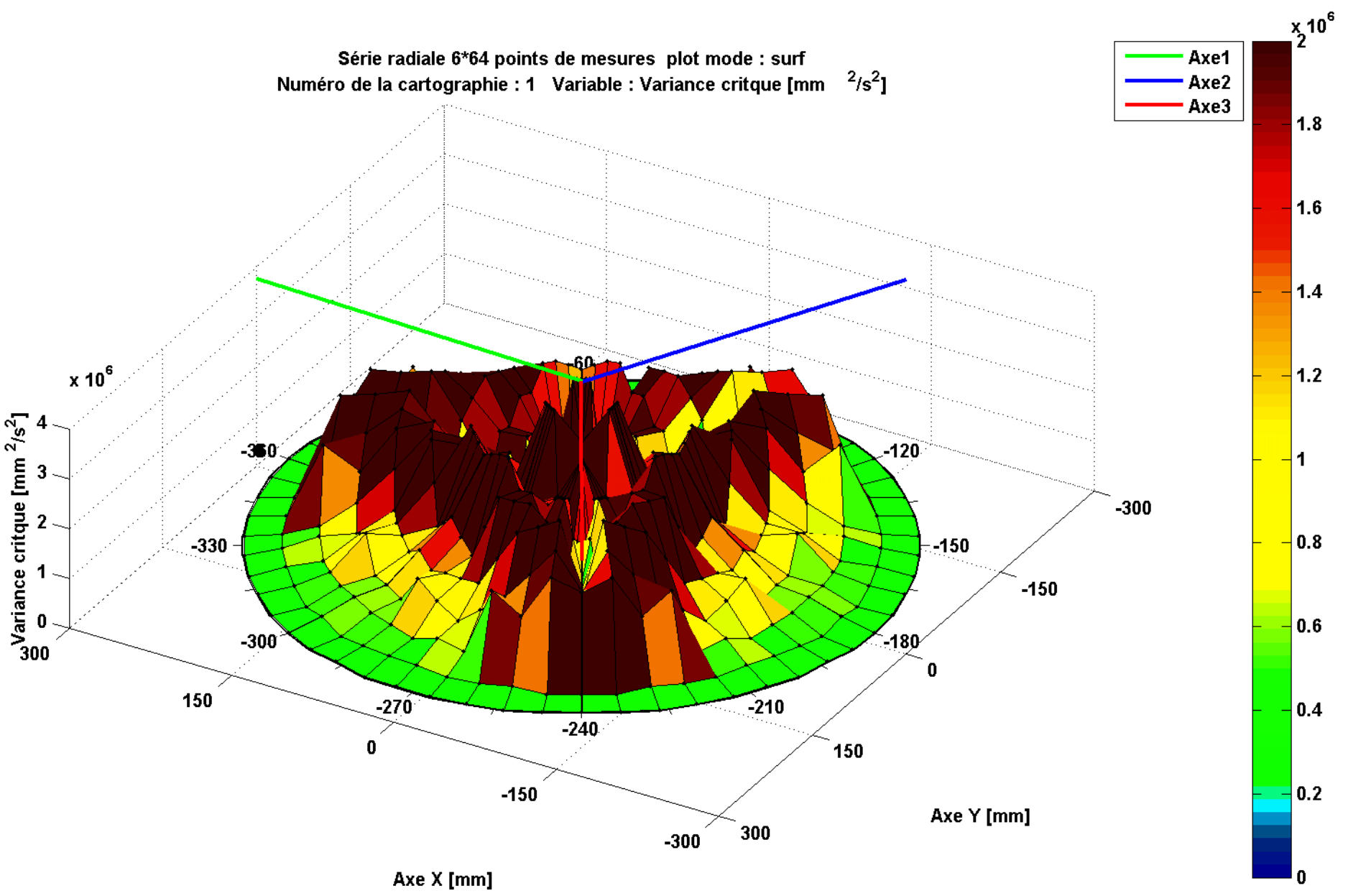
Cartographies

- **SERIE1_RADIALE_Z=-350MM_0KG_ACCEL_SURF**
- **SERIE1_RADIALE_Z=-350MM_0KG_VARIANCE_SURF**
- **SERIE10_RADIALE_Z=-350MM_0KG_ACCEL_SURF**
- **SERIE10_RADIALE_Z=-350MM_0KG_VARIANCE_SURF**
- **SERIE10_RADIALE_Z=-350MM_0KG_AXE_SURF**
- **SERIE11_TANGENTIELLE_Z=-350MM_0KG_ACCEL_SURF**
- **SERIE11_TANGENTIELLE_Z=-350MM_0KG_VARIANCE_SURF**
- **SERIE11_TANGENTIELLE_Z=-350MM_0KG_AXE_SURF**
- **SERIE30_RADIALE_Z=-350MM_1KG_ACCEL_SURF**
- **SERIE30_RADIALE_Z=-350MM_1KG_VARIANCE_SURF**
- **SERIE31_TANGENTIELLE_Z=-350MM_1KG_ACCEL_SURF**
- **SERIE31_TANGENTIELLE_Z=-350MM_1KG_VARIANCE_SURF**
- **SERIE40_RADIALE_Z=-350MM_1KG_ACCEL_PLATEAU=150000MM2s-2_SURF**
- **SERIE40_RADIALE_Z=-350MM_1KG_VARIANCE_PLATEAU=150000MM2s-2_SURF**
- **SERIE40_RADIALE_Z=-350MM_1KG_AXE_PLATEAU=150000MM2s-2_SURF**
- **SERIE50_RADIALE_Z=-450MM_1KG_ACCEL_SURF**
- **SERIE50_RADIALE_Z=-450MM_1KG_VARIANCE_SURF**

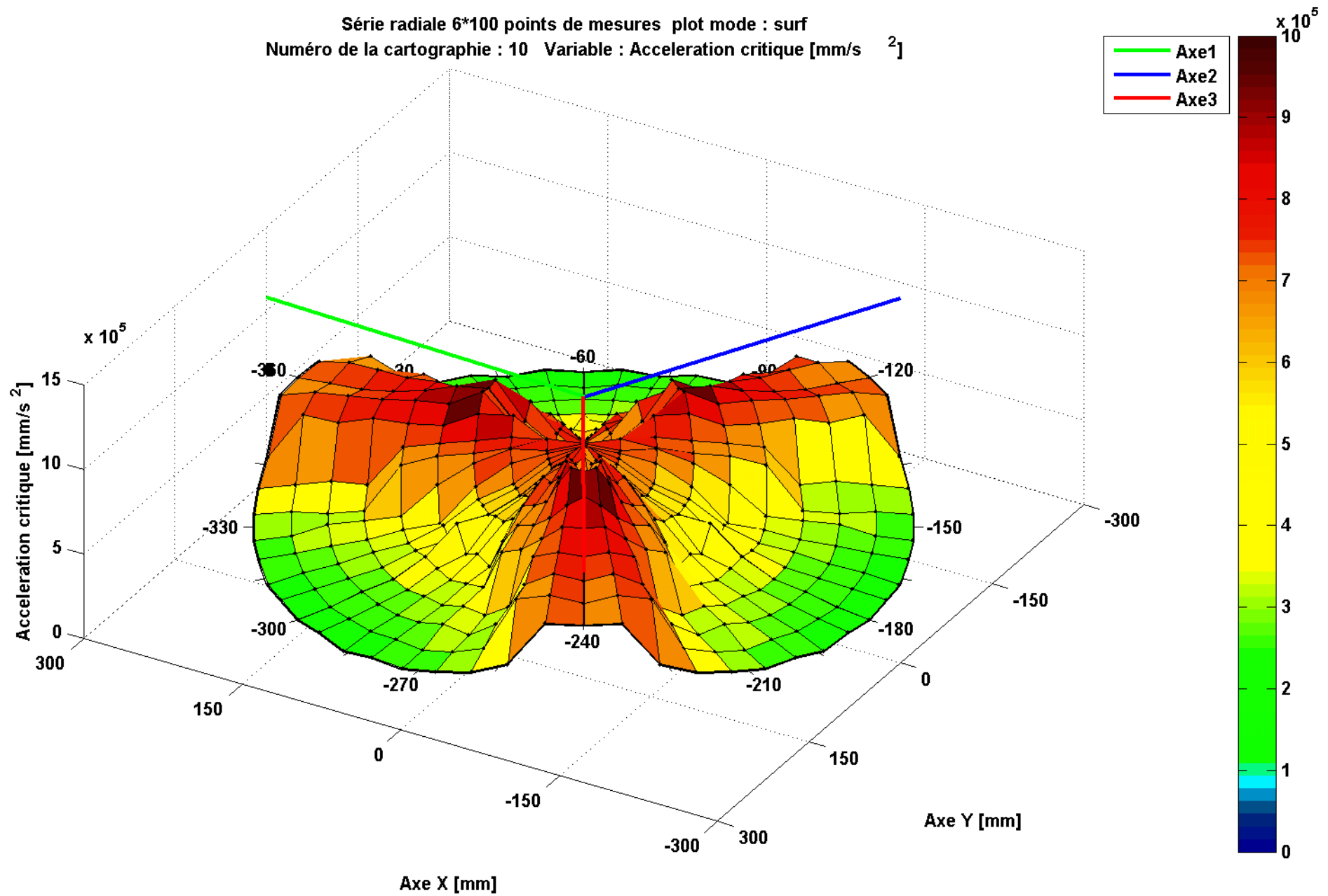
Série radiale 6*64 points de mesures plot mode : surf
Numéro de la cartographie : 1 Variable : Accelération critique [mm/s²]



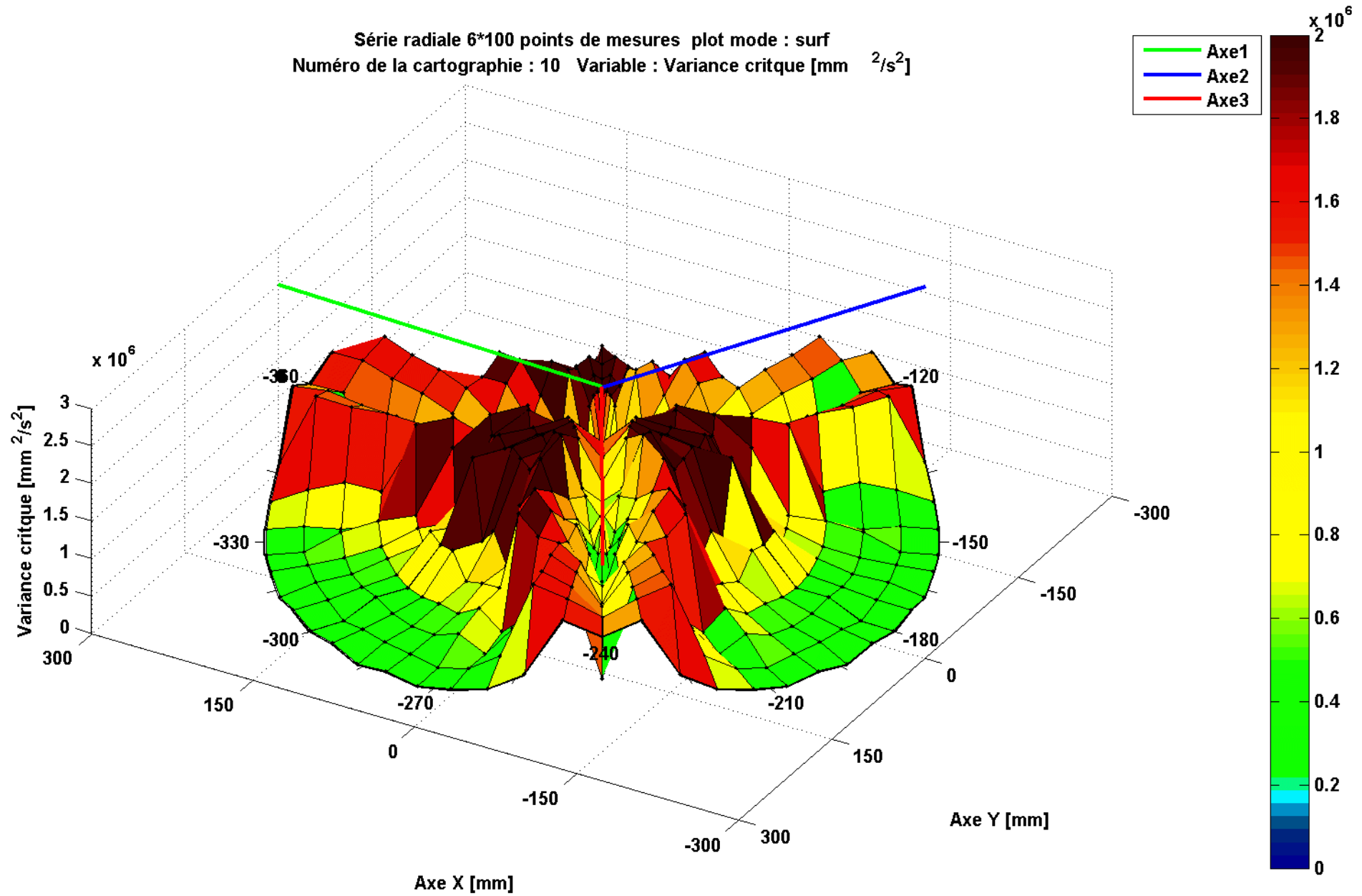
Série radiale 6*64 points de mesures plot mode : surf
Numéro de la cartographie : 1 Variable : Variance critique [mm²/s²]



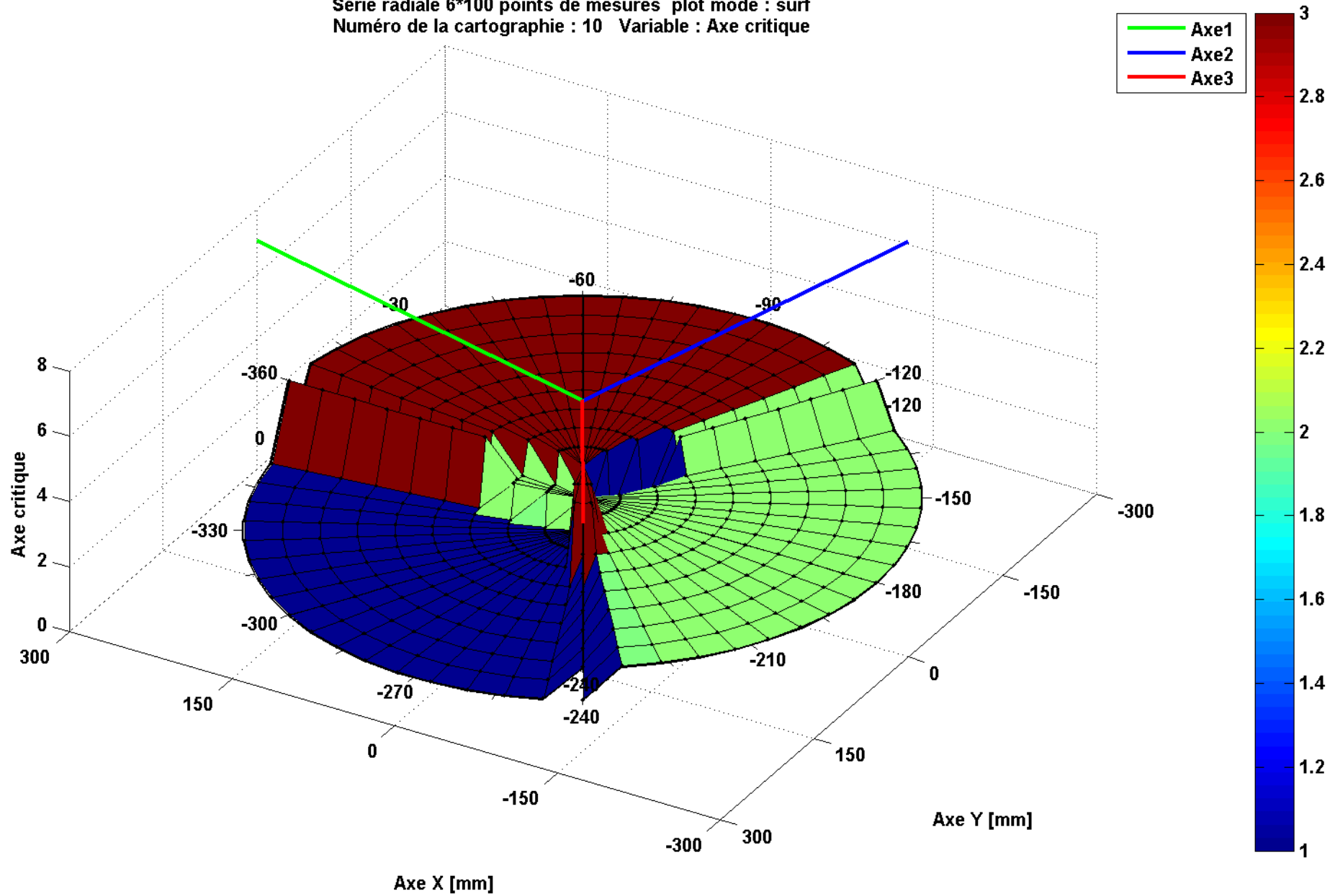
Série radiale 6*100 points de mesures plot mode : surf
Numéro de la cartographie : 10 Variable : Accelération critique [mm/s²]



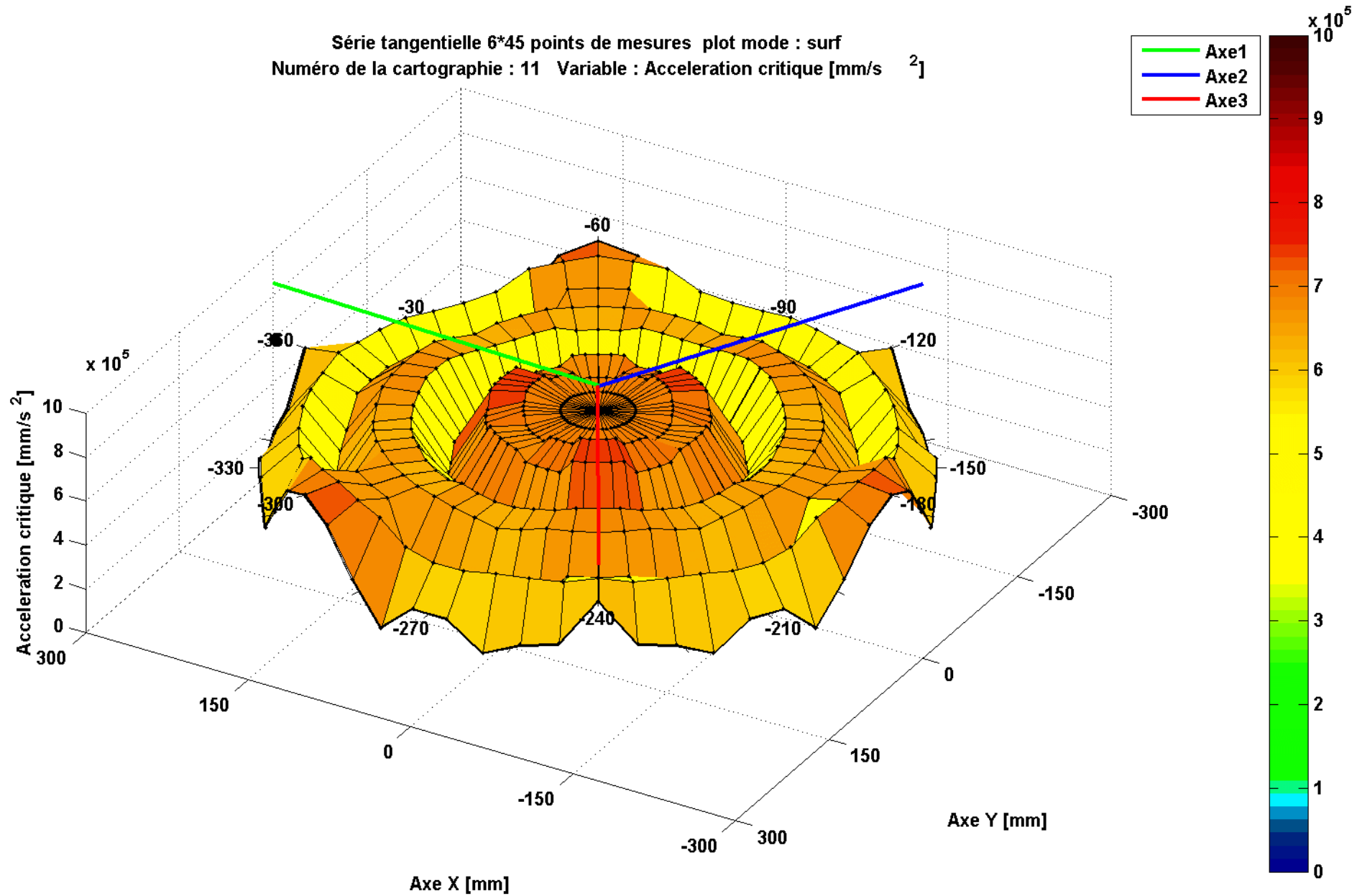
Série radiale 6*100 points de mesures plot mode : surf
Numéro de la cartographie : 10 Variable : Variance critique [mm²/s²]



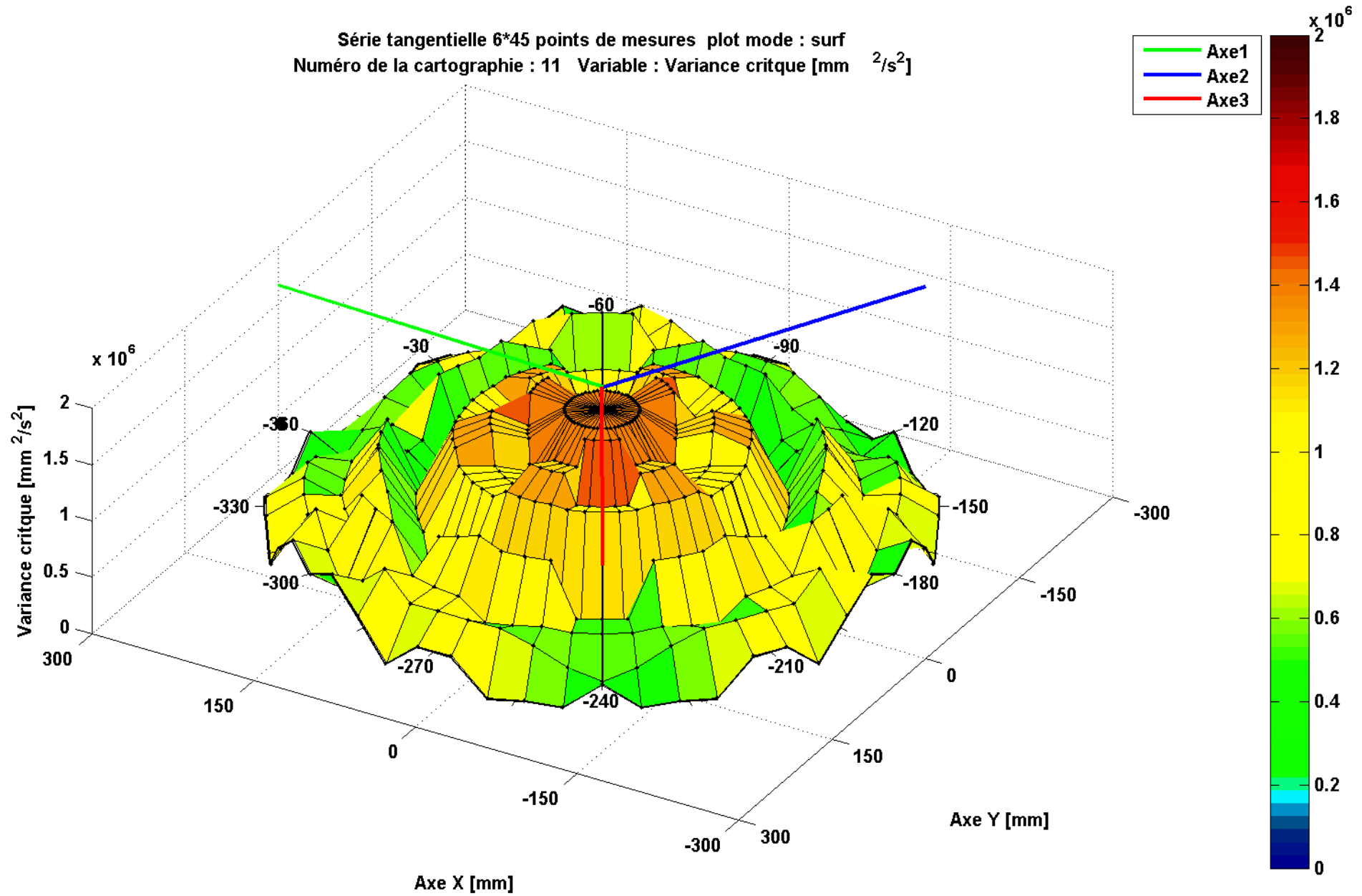
Série radiale 6*100 points de mesures plot mode : surf
Numéro de la cartographie : 10 Variable : Axe critique



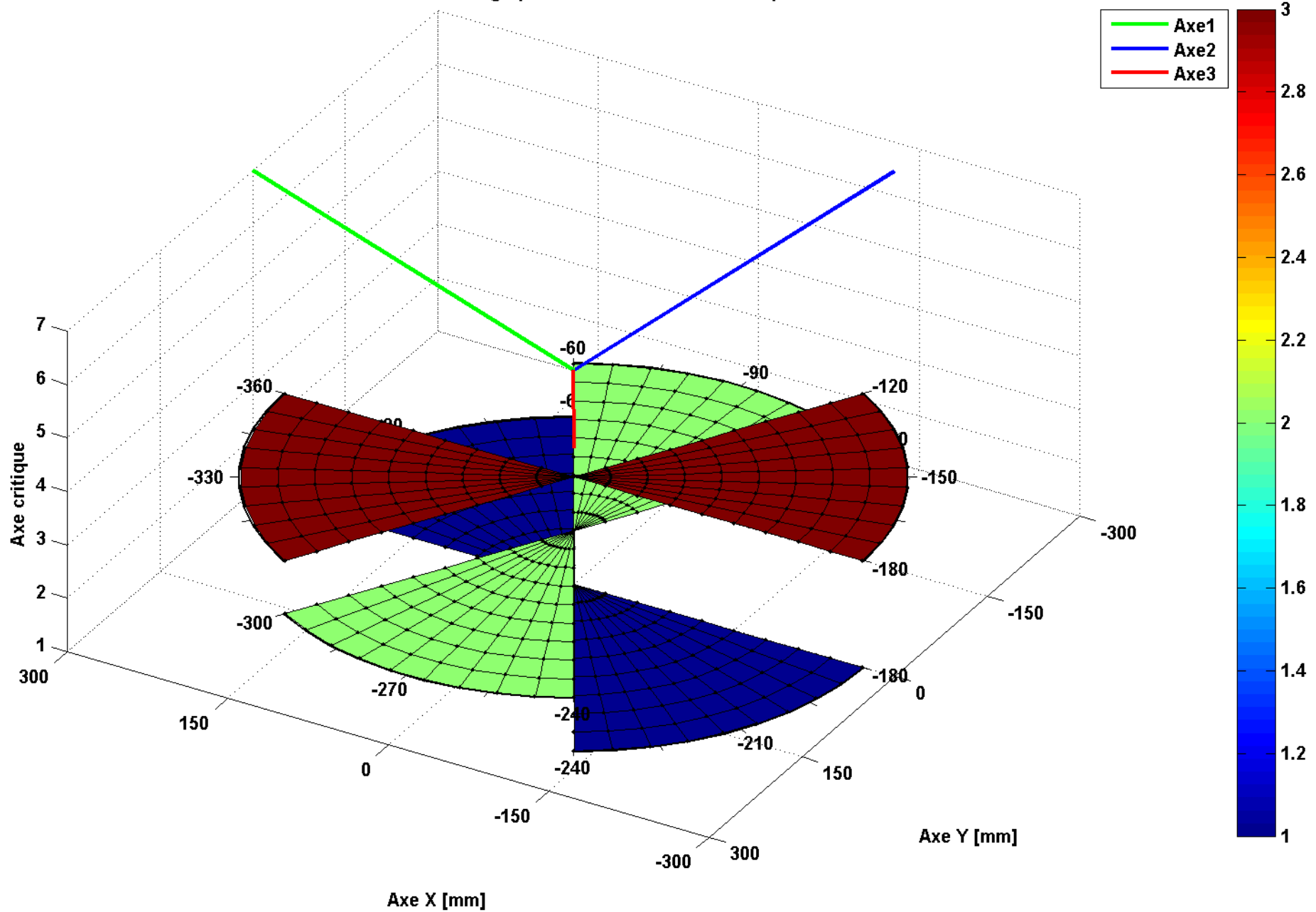
Série tangentielle 6*45 points de mesures plot mode : surf
Numéro de la cartographie : 11 Variable : Accelération critique [mm/s²]



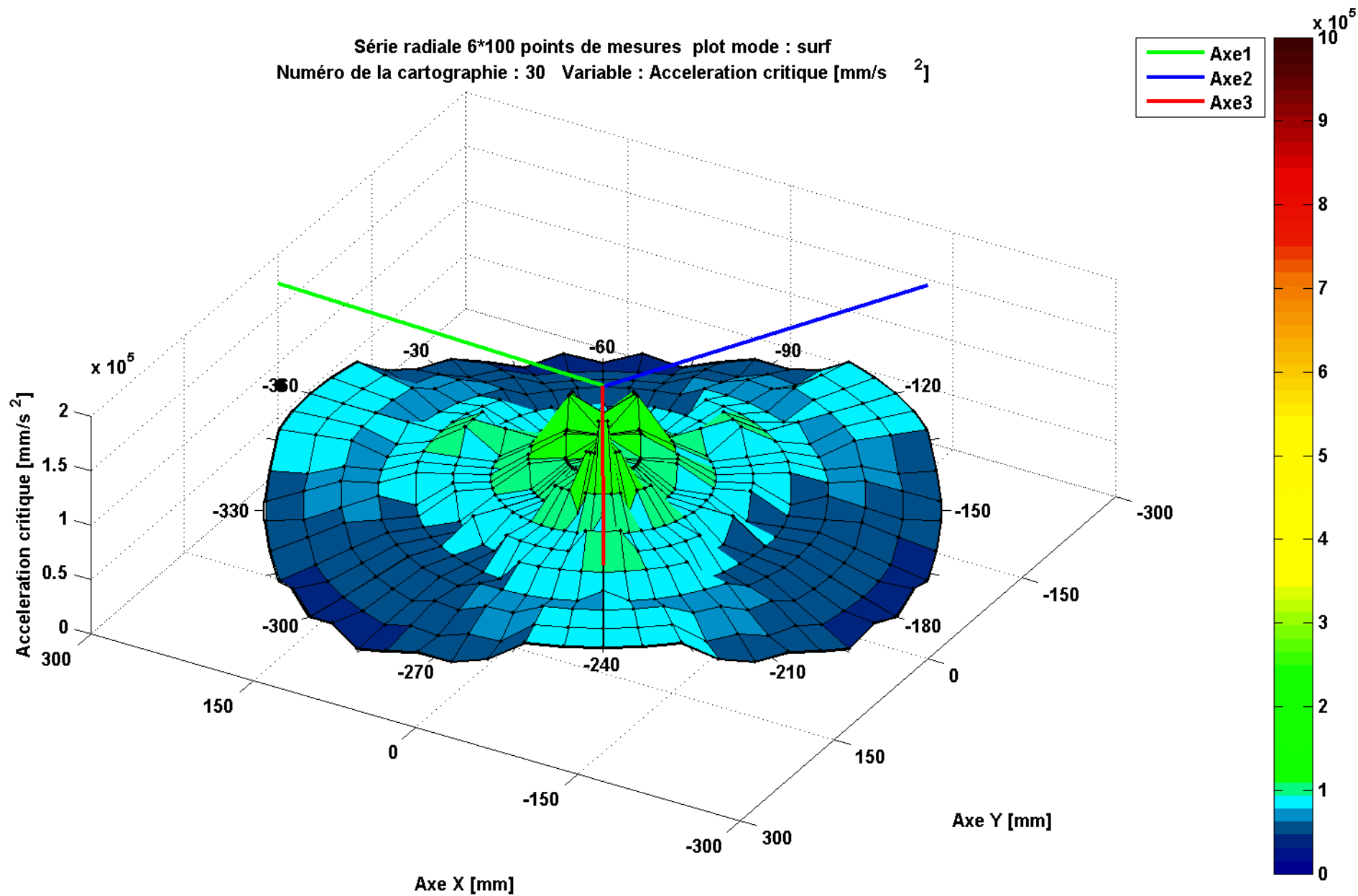
Série tangentielle 6*45 points de mesures plot mode : surf
Numéro de la cartographie : 11 Variable : Variance critique [mm^2/s^2]



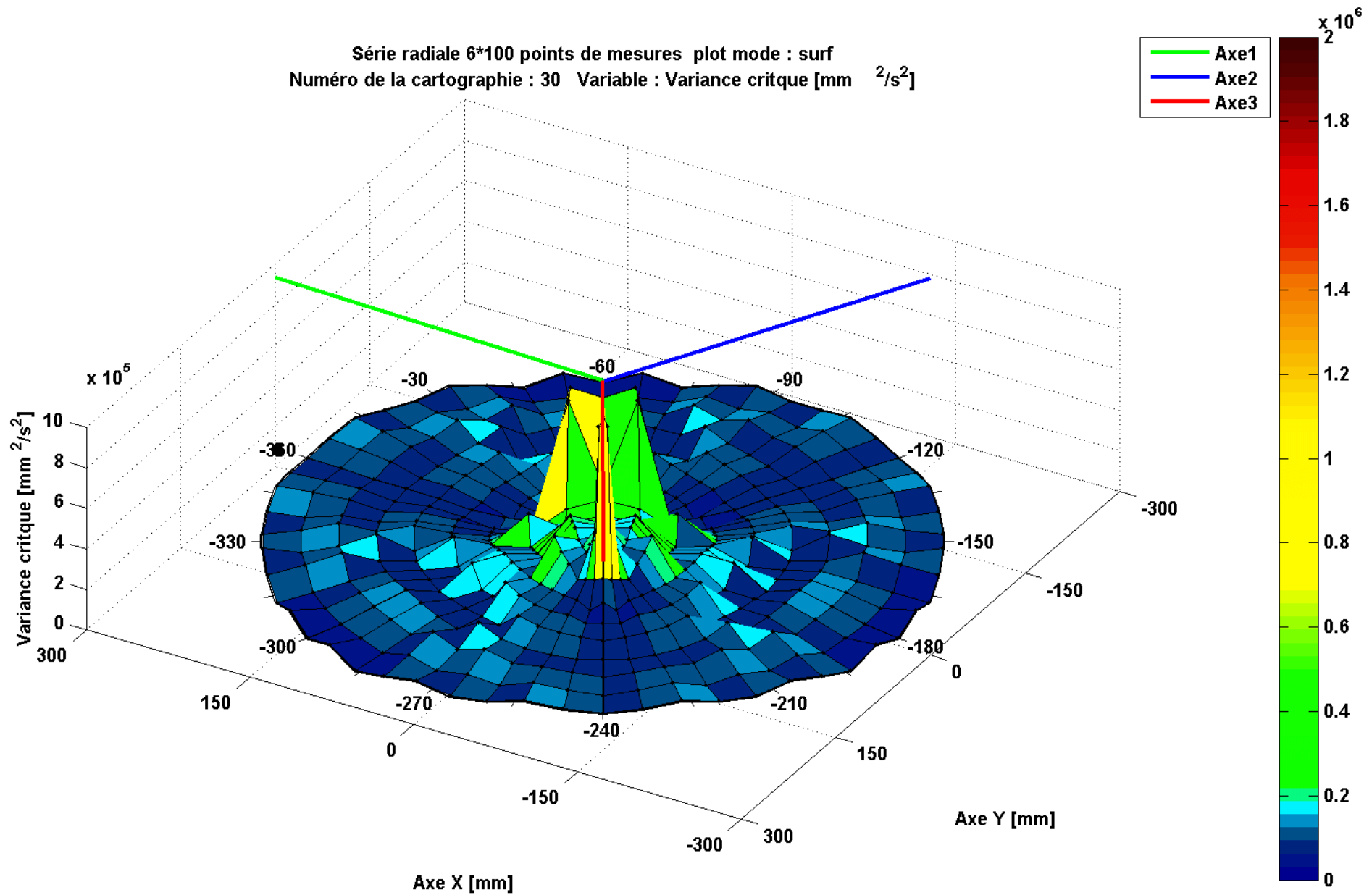
Série tangentielle 6*45 points de mesures plot mode : surf
Numéro de la cartographie : 11 Variable : Axe critique



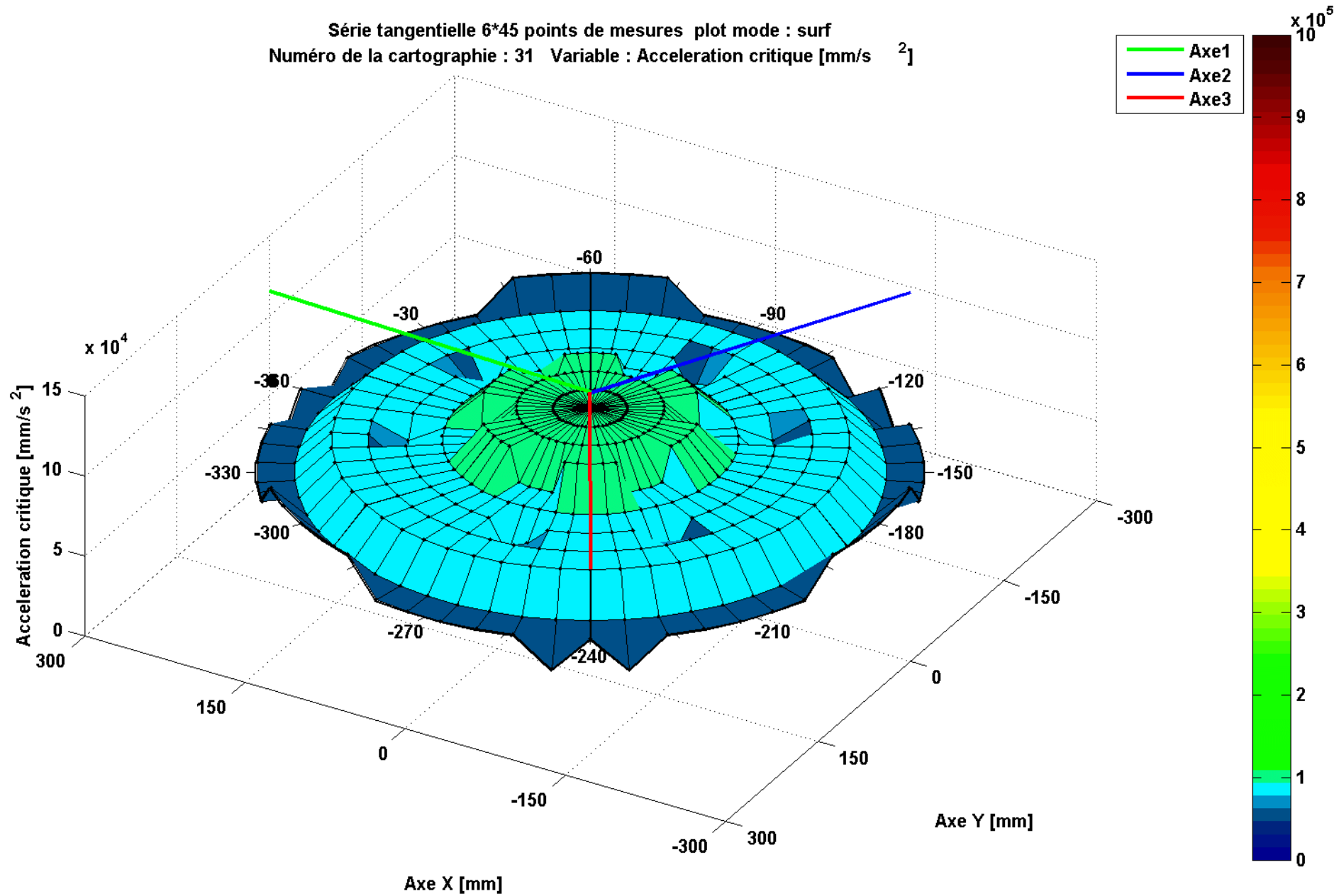
Série radiale 6*100 points de mesures plot mode : surf
Numéro de la cartographie : 30 Variable : Accelération critique [mm/s²]



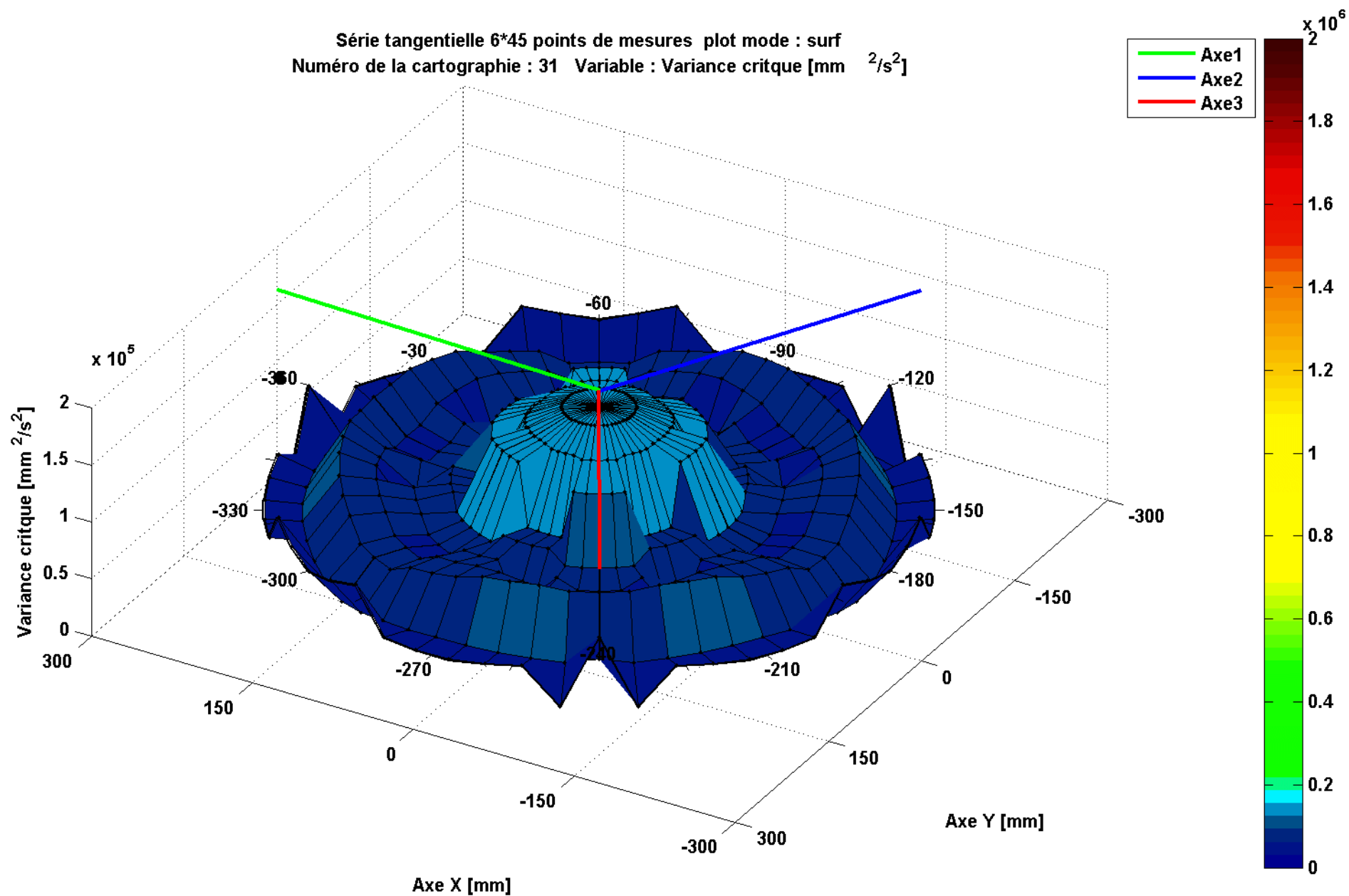
Série radiale 6*100 points de mesures plot mode : surf
Numéro de la cartographie : 30 Variable : Variance critique [mm²/s²]



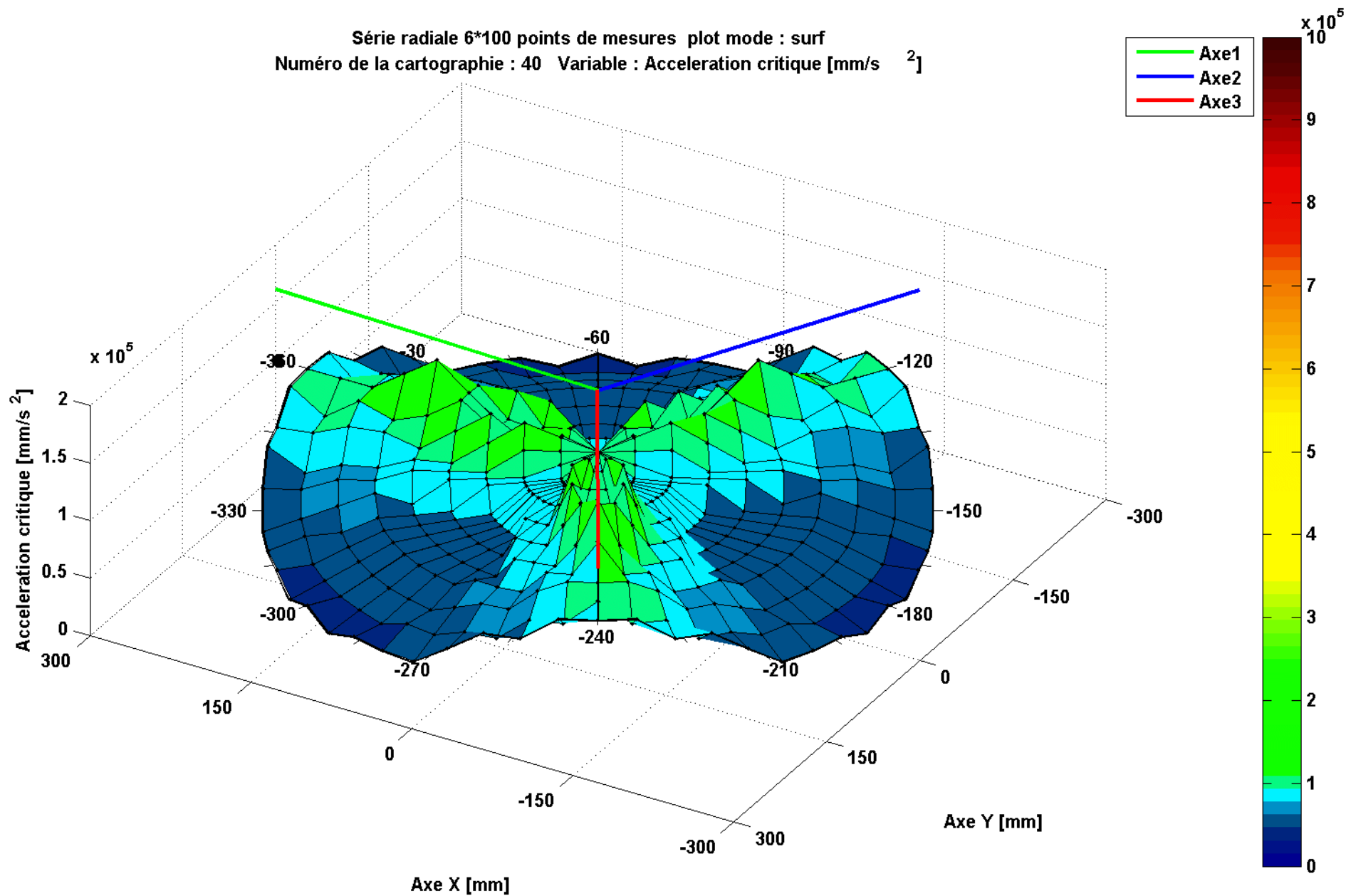
Série tangentielle 6*45 points de mesures plot mode : surf
Numéro de la cartographie : 31 Variable : Accélération critique [mm/s²]



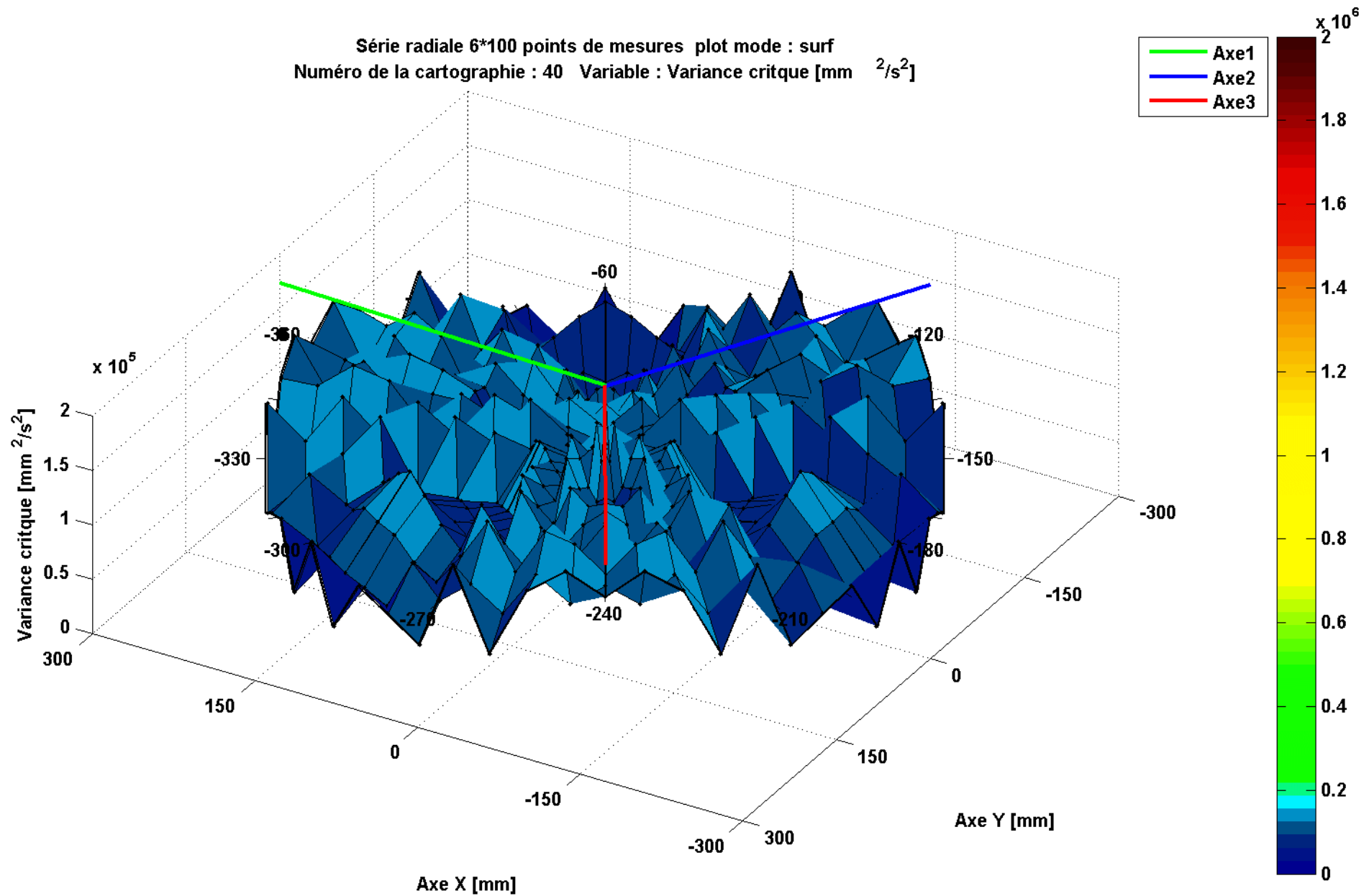
Série tangentielle 6*45 points de mesures plot mode : surf
Numéro de la cartographie : 31 Variable : Variance critique [mm^2/s^2]



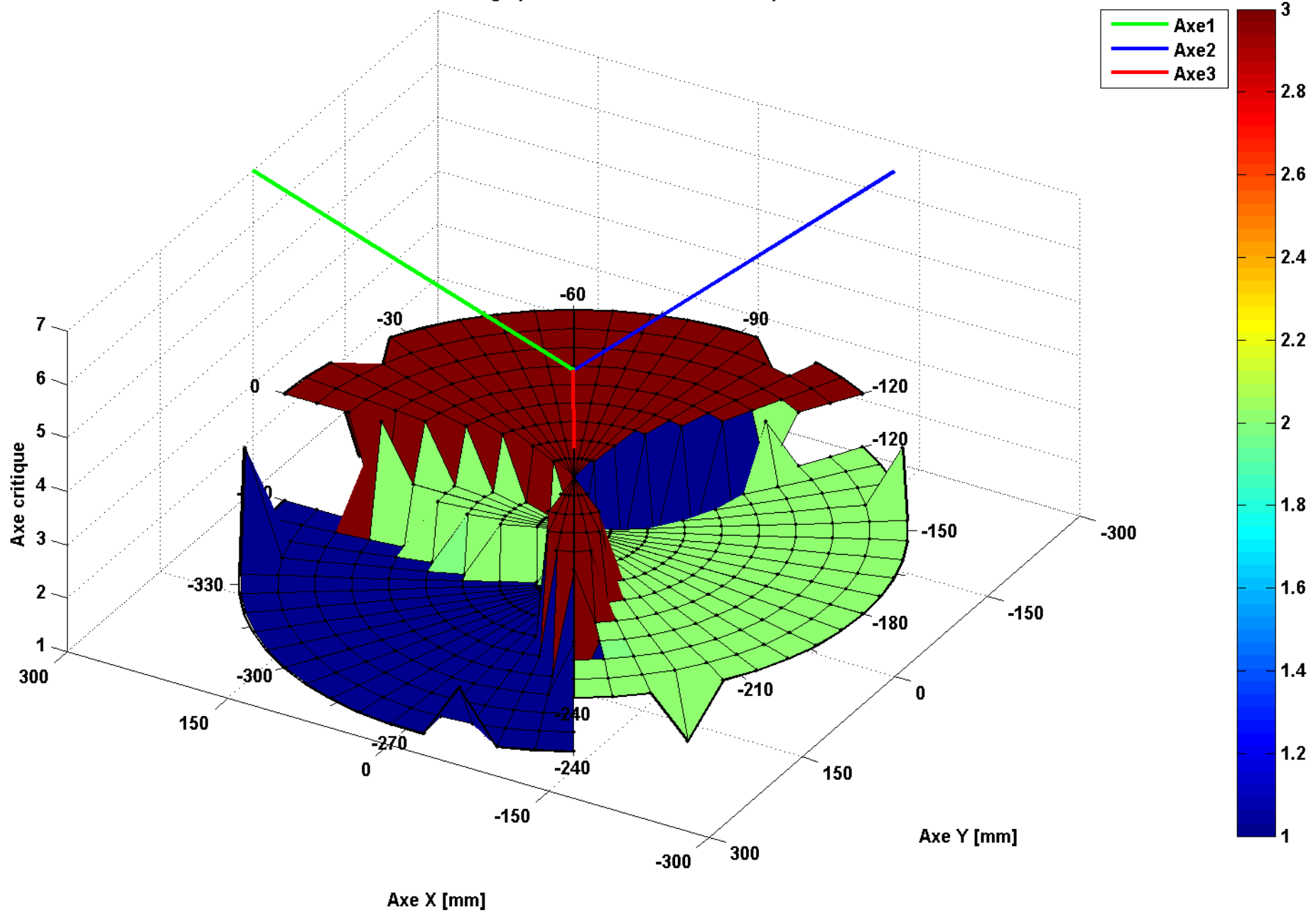
Série radiale 6*100 points de mesures plot mode : surf
Numéro de la cartographie : 40 Variable : Accelération critique [mm/s²]



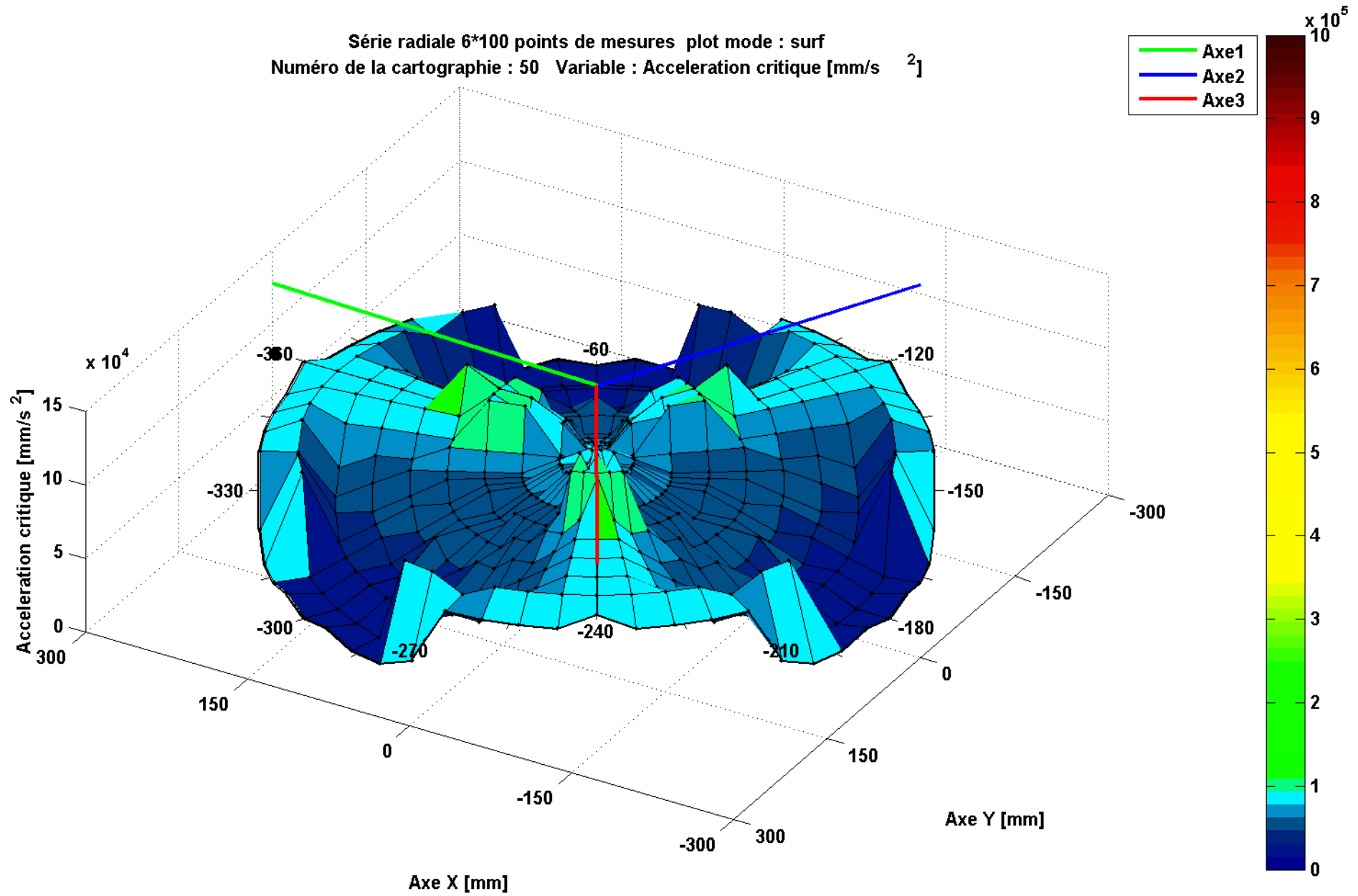
Série radiale 6*100 points de mesures plot mode : surf
Numéro de la cartographie : 40 Variable : Variance critique [mm²/s²]



Série radiale 6*100 points de mesures plot mode : surf
Numéro de la cartographie : 40 Variable : Axe critique



Série radiale 6*100 points de mesures plot mode : surf
Numéro de la cartographie : 50 Variable : Accelération critique [mm/s²]



Série radiale 6*100 points de mesures plot mode : surf
Numéro de la cartographie : 50 Variable : Variance critique [mm^2/s^2]

